NONLINEAR PROCESS MODELLING OF PH NEUTRALIZATION PROCESS IN CSTR USING SELECTIVE COMBINATION OF MULTIPLE NEURAL NETWORKS

DR. ZAINAL AHMAD

UNIVERSITI SAINS MALAYSIA
KAMPUS KEJURUTERAAN
2008

Laporán Akhir Projek Penyelidikan Jangka Pendek

Nonlinear Process Modeling of pH Neutralization Process in CSTR using Selective Combination of Multiple Neural Networks

by
Dr. Zainal Ahmad
Dr. Syamsul Rizal Abd Shukor

UNIVERSITI SAINS MALAYSIA

LAPORAN AKHIR PROJEK PENYELIDIKAN JANGKA PENDEK USM

Nama Penyelidik

Dr Zainal Ahmad
Dr Syamsul Rizal Abd Shukor

Tajuk Projek

Nonlinear Process Modeling of pH Neutralization Process in CSTR using Selective Combination of Multiple Neural Networks

Mac 2008

NIVERSITI SAINS MALAYSIA

LAPORAN AKHIR PROJEK PENYELIDIKAN JANGKA *PENDEK*

FINAL REPORT OF SHORT TERM RESEARCH PROJECT

Sila kemukakan laporan akhir ini melalui Jawatankuasa Penyelidikan di Pusat Pengajian dan Dekan/Pengarah/Ketua Jabatan kepada Pejabat Pelantar Penyelidikan

1. Nama Ketua Penyelidik: **Zainal Ahmad**
Name of Research Leader

☐ Profesor Madya/ *Assoc. Prof.* ■ Dr./ *Dr.* ☐ Encik/Puan/Cik *Mr/Mrs/Ms*

2. Pusat Tanggungjawab (PTJ): **Pusat Pengajian Kejuruteraan Kimia**
School/Department

3. Nama Penyelidik Bersama: **Dr Syamsul Rizal Abd Shukor**
Name of Co-Researcher

4. Tajuk Projek:
Title of Project

Nonlinear Process Modeling of pH Neutralization Process in CSTR using Selective Combination of Multiple Neural Networks

5. Ringkasan Penilaian/*Summary of Assessment:*	Tidak Mencukupi *Inadequate*		Boleh Diterima *Acceptable*	Sangat Baik *Very Good*	
	1	2	3	4	5
i) Pencapaian objektif projek: *Achievement of project objectives*	☐	☐	☐	■	☐
ii) Kualiti output: *Quality of outputs*	☐	☐	☐	■	☐
iii) Kualiti impak: *Quality of impacts*	☐	☐	■	☐	☐
iv) Pemindahan teknologi/potensi pengkomersialan: *Technology transfer/commercialization potential*	☐	☐	■	☐	☐
v) Kualiti dan usahasama : *Quality and intensity of collaboration*	☐	☐	■	☐	☐
vi) Penilaian kepentingan secara keseluruhan: *Overall assessment of benefits*	☐	☐	☐	■	☐

6. Abstrak Penyelidikan

(Perlu disediakan di antara 100 - 200 perkataan di dalam **Bahasa Malaysia dan juga Bahasa Inggeris**. Abstrak ini akan dimuatkan dalam Laporan Tahunan Bahagian Penyelidikan & Inovasi sebagai satu cara untuk menyampaikan dapatan projek tuan/puan kepada pihak Universiti & masyarakat luar).

Abstract of Research

(An abstract of between 100 and 200 words must be prepared in Bahasa Malaysia and in English).
This abstract will be included in the Annual Report of the Research and Innovation Section at a later date as a means of presenting the project findings of the researcher/s to the University and the community at large)

Abstract

pH control problem is very important in many chemical and biological systems and especially in waste treatment plants. The neutralization is very fast and occurs as a result of a simple reaction. However, from the control point of view it is very difficult problem to handle because of its high nonlinearity due to the varying gain and varying dynamics with respect to the operating point. Introduction of Artificial Neural Networks (ANNs) in modeling of process for control purposes is very useful due to their flexibility applications. In this research, feedforward neural network (NN) model technique are developed to predict the performance of a pH neutralization, which uses a sulphuric acid as the acidic stream and sodium hydroxide aques as the bes stream. Despite of many advantages of ANN that have been mentioned in the literature, some problems that can deteriorate neural networks performance such as lack of generalization has been bothering researchers. This problem has lead to a new approach in applying neural networks that is called as multiple neural networks (MNN). In MNN, the individual networks are developed from bootstrap re-samples of the original training and testing data sets. Instead of combining all the developed networks, these research propose selective combination techniques using backward elimination method. This techniques essentially combine those individual networks that, when combined, can significantly improve model generalization, in the other words, at first, all the individual networks are initially aggregated and some of the individual networks are then gradually eliminated until the aggregated network error on the original training and testing data sets cannot be further reduced. The analysis on the ability of neural network modeling is based on sum square error (SSE), mean square error (MSE), relative correlation (R-square) and residual error. The application results demonstrate that the multiple neural network (MNN) model techniques significantly create great model generalization.

Abstrak

Masalah pengawalan pH adalah amat penting dalam kebanyakan proses kimia mahupun biologi terutamanya dalam sistem rawatan air sisa. Dalam sistem ini, proses peneutralan berlaku begitu pantas dan hanya disebabkan oleh tindakbalas yang ringkas. Walau bagaimana pun, ianya adalah masalah yang sukar dari aspek sistem kawalan. Ini sebabkan oleh sistem yang sangat tidak lelurus yang berpunca dari dapatan dan dinamik sistem yang berubah-ubah. Pengenalan jaringan neural dalam permodelan dan juga sistem kawalan adalah satu langkah yang amat berguna disebabkan oleh ciri-ciri jaringan ini yang fleksibel. Dalam kajian ini, model jaringan neural suap depan dibangunkan untuk meramal prestasi proses peneutralan pH dengan menggunakan asid sulfurik sebagai laluan asid and natrium hidroksida sebagai laluan alkali. Walaupun terdapat banyak keistimewaan jaringan neural, terdapat juga celaan yang boleh memesong prestasi jaringan neural seperti kurangnya kebolehan untuk menyepadankan antara model dan data sebenar. Masalah ini telah menyedarkan para pengkaji dan mengambil langkah untuk mengaplikasikan satu lagi cabangan jaringan neural yang dipanggil jaringan neural pelbagai. Dalam jaringan neural pelbagai, setiap jaringan neural dibangunkan dengan kaedah pengsampelan semula ikat-but (bootstrap) set data latihan dan ujian yang asal. Bagi kajian ini, teknik kombinasi terpilih dengan kaedah penyingkiran belakang digunakan walaupun terdapat kaedah yang mengkombinasikan semua jaringan neural. Teknik ini mengkombinasikan jaringan neural yang mana dapat meningkatkan keberkesanan proses penyepadanan data model dan asal. Dalam kata lain, semua jaringan neural pada mulanya diagregatkan dan kemudian sebahagian darinya disingkirkan sehingga ralat jaringan agregat bagi set data latihan dan ujian mencapai tahap paling minima. Kaedah yang digunakan untuk mengukur kebolehan dan keberkesanan model jaringan neural adalah berdasarkan ralat jumlah kuasa dua, ralat min kuasa dua, pekali kolerasi relatif dan ralat baki. Keputusan bagi kajian ini menunjukkan bahawa teknik pemodelan jaringan neural pelbagai ini menghasilkan model yang mampu menyepadankan data model dan asal dengan baik.

7. **Sila sediakan laporan teknikal lengkap yang menerangkan keseluruhan projek ini.**
[Sila gunakan kertas berasingan]
Applicant are required to prepare a Comprehensive Technical Report explaning the project.
(This report must be appended separately)

Please refer Appendix A

Senaraikan kata kunci yang mencerminkan penyelidikan anda:
List the key words that reflects your research:

English	**Bahasa Malaysia**
neural networks	jaringan neural
multiple neural networks	jaringan neural pelbagai
selective combination MNN	kombinasi terpilih jaringan neural pelbagai
nonlinear process modeling	pemodelan process tidak lelurus
pH neutralization process	proses peneutralan pH

8. **Output dan Faedah Projek**
Output and Benefits of Project

(a) * **Penerbitan Jurnal**
Publication of Journals
(Sila nyatakan jenis, tajuk, pengarang/editor, tahun terbitan dan di mana telah diterbit/diserahkan)
(State type, title, author/editor, publication year and where it has been published/submitted)

1. Jurnal
    1. **Ahmad Z** , R A Mat Noor. Zhang, J. 2008. Multiple Neural Networks Modeling Techniques in Process Control:A Review, Asia Pacific Journal of Chemical Engineering (Submitted).

2. Persidangan
    1. **Ahmad Z** and R A Mat Noor. 2007. Improving Single Neural Network Model for Real pH Neutralization Process using Boostrap Re-sampling Technique. In the Proceeding of World Engineering Congress (WEC 2007), Penang Malaysia, pp 40-48.

    2. **Ahmad Z** and F Roslin. 2007. Modeling of Real pH Neutralisation Process using Multiple Neural Networks (MNN) Combination Technique. In the proceeding International Conference on Control, Instrumentation and Mechatronics Engineering, Johor Bahru ,Johor. CD-ROM.

    3. **Ahmad Z** and R A Mat Noor. 2007. A Review of Neural Network Modeling Technique: Multiple and Single Neural Networks, In the proceeding of SomChe 2007, CD-ROM.

    4. **Ahmad, Z** and R A Mat Noor. 2008. Improving Long Range Model Prediction through Backward Elimination Method in Multiple Neural Networks Combination. Submitted for ICONIP 2008, New Zealand.

    5. **Ahmad, Z** and Sachitananthan, S. 2008. Studying the Effect of CSTR Capacity and Stirring Rate in Controlling pH Neutralization Process in WWT Plant. Submitted for ICENV 2008, Penang, Malaysia.

Please refer Appendix B

(b) **Faedah-faedah lain seperti perkembangan produk, pengkomersialan produk/pendaftaran paten atau impak kepada dasar dan masyarakat.**
State other benefits such as product development, product commercialisation/patent registration or impact on source and society.

Tiada

* Sila berikan salinan/*Kindly provide copies*

(c) **Latihan Sumber Manusia**
Training in Human Resources

i) Pelajar Sarjana:
Graduates Students
(Perincikan nama, ijazah dan status)
(Provide names, degrees and status)

Tiada

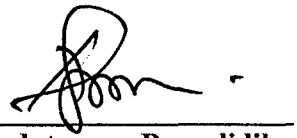
ii) Lain-lain:
Others
Projek Pelajar Tahun Akhir
1. Fairouze Roslin
   Tajuk:. Modeling of real pH Neutralization Process using Neural Networks
   Tahun: 2006/2007

2. Sharlinda Salim Satchinthanathan
   Tajuk: Studying the Effect of CSTR Capacity and Stirring Rate in controlling pH Neutralization Process
   Tahun: 2007/2008

Research Assistant (Jun – November 2007)
1. Rabiatul 'Adawiah Mat Noor (Jun – November 2007)

9. Peralatan yang Telah Dibeli:
Equipment that has been purchased

Tiada
Please refer Appendix C



Tandatangan Penyelidik
Signature of Researcher

15/05/08

Tarikh
Date

Komen Jawatankuasa Penyelidikan Pusat Pengajian/Pusat
Comments by the Research Committees of Schools/Centres

Projek penyelidikan ini telah bejaya dan tamat mengikut objektif asal seperti dalam cadangan asal.

23/5

PROFESOR ABDUL LATIF AHMAD, CEng FIChemE
Dekan
Pusat Pengajian Kejuruteraan Kimia
Kampus Kejuruteraan
Universiti Sains Malaysia, Seri Ampangan
14300 Nibong Tebal, Seberang Perai Selatan
Pulau Pinang

TANDATANGAN PENGERUSI
JAWATANKUASA PENYELIDIKAN
PUSAT PENGAJIAN/PUSAT
Signature of Chairman
[Research Committee of School/Centre]

Tarikh
Date

APPENDIX A

COMPREHENSIVE TECHNICAL REPORT

The aim of this works is to develop neural network modeling technique using real application of pH neutralization process and also develop a selective combination in multiple neural network modeling technique. This report consists of 4 sections which is pH neutralization rig, data sampling, result and discussion and lastly conclusion. The background/introduction of this study will not be given as this already been presented in the previous research proposal.

1.0 pH Neutralization Set Up

A schematic sketch of the experimental set up is shown in Figure 1. It consists of a 2L continuous stirred tank reactor (CSTR), supply tanks, pumps, pH electrodes (pHE), pH transmitter (pHT), a VR200 recorder (pHR) and a controller (pHIC). Agitation is provided in the reactor by means of a mechanical stirrer. Two supply tanks, Tank 1 and 2 each of 35L capacity contain the required base and acid are connected to each pump. Two liquid streams, a strong acid 0.01M H_2SO_4 is feed into the CSTR at a constant flow rate by a masterflex pump and a strong base 0.1M NaOH is feed in by a metering pump. An exit valve is manually adjust to ensure constant liquid volume in the CSTR and allows the effluent to flow out continuously into the waste tank. pH at the outlet of the CSTR is monitored by a pH transmitter through a pH electrode. The pH transmitter reading is sent to the recorder and controller. The control objective will be achieved through manipulation of base flow rate that receive corrective signal from the controller. A sampling period of 1s is recorded in the recorder.

1.1 Raw materials and chemicals

Alkaline (bes) solution

1. Sodium hydroxide (NaOH) 0.1 M

Acid solution.

1. Strong acid – Sulphuric acid (H_2SO_4) 0.01M

1.2 Experimental set up

In this part, the overall experiments set up will be show according to the flow chart in Figure 2. The instrumentations are important including where it must be in good condition. Some of them like pH electrode (sensor), pH transmitter, recorder, controller and pump must be calibrated first. After that, other equipments like CSTR reactor, stirrer, and tubing must be prepared so that the experiment can be run. If there is problems occur during the test especially on the accuracy of the reading, the instrumentations part must re-check or recalibrated to make sure that the data taken from this experiment is in high quality (less noise and accurate).

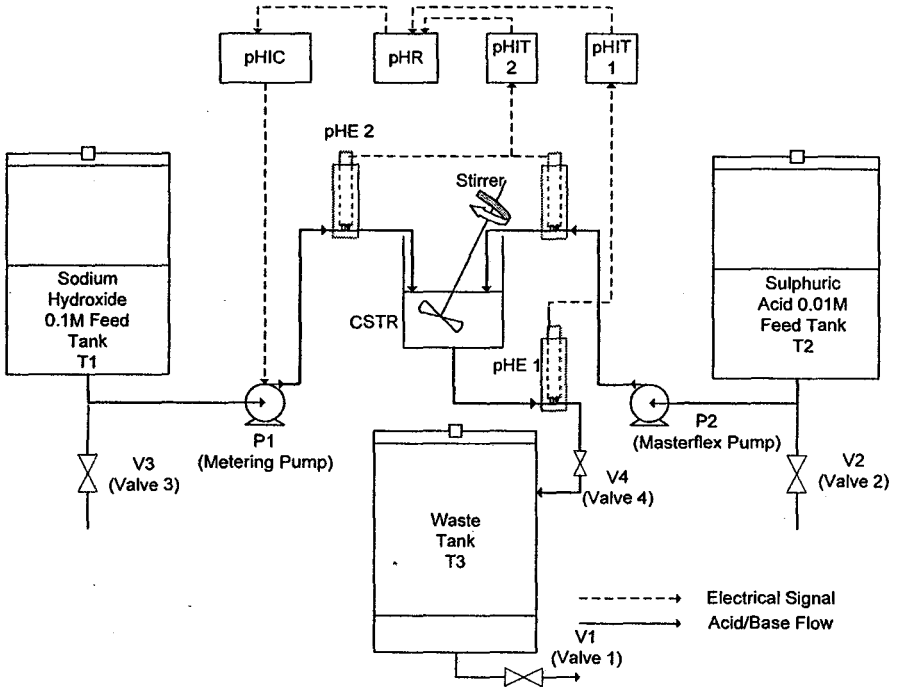


Figure 1: Schematic sketch of the experimental set-up

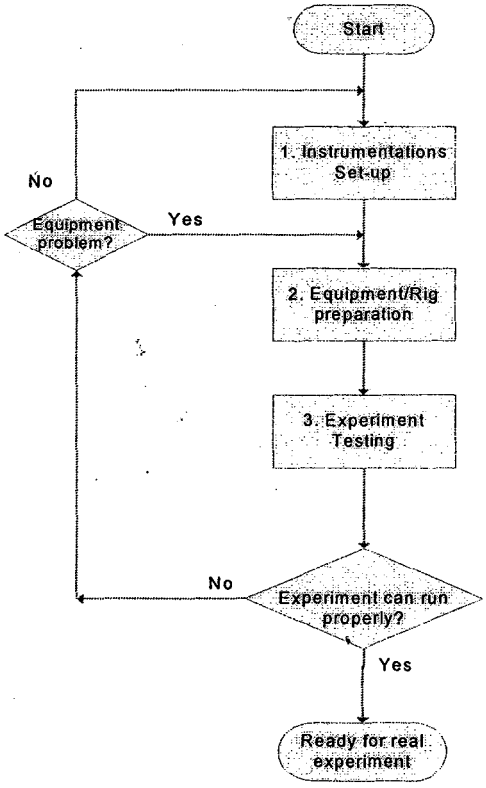


Figure 2: A methodology for the experiment start-up

1.2.1 Pump Calibration

1. To calibrate two different type of pump.

a. Metering pump

b. Masterflex pump

1.1.2 Materials required

1. Tank (21 inch long by 15.5-inch wide).
2. Stop watch.

1.1.3 Procedures

1. Each tank is filled up with a certain volume. Ensure that the valve V1, V2 and V3 are closed.
2. A metering pump is selected, and switch on.
3. The pump calibration is started by manually set the stroke length on the pump into 10%
4. Adjust the manipulated variable percent, MV% on the controller for 10%.
5. As water drop into a mixer tank that pass through the tube plastic which is connected to pump, the stopwatch as a timer was started and allowed to run throughout the calibration.
6. Continue to monitor the falling water level in mixer tank.
7. As soon as pumped to a certain amount of level (500 ml), turn the pump off and the water level and time were recorded.
8. After the draining ends, the tank was filling up again. Allow the water level to return to the original height.
9. Next, continued to increasingly the MV% to 20, 30,40,50,60,70,80,90 and 100.
10. After finish the 100% for the MV, step 3 is continued by increasingly stroke % until approximately 20, 30,40,50,60,70,80,90 and 100.
11. The pump is calibrated within 10 to 100 stroke%.
12. The experiment is continued to masterflex pump without changing the MV%.
13. Record the flow rate (ml per second) for each pump started the value stored in the Recorder.

1.1.4 Results and Graph

Table 1 Time in second, (s) needed to fill the 500 ml reactor for metering pump, P1

Metering pump		Pump stroke %									
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
MV %	**0**	0	0	0	0	0	0	0	0	0	0
	10	5136	2049	1184	1616	1158	920	695	668	602	584
	20	2076	932	539	717	533	415	348	304	447	263
	30	1290	593	344	457	334	263	217	194	176	167
	40	936	443	254	335	246	196	165	142	129	125
	50	758	365	201	267	198	157	131	115	103	99
	60	664	303	169	224	165	129	103	95	85	85
	70	577	274	134	191	139	106	90	80	72	69
	80	515	237	121	163	120	95	80	70	64	61
	90	490	207	111	151	107	84	71	64	57	54
	100	444	186	98	134	99	75	64	58	51	49
	106.3	443	174	94	133	98	75	64	58	51	49

Table 2 Time in second, (s) needed to fill the 500 ml reactor for masterflex pump, P2

Masterflex pump	stroke/min				
stroke length	**20**	**40**	**60**	**80**	**100**
20	981	211	345	292	215
40	552	277	212	154	120
60	407	219	154	155	90
80	332	166	127	96	75
100	287	143	108	82	63

All reading in unit second(s)

Calibration graph

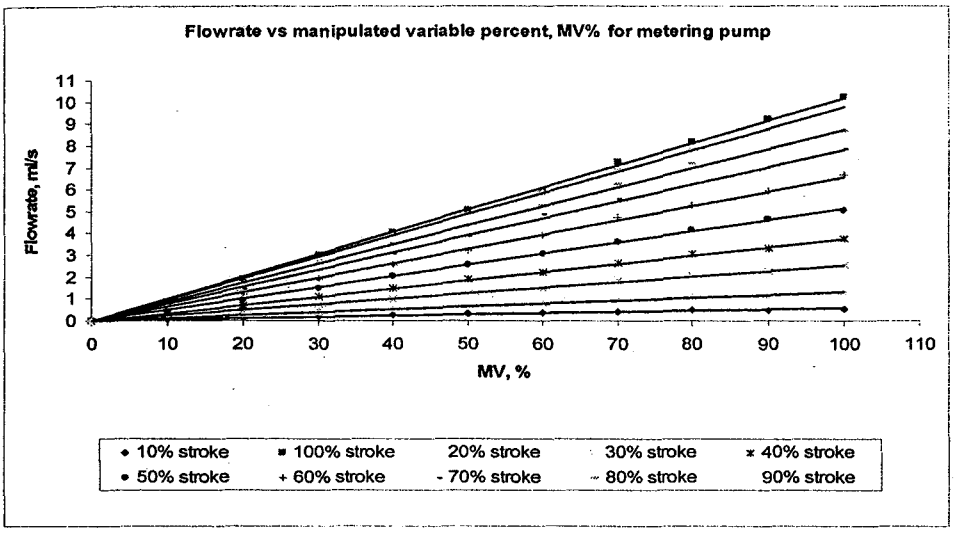


Figure 3: Flowrate versus manipulated variable, MV% for metering pump

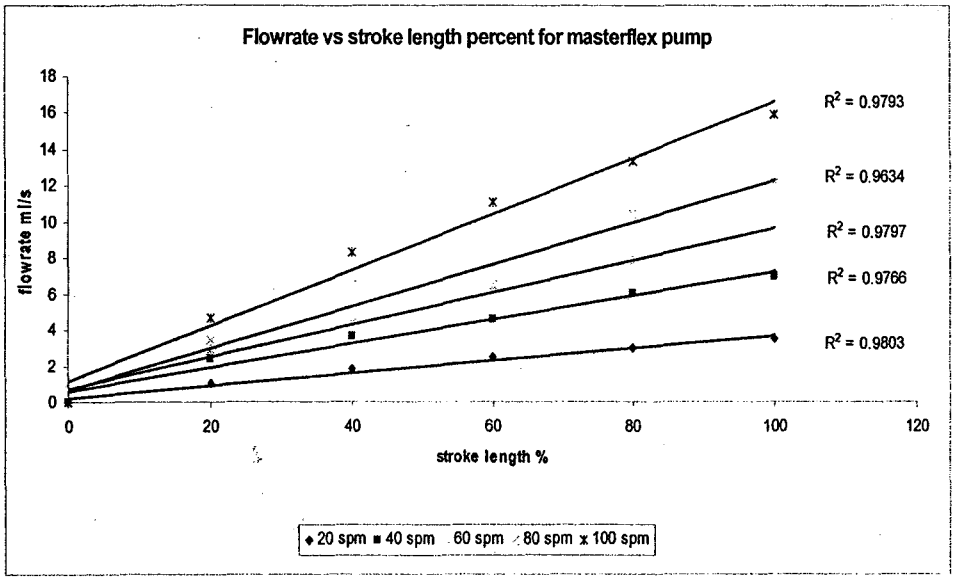


Figure 4: Flowrate versus stroke length for masterflex pump

Figure 3 and Figure 4 shown that both metering pump and the masterflex pump are in the good condition and can be use to run this experiment. For data generating purposes, the masterflex pump is kept at 20% stroke length and 20 spm constantly mean while the metering pump is varies from 20,30,50,60,70,80 and to 90 %stroke.

2.0 Data sampling

The instruments must be in a good condition before an experiment can be run in order to achieve a good result and also obtained the quality data for modeling. Therefore, several instruments; recorder, pH sensor and transmitter, needs to be calibrate in order to justify that these instruments are in good conditions and works well. Once the instruments were calibrated, the wires are ensured to be connected at the right terminals.

2.1 Recorder VR200 Calibration

The objective of this calibration is to check whether the reading of the pH value from the input and output signal is accurate or same signal. It clearly shown in Figure 5, that pH values are proportional with its input signal. Therefore the signals transfer to the recorder and linearly corresponding the physical reading of the pH. The data was recorded at every one second.

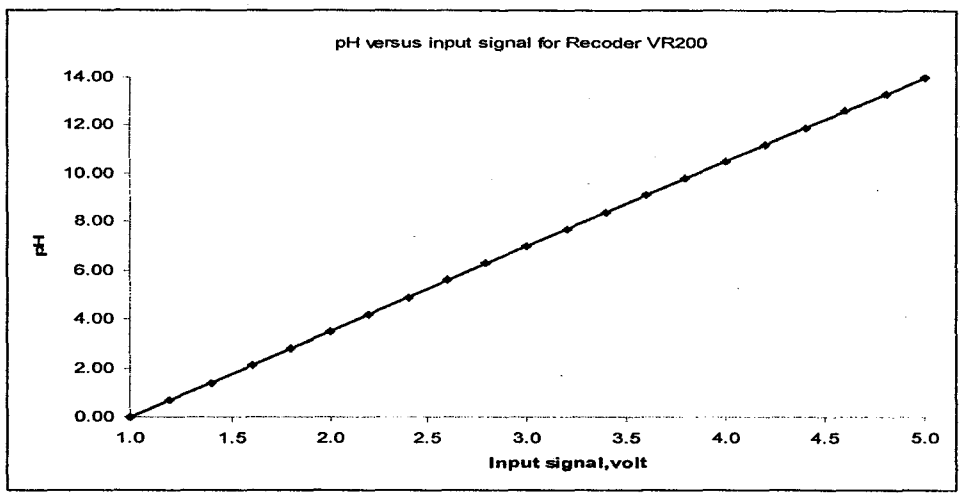


Figure 5: Calibrated result of Recorder VR200

2.2 pH Sensor and Transmitter Calibration

The objective of this calibration is to check the capability and sensitivity of the reading. In this case is the pH reading from the reactor to the recorder and also to the controller? pH tester has been used as a reference point for this calibration as we assume that the pH tester will gave an accurate reading. Based on the error calculated in Table 3, the maximum error was only 0.4, thus it is assumed that these errors are small and negligible.

Table 3: Errors calculated at pH transmitter reading

pH Tester	pH Transmitter	Error
2.0	1.8	0.2
2.8	2.7	0.1
3.1	2.8	0.3

3.6	3.9	0.3
4.6	5.0	0.4
5.5	5.5	0.0
5.8	6.0	0.2
6.2	6.5	0.3
6.6	7.0	0.4
6.8	7.2	0.4
7.0	7.3	0.3
7.4	7.7	0.3
8.0	8.2	0.2
8.6	8.8	0.2
8.8	9.0	0.2

2.3 Instrumentation and wiring

The objectives are to make sure that all the instrumentations have been connected correctly at the right terminals to ensure the accuracy and safety during the experiment. The connections of the instruments are shown in Figure 6. Once the wiring connections were checked, an experiment was run to test the links of each instrument. A perfect connection showed that pH transmitter 1 reading was displayed at the controller and recorder. Then, the auto mode and set point were set at the controller and it should be manipulating the base flow rates to reach the set point and the MV, % of the metering pump is also recorded.

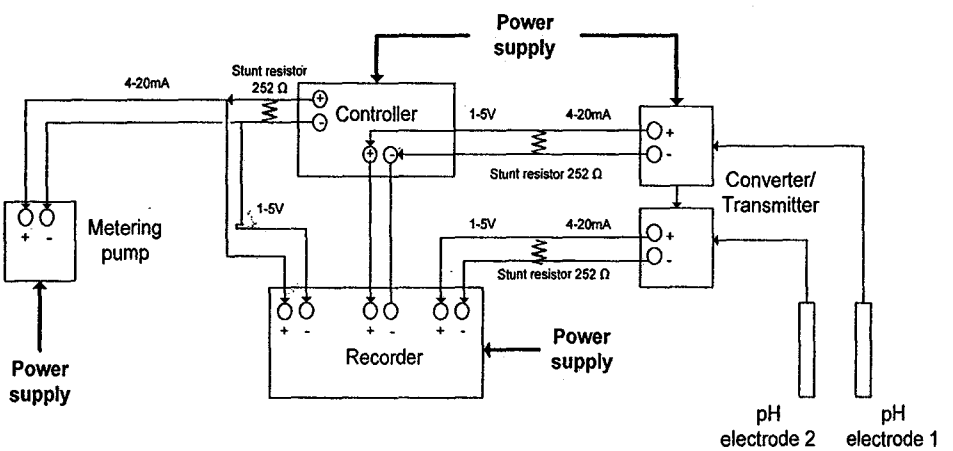


Figure 6: Instrumentation wiring block diagram

2.4 Modeling technique: Training Using Levenberg-Marquardt method.

Levenberg-Marquardt is an algorithm that trains a neural network 10 to 100 faster than the usual gradient descent backpropagation method. It will always compute the approximate Hessian matrix, which has dimensions n-by-n. The Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as $H = J^T J$ and the gradient can be computed as $g = J^T e$ where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix. The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \qquad (1)$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm. In the following code, we reinitialize our previous network and retrain it using the Levenberg-Marquardt algorithm. The training parameters for trainlm are epochs, show, goal, time, min_grad, max_fail, mu, mu_dec, mu_inc, mu_max, mem_reduc. Once the network weights and biases have been initialized, the network is ready for training. The network can be trained for function approximation (nonlinear regression), pattern association, or pattern classification. The training process requires a set of examples of proper network behavior - network inputs p and target outputs t. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function. The default performance function for feedforward networks is mean square error mse - the average squared error between the network outputs a and the target outputs t. Training one hidden layer neural network using Levenberg-Marquardt method.

function [w1,b1,w2,b2,ise1,ise2]=nntrlm(w1,b1,f1,w2,b2,f2,x,y,xt,yt,opt)
[w1,b1,w2,b2,ise1,ise2]=nntrlm(w1,b1,f1,w2,b2,f2,x,y,xt,yt,opt)
w1 - hidden layer weights, nh x n
b1 - hidden layer bias, nh x 1
f1 - activiation function for hidden layer
w2 - output layer weights, 1 x nh

b2 - output layer bias, 1 x 1
f2 - activation function for output layer
x - input data (training), nr1 x n
y - output data (training), nr1 x 1
xt - input data (testing), nr2 x n
yt - output data (testing), nr2 x 1
opt(1) - maximum iterations between training and testing, default=50
opt(2) - regularization parameter, default=0
Epochs = Maximum number of epochs to train.
Show = Epochs between showing progress.
Goal = Performance goal.
Time = maximum time to train in seconds.
Min_grad = minimum performance gradient .
Max_fail = maximum validation failures.

2.4.1 Scale

Before training, it is often useful to scale the inputs and targets so that they always fall within a specified range. The function scale can be used to scale inputs and targets so that they fall in the small range [-1,1]. Another approach for scaling network inputs and targets is to normalize the mean and standard deviation of the training set.
function sx = scale(x,means,stds)

2.4.2 Rescale

After training, it is often useful to rescale the outputs so that they always fall back within an original range. Another approach for rescaling network outputs is to normalize the mean and standard deviation of the data set.
function rx = rescale(x,mx,stdx)

2.4.3 Evaluation

1. Sum square error, SSE

   Sum of Squares Due to Error. This statistic measures the total deviation of the response values from the fit to the response values. It is also called the summed square of residuals and is usually labeled as SSE. A value closer to 0 indicates a better fit.

$$SSE = \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2 \qquad (2)$$

2. Relative error, R-square

   This statistic measures how successful the fit is in explaining the variation of the data. Put another way, R-square is the square of the correlation between the response values and the predicted response values. It is also called the square of the multiple correlation coefficients and the coefficient of multiple determinations. R-square can take on any value between 0 and

1, with a value closer to 1 indicating a better fit. For example, an R2 value of 0.8234 means that the fit explains 82.34% of the total variation in the data about the average. SST is also called the sum of squares about the mean, and is defined as

$$SST = \sum_{i=1}^{n} w_i (y_i - \bar{y}_i)^2 \tag{3}$$

$$R - square = R^2 = 1 - \frac{SSE}{SST} \tag{4}$$

3. Residuals

   The residuals from a fitted model are defined as the differences between the response data and the fit to the response data at each predictor value.

   residual = data – fit

   Mathematically, the residual for a specific predictor value is the difference between the response value y and the predicted response value $\hat{y}$

$$r = y - \hat{y} \tag{5}$$

   Assuming the model you fit to the data is correct, the residuals approximate the random errors. Therefore, if the residuals appear to behave randomly, it suggests that the model fits the data well. However, if the residuals display a systematic pattern, it is a clear sign that the model fits the data poorly.

4. Mean sum square error, MSSE

   This statistic is also known as the fit standard error and the standard error of the regression.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2 \tag{6}$$

   A MSE value closer to 0 indicates a better fit

2.5 Data sampling and Division

The numbers of data are not equal for every pump percent stroke because it depend on the reaction occur during the process as shown in Figure 7, Figure 8 and Figure 9 respectively. The data were separated into 3 divisions which are training, testing and validation (unseen data)

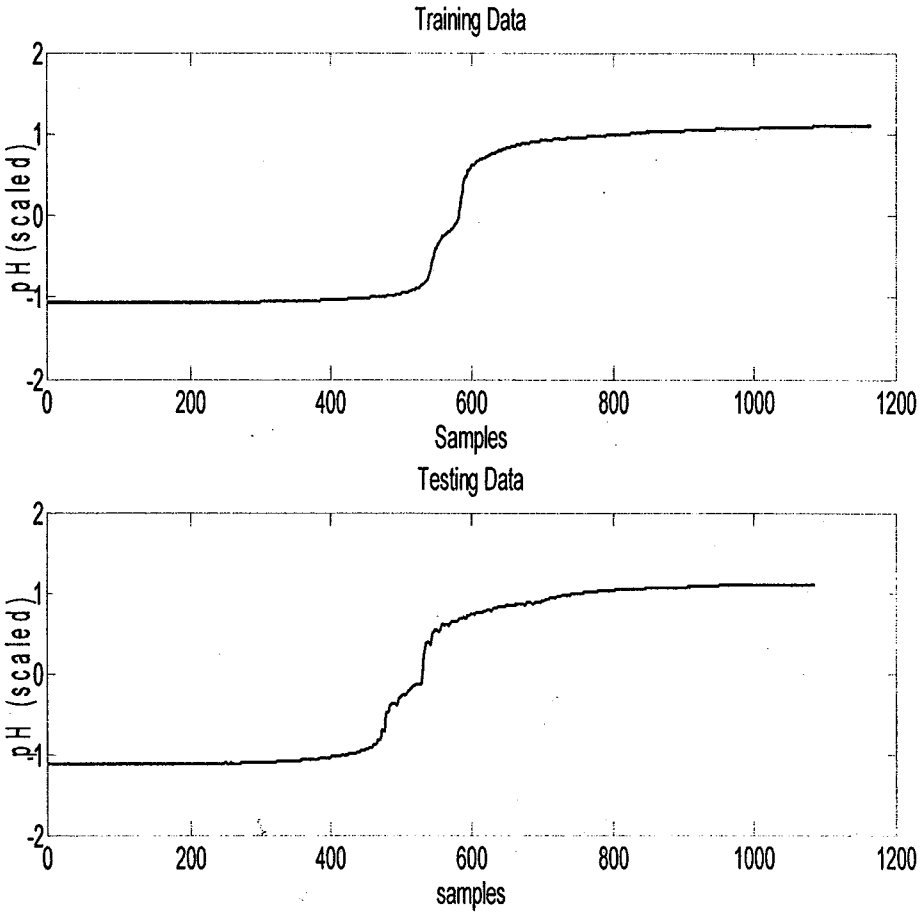


Figure 7: Original training 40 % stroke) and testing (100 % stroke) data for pH (scale)

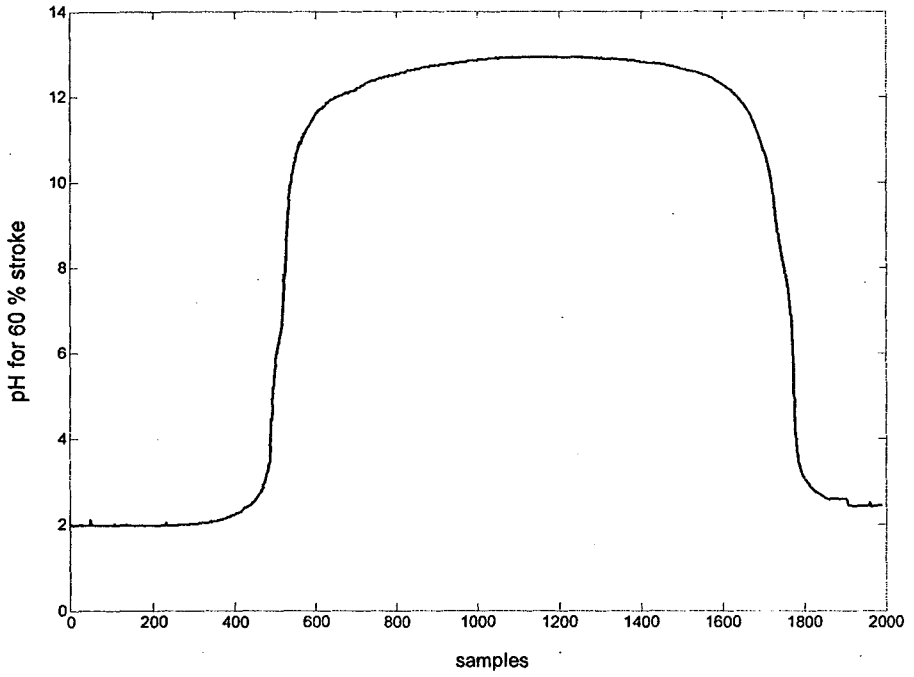


Figure 8: Original validation data for 60 % stroke data

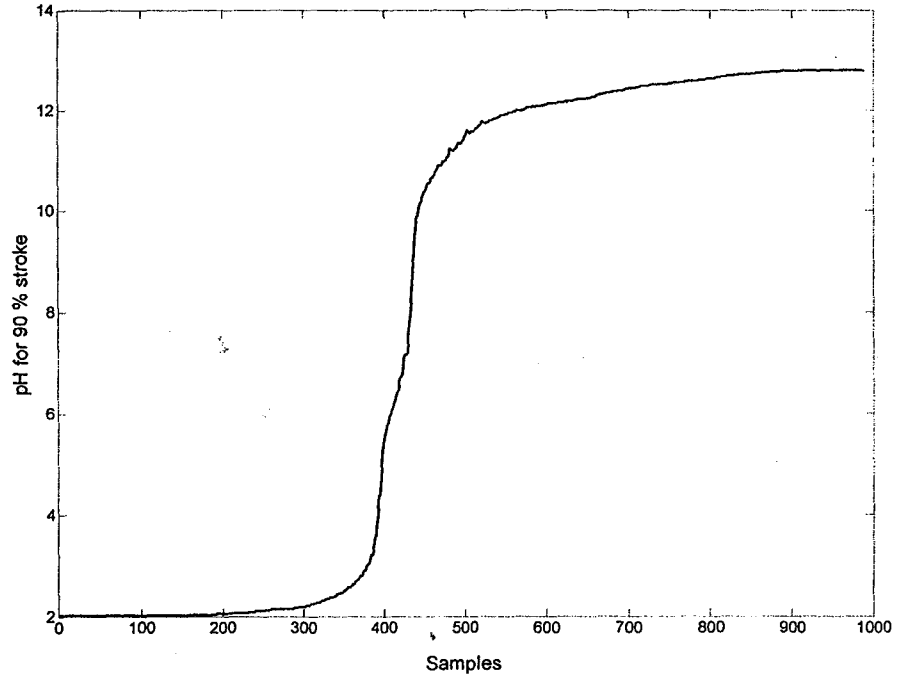


Figure 9: Original validation data for 90 % stroke data

3.0 Result and Discussion

3.1 Modeling using Single Neural Network (SNN)

Figure 10 and Figure 11 shows the model and actual output in the validation data for single neural networks (SNN) using original training and testing data. It clearly seen that the single neural networks was performed quite well. The predicted model output showed quite the same as the experiment data, but there is some errors occurred at the low pH region as well as at the end of the high region and also at the transition between the low region and middle region. This might be due to the transition of the pH especially from low region to higher region where the neutralization process was very fast, small changes in the input (acid flow) give a lot of affect to the process.

This modeling performance was supported by the quantitative analysis in Table 4 and also the residue analysis in Figure 12 and Figure 13 respectively. From Figure 12 and Figure 13, the relative error or the predicted and actual data is small which is around 0.05 and 0.1 respectively. It shows that the variation of error in this model is relatively reasonable and can be assume closed to zero. By evaluating both performances in graphical measures; the residuals appear randomly scattered around zero indicating that the model captured the experiment data well.

Further, in the Table 4, the sum square error for training is 0.0392 lower than testing 0.6935. The correlation coefficient for training is 1 and for the testing is 0.9994. Although the sum square error for testing is higher than the training, but these phenomena is expected due the model that been developed using training data.

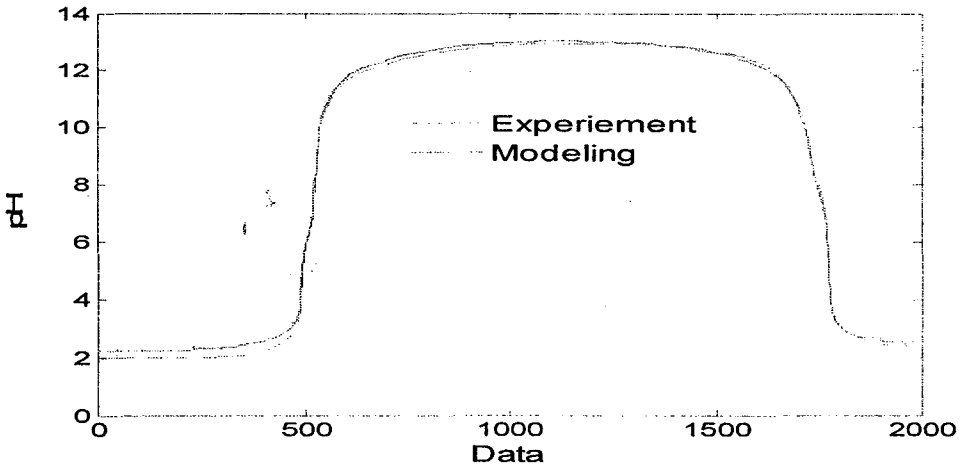


Figure 10: Validation for 60% stroke in pH

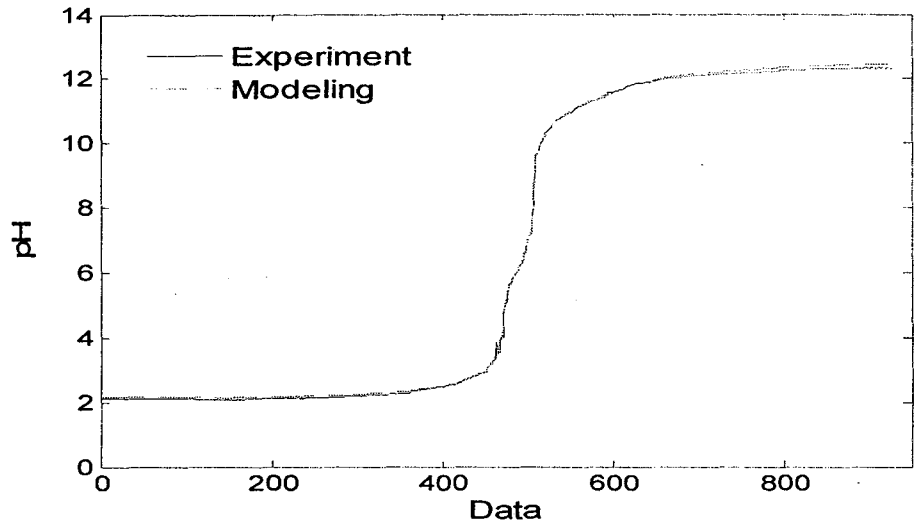


Figure 11: Validation for 90% stroke in actual pH

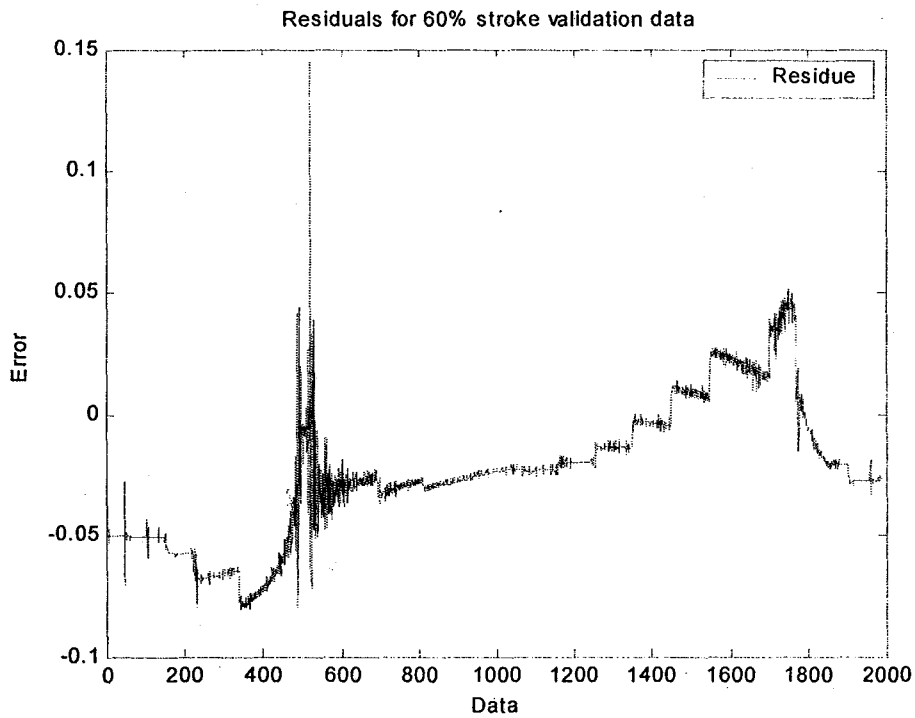


Figure 12: Residuals for validation 60% data

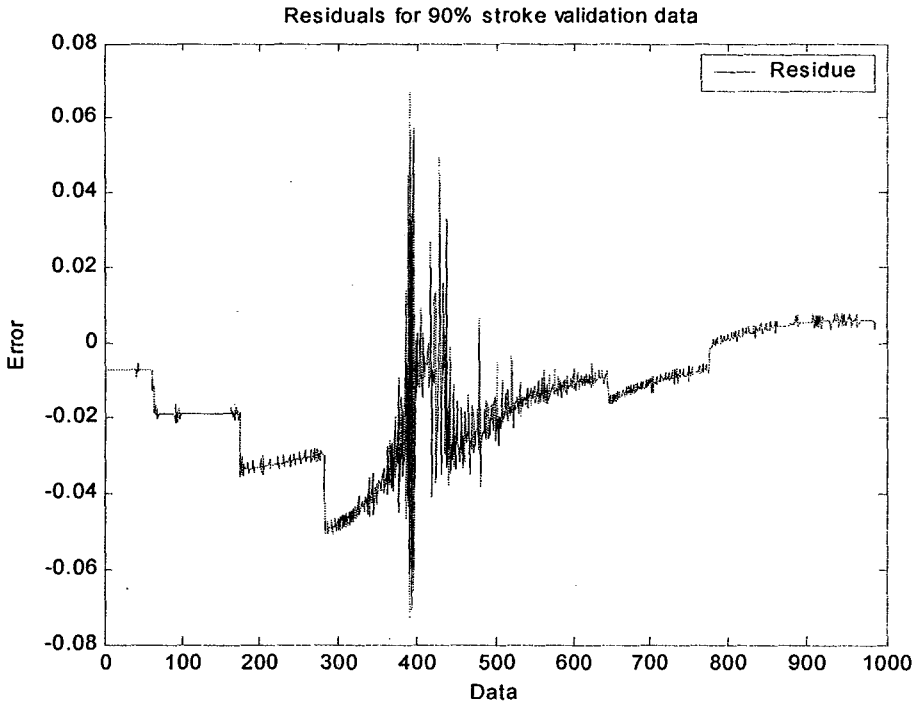


Figure 13: Residuals for validation 90% data

Table 4 Sum square error, mean square error, and R-square for validation data

Data	SSETr	SSETs	SSETv	MSSETv	RsquareTr	RsquareTs	RsquareTv
60	0.0392	0.6935	2.6757	0.0013	1	0.9994	0.9977
90	0.0392	0.6935	0.4584	4.64E-04	1	0.9994	0.9996

3.2 Modeling using Single Neural Network (SNN) using resamples Technique (bootstrap)

Bootstrap application or bootstrap technique was first introduced in 1979 as a computer based method for estimating the standard error of empirical distribution. In neural networks bootstrap basically relate or deals with the sampling to create random data sets for training and testing. By creating an equal number of bad and good data sampling, it actually improves the generalization ability because it helps the identification of the characteristic of the scarce class. The motivation of creating those different inputs or partitions is to create the effective neural network model and also network ensembles. The bootstrap or bagging basically refers to replication of a training data set where the bootstrap algorithm re-samples the original training data set. Some of the data samples may occur several times, and other may not occur in the sample at all. The individual training sets are independent and the neural networks can be trained in parallel.

As shown in Figure 14 and Figure 15, there is a different between the original data and the data after applying the boostrap re-sampling techniques.

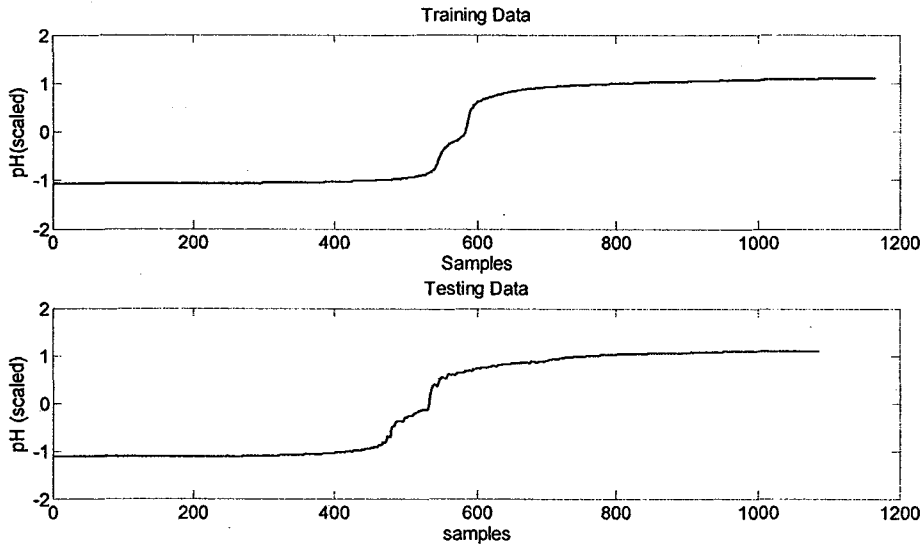


Figure 14: Original training and testing data for pH (scale)

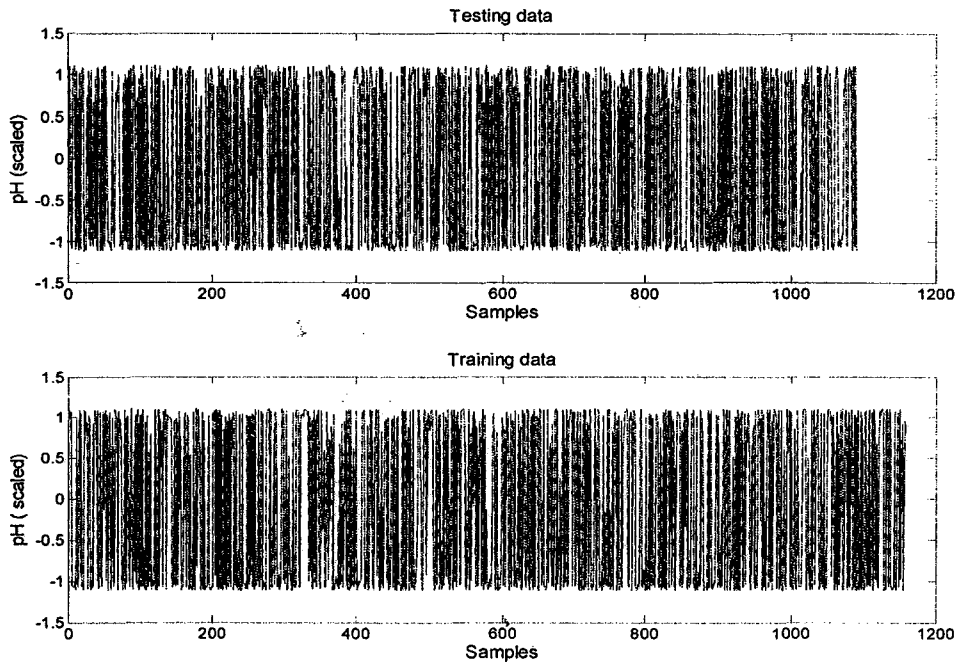


Figure 15: Training and testing data for pH (scale) after re-sampling using boostrap.

Then re-sampling technique using boostrap approach is applied and the result was shown in Figure 16 and Figure 17 for 60 % and 90 % stroke data respectively. It clearly seen that from Figure 16 and Figure 17, single neural networks prediction using resample technique is significantly better than single neural networked using original data generated in the previous section. The predicted and the experiment value can be seen exactly matching for both data. In order to test further the performance of the model, statistical analysis was carried out which is sum square error (SSE), mean square error (MSSE) and relative correlation R-square analysis.

The overall statistical analysis result of SSE, MSSE and relative correlation R-square shown in the Table 5. It is clearly shown in Table 5 that the SSE and the MSSE is quite small, the relative correlation (R-square) is 1 for re-sampling SNN while in original SNN prediction is slightly lower than 1. Meanwhile for SSE and MSSE, the original SNN produced higher value compare to re-sampling SNN. It is shown that the re-sampling SNN model can predict significantly well even though using real process data compare to original SNN.

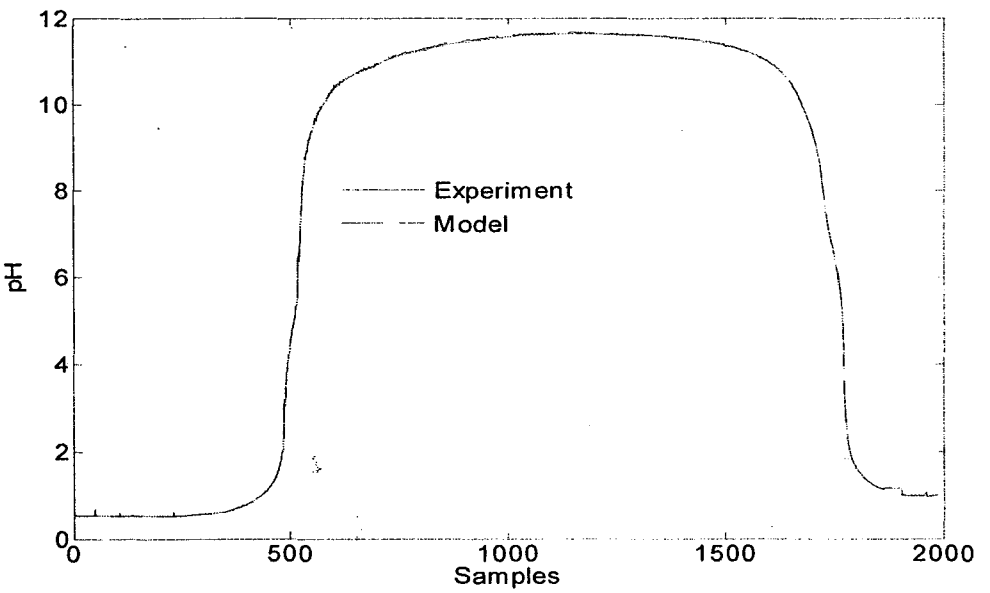


Figure 16: Validation output for re-sampling SNN for 60 % stroke data

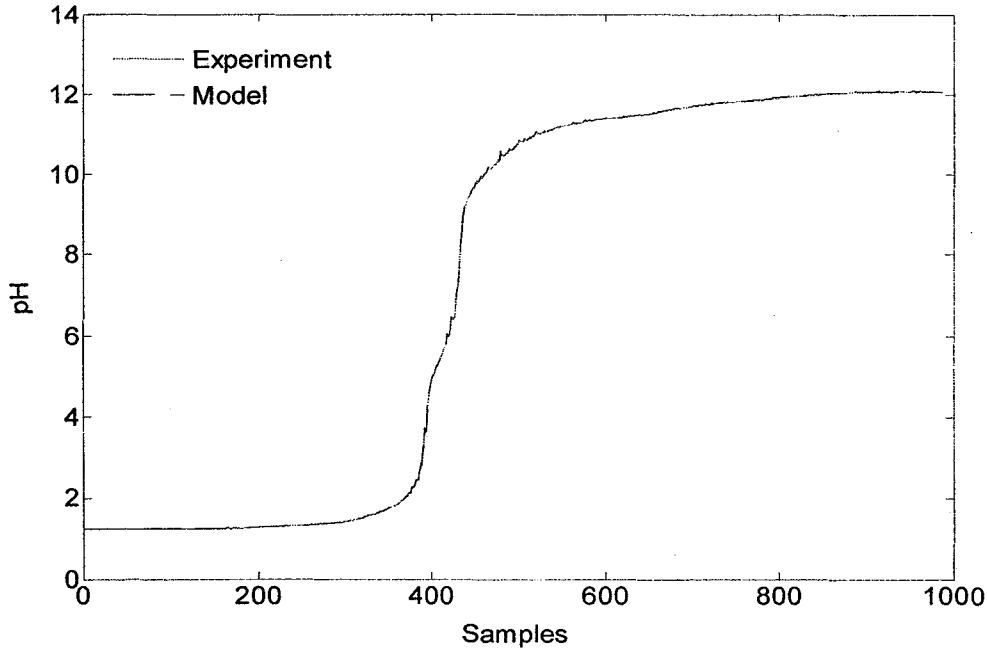


Figure 17: Validation output for re-sampling SNN for 90 % stroke data

Table 5: Result of the output based on the single neural networks application on the validation data.

Data	SSETv		MSSETv		RsquareTv	
	Original SNN	Re-sampling SNN	Original SNN	Re-sampling SNN	Original SNN	Re-sampling SNN
60	2.6757	0.0988	1.3E-3	4.9652E-005	0.9977	1.0000
90	0.4584	0.0442	4.64E-04	4.4782E-005	0.9996	1.0000

3.3 Modeling using Multiple Neural Networks (MNN) using resamples Technique (bootstrap)

Multiple neural networks (MNN) combination approach is applied and the result was shown in Figure 18 and Figure 19 for 60 % and 90 % stroke data respectively. It clearly seen that from Figure 18 and Figure 19, the multiple neural networks prediction is significantly better than single neural networked. The predicted and the experiment value can be seen exactly matching for both data. The performance of MNN combination is encouraging especially based on the residue analysis which is shown in Figure 20 and Figure 21. The residue is constant for MNN but for SNN is quite inconsistent especially in the transition of low and upper region. This contributed to the large number of SSE for SNN prediction.

In order to test further the performance of the model, statistical analysis was carried out which is sum square error (SSE), mean square error (MSSE) and relative correlation R-square analysis as well as residue analysis. The overall statistical analysis result of SSE, MSSE and relative correlation R-square shown in the Table 6 and Table 7. It is clearly shown in Table 6 that the SSE and the MSSE is quite small and in Table 7, the relative correlation (R-square) is nearly to 1 for MNN while in SNN prediction, it's slightly larger for SSE and MSSE. It is shown that the MNN combination model can predict significantly well even though using real process data.

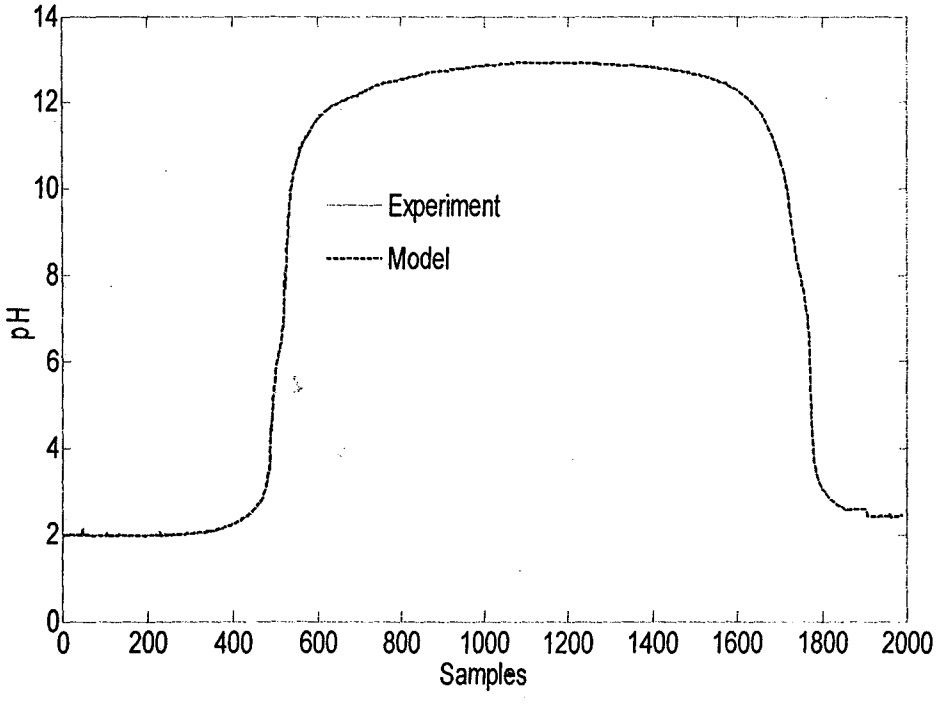


Figure 18: Multiple Neural Networks validation output for 60 % stroke data

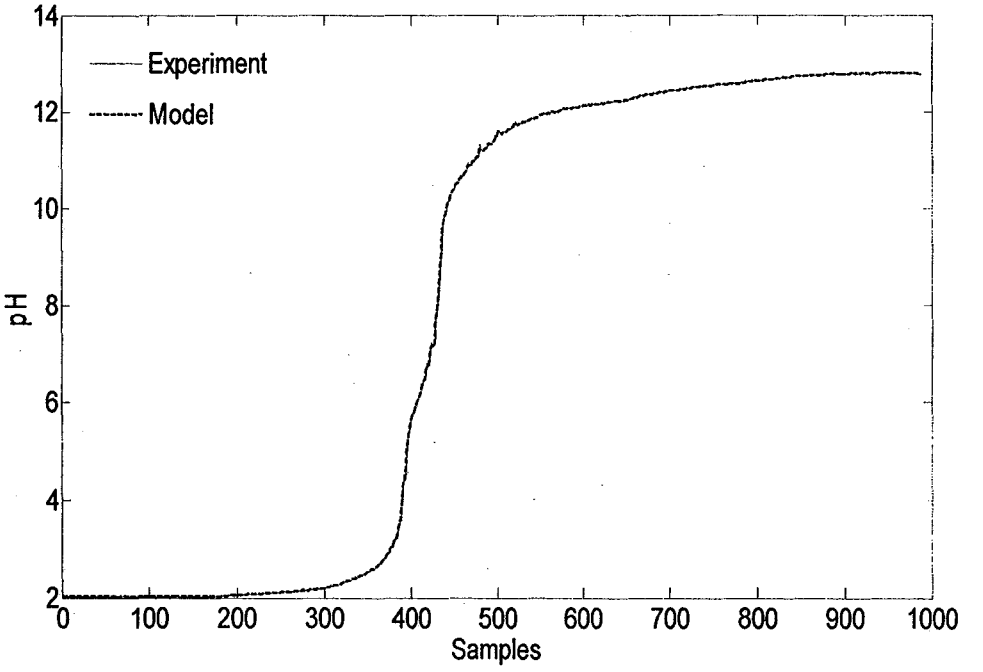


Figure 19: Multiple Neural Networks validation output for 90 % stroke data

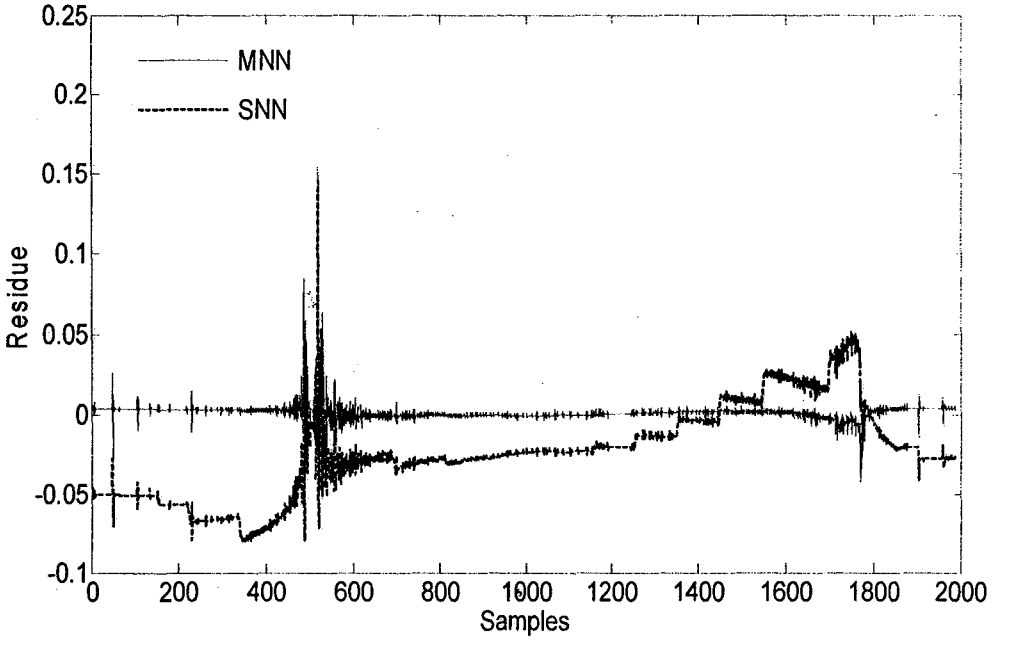


Figure 20: Residue for multiple neural networks (MNN) and single neural network (SNN) prediction for 60 % stroke data

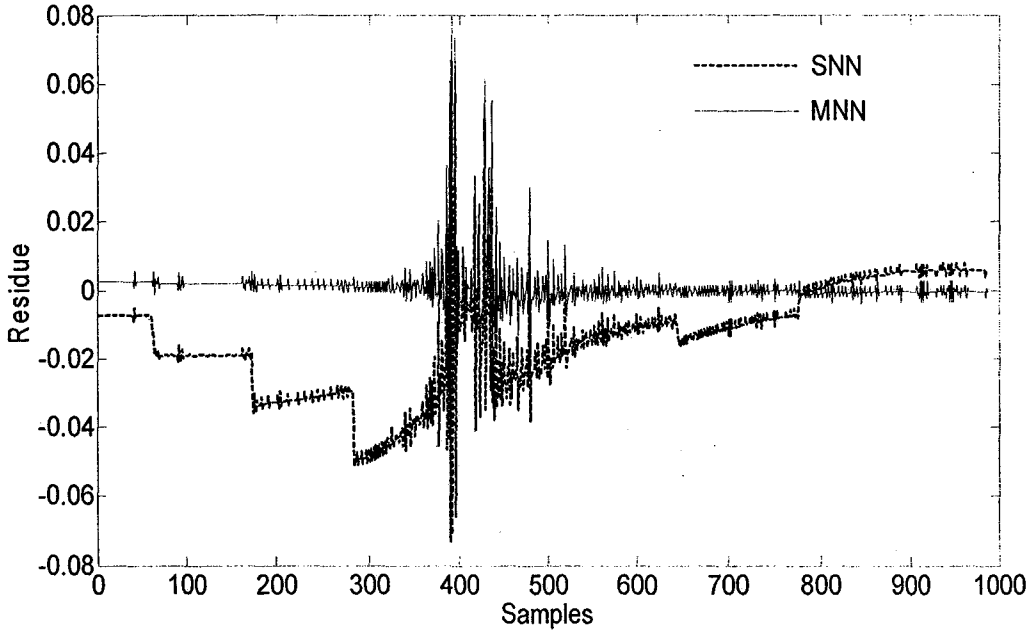


Figure 21: Residue for multiple neural networks (MNN) and single neural network (SNN) prediction for 90 % stroke data

Table 6: Result of the output based on the single and multiple neural networks application on the validation data.

Data	SSETv		MSSETv	
	SNN	MNN	SNN	MNN
60	2.6757	0.0880	0.0013	4.4234e-005
90	0.4584	0.0458	4.64E-04	4.6383e-005

Table 7. Result of the output based on the single and multiple neural networks application on the validation data for R-square.

Data	RsquareTv	
	SNN	MNN
60	0.9977	0.9999
90	0.9996	1.0000

3.4 Modeling using selective combination of Multiple Neural Networks (MNN).

Suppose that neural network models are to be developed from the data set $\{X, Y\}$, where $X \in R^{N \times p}$ is the input data, $Y \in R^{N \times q}$ is the output data, N is the number of samples, p is the number of input variables, and q is the number of output variables. To develop an aggregated neural network model containing n individual networks, the original data set can be re-sampled using bootstrap re-sampling with replacement to form n replications of the original data set. The n replications can be denoted as $\{X_{(1)}, Y_{(1)}\}$, $\{X_{(2)}, Y_{(2)}\}$, ..., $\{X_{(n)}, Y_{(n)}\}$, where $X_{(i)} \in R^{N \times p}$, $Y_{(i)} \in R^{N \times q}$, i=1, 2, ..., n. A neural network model can be developed on each of these replications, which can be partitioned into a training data set and a testing data set if cross-validation is used in network training and network structure selection. If the predictions of these n networks on the original data set are denoted as $\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}n$, then the sum of squared errors (SSE) of the ith network can be calculated as

$$SSE_i = \text{trace}[(Y - \hat{Y}_i)^T (Y - \hat{Y}_i)] \tag{7}$$

For the sake of simplicity in illustration, the simple average method is used in combining the selected networks. If all n networks are combined, then the aggregated network output is:

$$\hat{Y} = \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_i \tag{8}$$

At first we propose to apply support vector machine (SVM) technique. This technique basically refer to object recognition. Currently this technique is applied to regression and time series prediction task. Therefore it will be good if we can utilise the capability of the vector machine to combined the output based on the multiple neural netwoks model. However the SVM were hardly differentiate or select the best output of the MNN due to the MMN itself the predict the same pattern. Therefore to apply SVM we need to used totally different individual output then it can work. Thefore we proposed a step wise method using simple averaging approach as what we call backward elimination (BE) technique.

3.4.1 Backward Elimination

The BE approach begins with the aggregated neural network containing all the individual networks and removes one network at a time until the SSE on the training and testing data cannot be further reduced. The network deleted at each step is such selected that its deletion results in the largest reduction in the aggregated network SSE on the training and testing data. The BE method is summarized as follows:

Step 1 Generate n replications of the original data set using bootstrap re-sampling, $\{X_{(1)}, Y_{(1)}\}$, $\{X_{(2)}, Y_{(2)}\}$, ..., $\{X_{(n)}, Y_{(n)}\}$, and develop a neural network on each replication. Denote the prediction of the ith

network on the original data set as $\hat{Yi}$. Calculate the SSE of these networks on the original data using Eq (1).

Step 2 Set j=1 and denote I as a set containing the indices of the networks currently included in the aggregated network and I=[1, 2, ..., n]. Denote J as a set containing the indices of the networks currently deleted from the aggregated network and J=[], i.e. J is initially empty. Denote $\hat{Y}_{a,j}$ and SSE(j) as, respectively, the predictions and SSE of the aggregated network at stage j.

$$SSE(j) = \text{trace}[(\frac{1}{n}\sum_{i\in I}\hat{Y}_i - Y)^T(\frac{1}{n}\sum_{i\in I}\hat{Y}_i - Y)] \quad (9)$$

Step 3 If n-j=0, then go to Step 5;

else

j=j+1

for i∈I

$$\hat{Y}_{a,j}^{(i)} = \frac{1}{n-j}\sum_{l\in I-i}\hat{Y}_l$$

end

$$k = \arg\min_{i\in I}\text{trace}[(\hat{Y}_{a,j}^{(i)} - Y)^T(\hat{Y}_{a,j}^{(i)} - Y)]$$

$$SSE(j) = \text{trace}[(\hat{Y}_{a,j}^{(k)} - Y)^T(\hat{Y}_{a,j}^{(k)} - Y)]$$

Step 4 If SSE(j)≥SSE(j-1), then go to Step 5;

else

I=I - k (i.e. remove k from set I)

J=[J, k]

go to Step 3.

Step 5 Stop

Figure 22 show the multi steps-ahead prediction performance of individual neural networks. It can be seen from Figure 21 that the individual networks give inconsistent multi steps-ahead prediction performance on the training and testing data and on the unseen validation data. For example in Figure 22 shows that network number 14 among the networks with various structures gives the worst performance on the training and testing data. However, its performance on the unseen validation data is quite good. This demonstrates the non-robust nature of individual networks. Figure 23 shows the SSE of multi steps-ahead predictions from aggregated neural networks with various structures. The

aggregated networks under selective combination scheme give quite consistent prediction performance on the training and testing data and on the unseen validation data. This patent was also observed for the fixed structure.

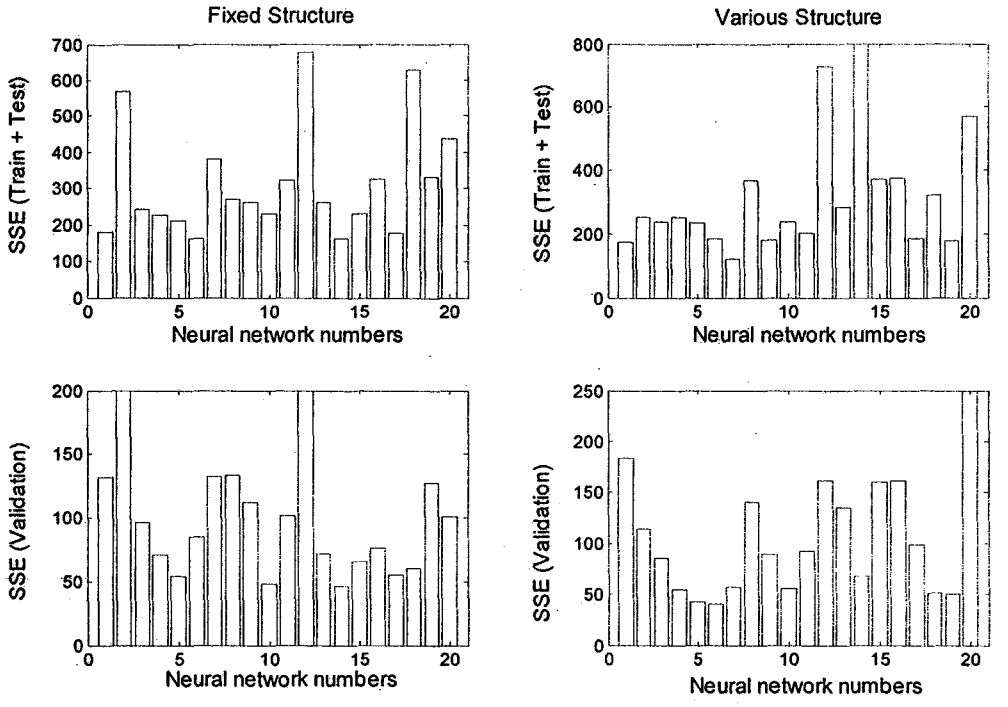


Figure 22: SSE of long range predictions from individual neural networks in pH neutralization process

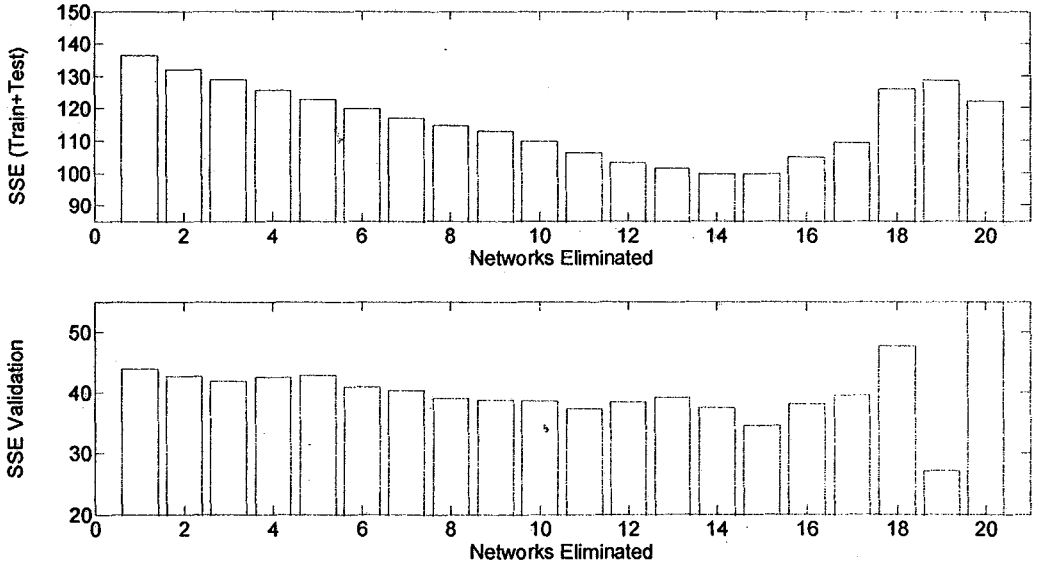


Figure 23: SSE from aggregated neural networks with various structures in pH neutralization process

Table 8 gives the SSE on the unseen validation data of different combination schemes. It can be seen that the worse one of BE selective combination schemes gives better performance than combining all the networks and the median of individual networks. In the BE selection methods 5 networks (networks 1, 6, 11, 14, and 17) and 7 networks (networks 1, 5, 7, 11, 17, 18, and 20) were combined for fixed and various structures. The median of the individual network SSE on the unseen validation data for fixed and various structures are 90.44 and 90.52 respectively.

Table 8: Overall Results for pH Neutralization Process

Combination schemes			SSE on validation data
Median	Fixed structure		90.44
	Various structures		90.52
Average	Feedback before combination	Fixed structure	57.31
		Various structures	43.84
BE	Feedback before combination	Fixed structure	41.77
		Various structures	37.44

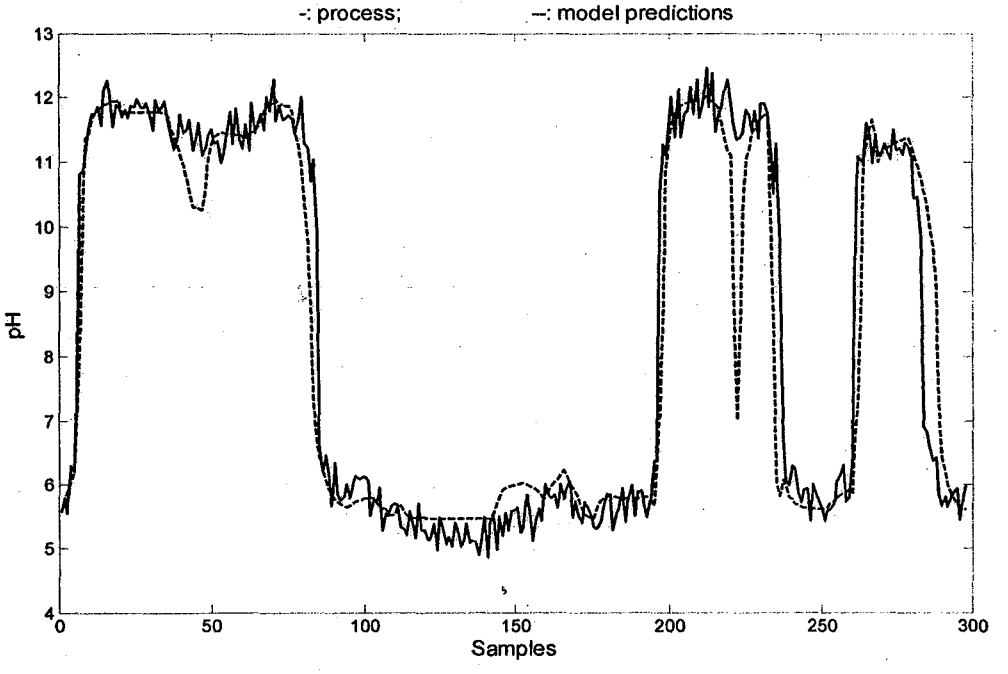


Figure 24: Long range predictions from the best aggregated neural network combination

Table 9: Mean and Standard Deviation When Varying the Parameter in Neural Network Modeling

Combination schemes			Mean	Std Deviation
Median	Fixed structure		93.48	3.52
	Various structures		94.43	4.38
Average	Feedback before combination	Fixed structure	59.16	4.99
		Various structures	51.36	4.41
BE	Feedback before combination	Fixed structure	50.47	3.14
		Various structures	38.37	1.29

The best combination scheme in this case is "BE with fixed structures with feedback before combination" with an SSE of 37.44 on the unseen validation data. Figure 24 shows the multi steps-ahead predictions from this aggregated neural network. Lastly, the initial parameter was change in order to test whether the proposed methods can get a consistent result even though some of the condition is different. The result is quite consistent for BE selection method where the mean and standard deviation is smaller compare to median and averaging methods as shown in Table 9.

3.5 Studying the Effect of CSTR Capacity and Stirring Rate in Controlling pH Neutralization Process

3.5.1 Case Study 1: Reactor Tank Capacities

The set points in pH adjustment processes are usually at the steepest part of the titration curve, near the neutral pH of 7. The process has extremely high gain or sensitivity at this point meaning a small amount of changes in reagent will cause remarkable changes in pH value. To study the effect of tank capacities on the system response, three different pH tracking were used. The servo was varied after 30 minutes at each set point from pH 7 to 9 followed by pH 5. Figure 26 shows the performance of the pH process under different tank capacities for servo control problems with the corresponding to the base flow rate.

When the set point tracking behavior of each tank was compared, the system was more capable of bringing the pH to the set points in the largest tank, 1.5L. By using this tank, PID showed a faster response with minimum oscillations and over/undershoots and settling times less than 10 minutes toward the set point than did in the 1.0L and 0.5L tank as in Figure 25. The errors caused by the set point changes were instantaneously sensed by the controller and immediate corrections in the base flow were taken as can be seen in Figure 26. This was probably because in 1.5 L tank capacity with 350 rpm stirring rates, the mixture of acid and base was well mixed. As base flows in the tank, the pH changes gradually resulting excellent pH control at each set point.

In 1.0L tank with 350rpm stirring rates, PID also showed good control performance but with oscillations and over/undershoots especially at set point pH 5. This was due to severe changes of base flow rates into the reactor had caused the pH suddenly changes until the controller finally reached a suitable flow rates of base. This had taken a longer time especially in obtaining an acidic solution of pH 5. As the controller was manipulating the base flow rates, a high amount of base was used.

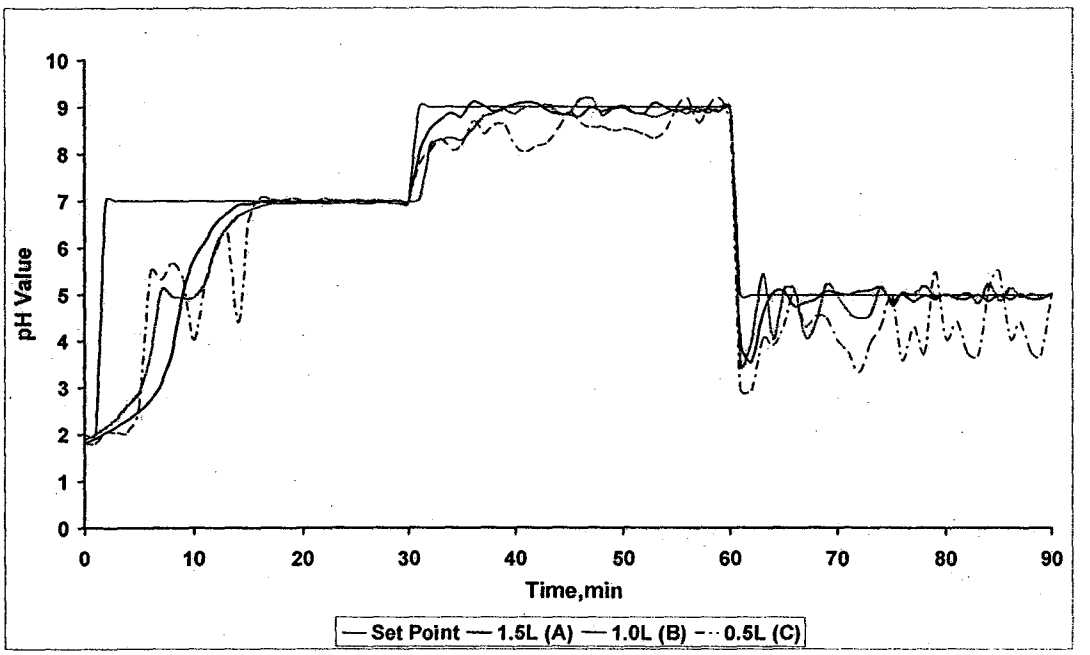


Figure 25: Set point changes at different tank capacities, Time (min) versus pH value

Meanwhile, in tank of 0.5L, PID showed very slow response with extremely poor performance. This was most probably because as the tank volume is smaller, minor changes of base flow rates will caused major changes of pH in the mixture. The controller could not obtain the suitable amount of base needed in this small volume of mixture as the pH was fluctuating. A high amount of base had been used in this process without success.

In order to assess the robustness of the controller at different capacities of tanks, its ability to maintain the pH value of the effluent stream at the neutral value of pH 7 in the presence of disturbances was examined. Instead of keeping the acid flow rate constant, it was changed from 1.5 ml/s to 3.4 ml/s for 15 seconds. It was disturbed respectively at time 30 minutes which was after it reached steady state at pH 7. This characteristic is important in applications, such as waste-water treatment, where disturbances should not cause the pH value of the effluent stream to deviate too much from the set point.

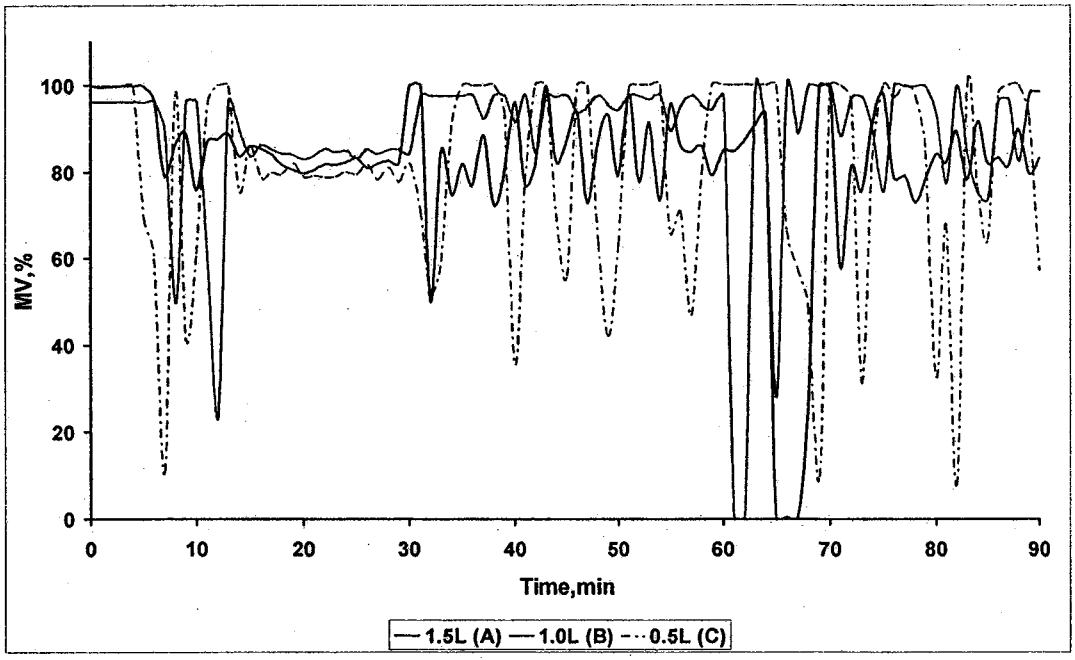


Figure 26: Changes of base flow rate, MV % versus time (min)

Figure 27 displayed the drop of pH caused by disturbance while Figure 28 showed the controller performance in rejecting disturbance. Tank A showed a small change of pH value because the disturbance was introduced only for a short time of 15 seconds, thus it took only 4 minutes to return back to pH 7. The pH value of Tank B drop to pH 5 with 7 minutes needed to recover back to the process pH while Tank C took 10 minutes to trace the set point from pH 3.5. All three different tank capacity managed to reject the disturbance but at different range of time.

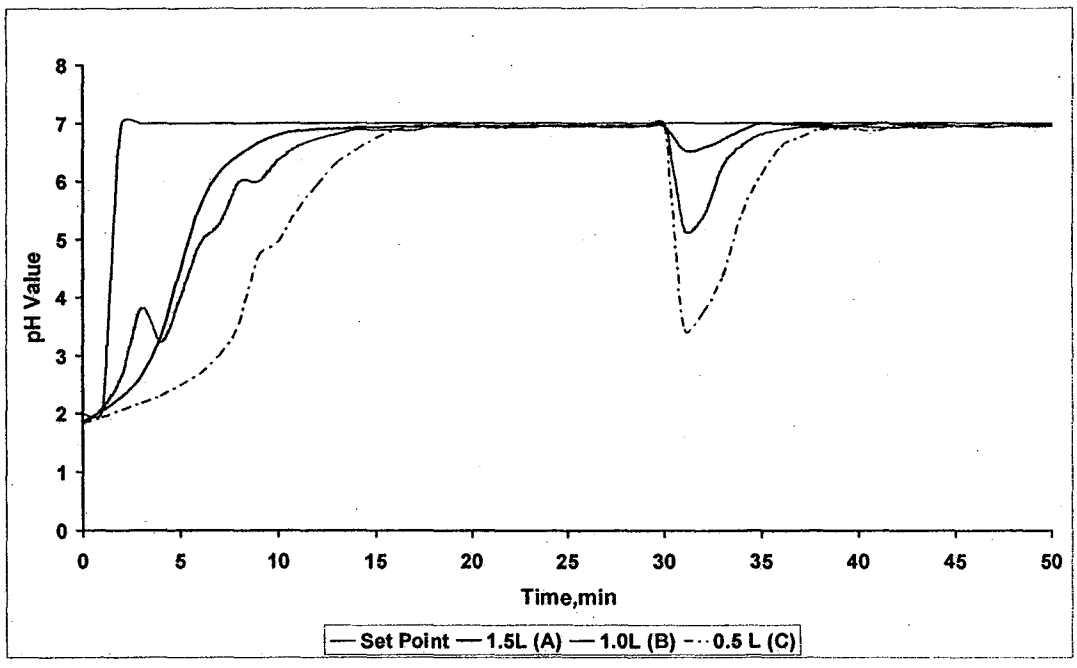


Figure 27: Disturbance effect at different tank capacities, Time (min) versus pH value

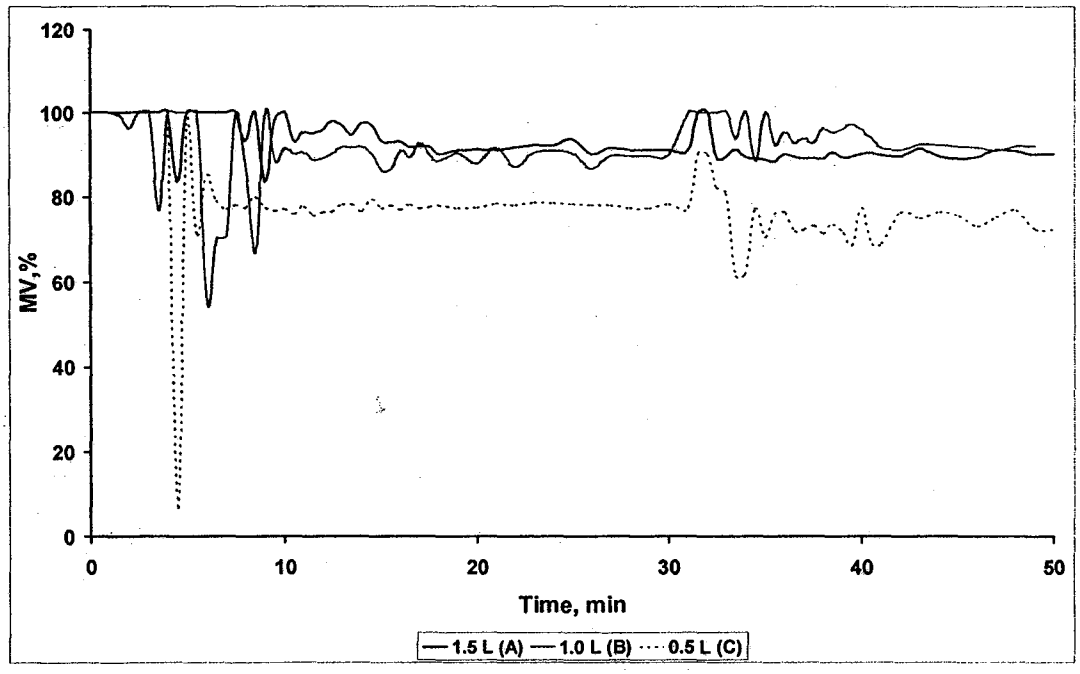


Figure 28: Changes of base flow rate, MV % versus time (min)

3.5.2 Case Study 2: Stirring Rates

The set points were changed similar to the first experiments which were pH 5, 7 and 9 to evaluate the controller response at different stirring rates. The servo was varied after 30 minutes at

each set point from pH 7 to 9 followed by pH 5. Figure 29 shows the control of pH at different stirring rates with the corresponding base flow rate is given in Figure 30.

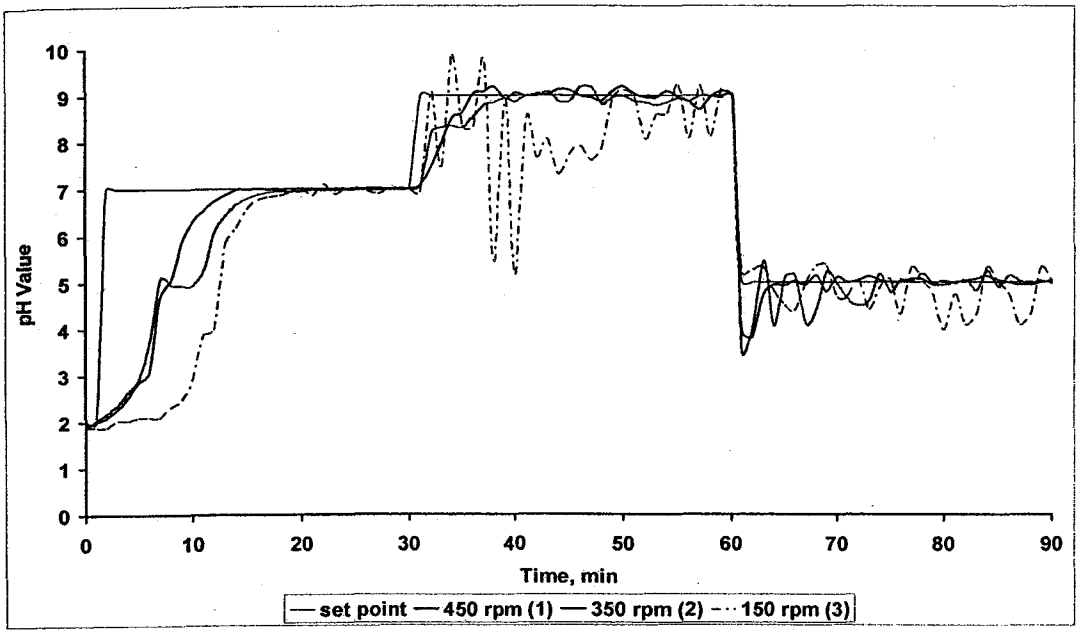


Figure 29: Set point changes at different stirring rates, Time (min) versus pH value

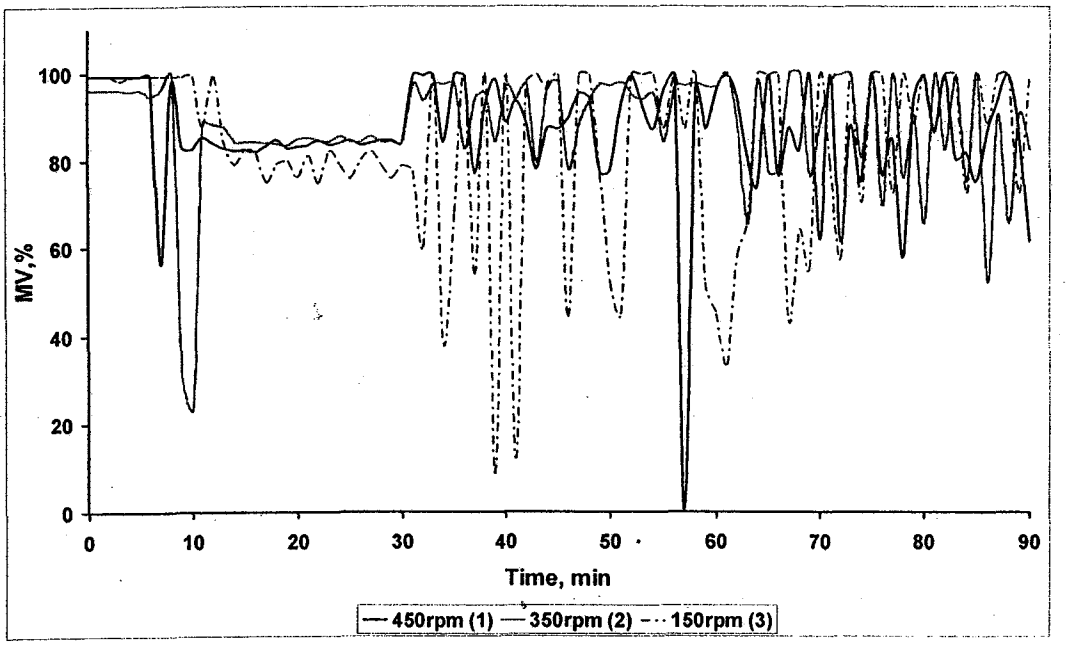


Figure 30: Changes of base flow rate, MV % versus time (min)

Overall, at 450rpm in 1.0L tank, the controller was able to control the pH value at different set points with minimum oscillations and over/undershoots. This resulted from a complete mixing in which acid and base molecules were dissociated completely and the pH changes are instantaneous. This

vigorous mixing ensures uniform composition through out the reaction tank thus an accurate measurement of pH was obtained and consequently an easier and accurate control of pH value. Referring to Figure 30, it can be seen that the controller was able to manipulate and maintained the base flow rates efficiently at each set point.

At rate 350rpm, the set point tracking response was not as good as at 450rpm especially at pH 5. In obtaining pH 5, the flow rates of base was decreased, thus stirring rate plays a crucial role in mixing this less amount of base in the mixture in a short time. This is probably the reason at 350rpm, the mixing was not fully complete and thus more time is needed to perfectly mix the acid-base mixture. Therefore, a longer time needed by the controller to reach the set points.

The mixing process with rate 150rpm was inadequate therefore it exhibited a poor controller performance. Inadequate mixing resulted inaccurate reading of pH measurement. Since this controller responded based on error of measurement and set point, it will proceed with its action. Therefore, as shown in Figure 30, several actions taken were inappropriate such as low amount of base were feed in, in order to obtain an alkaline solution of pH 9.

To study the controller ability to maintain the pH value of the effluent stream at the neutral value of pH 7, disturbance was introduce by increasing the acid flow rate for 20 seconds. The result was showed in Figure 31 and the controller response as in Figure 32. From the below Figure 31, at Rate 3, 150rpm, the pH changes faster followed by Rate 2 and then Rate 1. However, the rejection of disturbance at the three different rates was almost the same, after 10 minutes, the disturbance was rejected and back to its initial set point. Therefore, even though Rate 3 showed faster changes of pH, the time it took to reach the original pH is the same as Rate 1. Thus, Rate 1 can still be considered as the best stirring rates in rejecting disturbance.

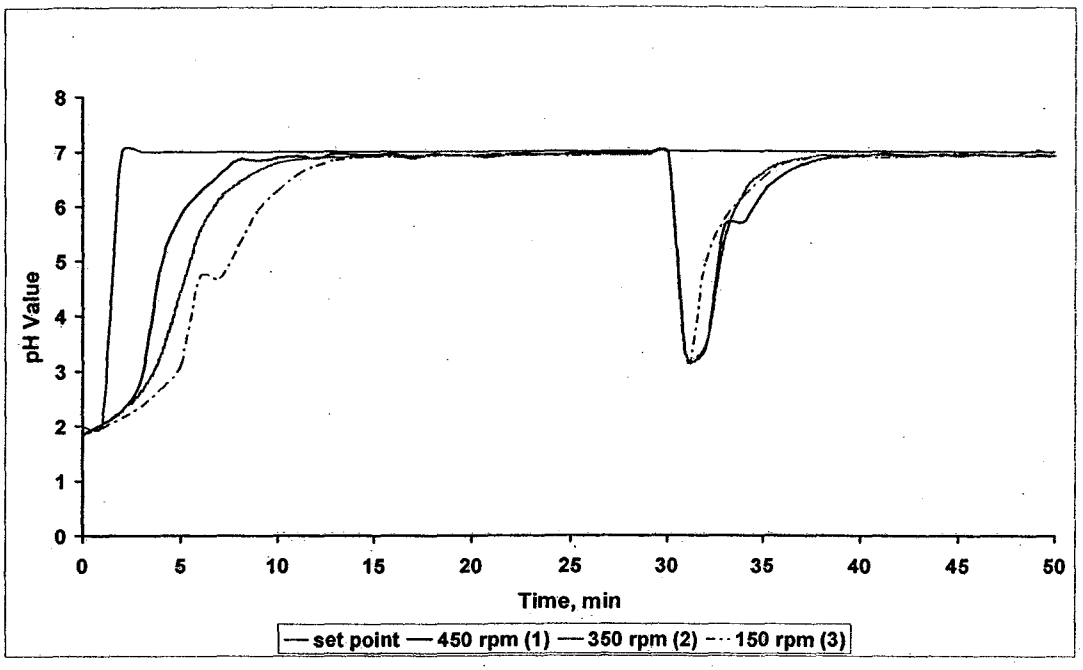


Figure 31: Disturbance effect at different stirring rates, Time (min) versus pH value

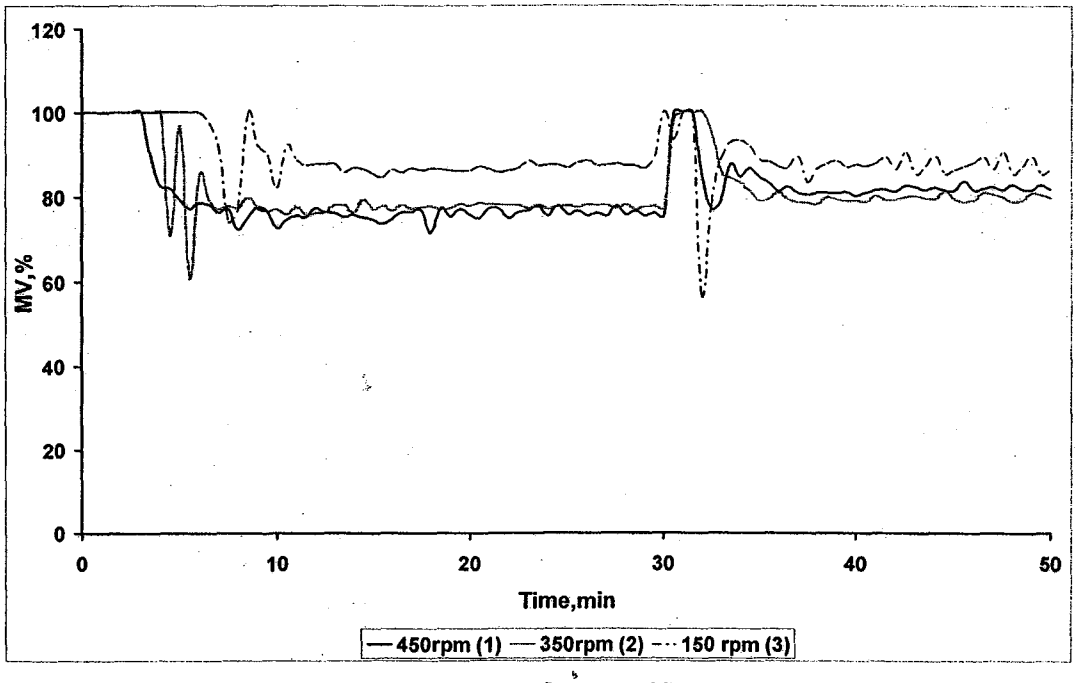


Figure 32: Changes of base flow rate, MV % versus time (min)

4.0 Conclusion

A single neural network (SNN) and multiple neural network (MNN) was developed to model the performance of a pH neutralization process using experimental data, which was subjected to a series of different stroke percent for sodium hydroxide stream. The inputs to the network were the sodium

hydroxide stream flow rate and metering pump percent stroke, and the output was the pH values of the effluent. The Levenberg–Marquardt optimization technique was used together with the 'early stopping' and regularization methods to improve the robustness of the network. Application to the real pH neutralization process shows that combining multiple neural networks (MNN) increased the robustness of neural network models compared to single neural network (SNN). The SSE is decreased as well as the increment of R-square analysis compare to single neural networks in all validation data. The result for multiple neural networks combination was consistent especially in residue analysis as well as in R-square.

Then the selective combination technique which is using Backward elimination methods is proposed in order to improve the model generalization performance. In the BE method, initially all individual networks are included in the aggregated network. Individual networks are then eliminated one at a time from the aggregated network until the aggregated network error on the original training and testing data cannot be further reduced. BE selective combination methods have shown their superiority compared to the combination of all networks and the median in this case study and it's concluded that combining multiple neural networks can significantly produced better models.

In addition to studying the dynamics of the pH neutralization itself, it was found that the dynamics of the system are totally dependent on the size of the reactor and also the rate of the stirrer. Therefore to model the system accurately, the size and the dynamics of the system must be included as part of the input to the system.

APPENDIX B

Multiple Neural Networks Modeling Techniques in Process Control:A Review

Journal:	*Asia-Pacific Journal of Chemical Engineering*
Manuscript ID:	APJ-08-0019
Wiley - Manuscript type:	Review
Date Submitted by the Author:	15-Feb-2008
Complete List of Authors:	Ahmad, Zainal; Universiti sains Malaysia, School of Chemical Engineering Mat Noor, Rabiatul 'Adawiah; Universiti Sains Malaysia, School of Chemical Engineering Zhang, Jie; Newcastle University, School of Chemical Engineering and Advanced Materials
Keywords:	Neural Networks, Multiple Neural Networks, Nonlinear Process Modeling, Process Control

Multiple Neural Networks Modeling Techniques in Process Control: A Review

Zainal Ahmad[a], Rabiatul 'Adawiah Mat Noor[a], Jie Zhang[b]

[a]*School of Chemical Engineering, University Sains Malaysia, Engineering Campus, Seri Ampangan, 14300 Nibong Tebal, Seberang Perai Selatan, Pulau Pinang, Malaysia.*
Tel: +604-5937788 ext 6462, Fax: +604-5941013
E-mail: chzahmad@eng.usm.my

[b]*School of Chemical Engineering and Advanced Materials, Newcastle University, Newcastle upon Tyne NE1 7RU, UK*
E-mail: jie.zhang@newcastle.ac.uk

ABSTRACT: This paper reviews new techniques to improve neural network model robustness for nonlinear process modeling and control. The focus is on multiple neural networks and also some other techniques that have been reported in the literatures. Single neural networks have been dominating the neural networks "world". Despite of many advantages that have been mentioned in the literature, some problems that can deteriorate neural networks performance such as lack of generalisation has been bothering researchers. Driven by this, neural networks "world" evolves and converges towards better representations of the modeled functions that can lead to better generalization and manages to sweep away all the glitches that have shadowed neural networks applications. This evolution has lead to a new approach in applying neural networks that is called as multiple neural networks. Just recently, multiple neural networks have been broadly used in myriad applications since their performance is literally better than using single neural networks in representing nonlinear systems.

KEYWORDS: Neural Networks, Multiple Neural Networks, Nonlinear Process Modeling, Process Control

INTRODUCTION

Artificial neural networks have been shown to be able to approximate any continuous non-linear functions and have been used to build data base empirical models for non-linear processes [1]. Hence what is a neural network? According to [2].

'A neural network is a massive parallel-distributed processor that has a natural capability for storing experiential knowledge and making it available for use. It resembles the brain in two respects knowledge is acquired by the networks through a learning process. Interneuron connection strengths known as synaptic weights are used to store the knowledge'

Furthermore, the main advantage of neural network based process models is that they are easy to build. This feature is particularly useful when modeling complicated processes where detailed mechanistic models are difficult to develop. However a critical shortcoming of neural networks is that they often lack robustness unless a proper network training and validation procedure is used. Robustness of the model can be defined as one of the baseline to judge the performance of neural network models and it is really related to the learning or training classes as what Bishop [3] described:

'The importance of neural networks in this context is that they offer very special powerful and very general framework for representing non-linear mappings from several input variables to several output variables, where the form of the mapping is governed by a number of adjustable parameters.'

Many factors contributed to the successful research on neural networks and among them the two main factors are as follows. The first one is that neural networks are very powerful modeling tool capable of modeling extremely complex functions [4, 5, 2]. In particular, neural networks are non-linear models, which are very useful in modeling nonlinear systems that cannot be successfully modeled by linear models. The second main factor is that neural networks are easy to use and develop and they basically learn by examples. The neural network users gather representative data, and then invoke a

training algorithm to automatically learn the structure of the data (e.g. [6, 7, 8]). Because of the tremendous capability of neural networks, currently there are a lot of applications of neural networks in industry and business and they are applied in pattern recognition such as automated recognition of hand-written text, finger print identification and moving target on a static background (e.g. [9, 10, 11]). Neural networks have also been used in speech production where a neural network model is connected to a speech synthesizer (e.g. [12, 13]).

Real time control is also a major application area of neural networks with neural network models having been applied in the monitoring and control of complex plants such as chemical plants (e.g. [14, 15]). Neural networks have been employed in business where neural network model have played a role in predicting the stock market trend in certain period of time (e.g. [16, 17]). Another area of applications of neural network models is in signal processing and other typical applications such as noise suppression, filtering and digital signal processing technology (e.g. [18]).

In order to improve the robustness of neural networks a number of techniques have been developed lately like regularization (e.g. [19]) and the early stopping method (e.g. [20]). Ohbayashi [21] implemented the universal learning rule and second order derivatives to increase the robustness in neural network models. Robustness is enhanced by minimizing the change in the values of criterion function caused by the small changes around nominal values of system parameters [21]. Lack of the robustness in individual neural networks is basically due to the overfitting of the models (e.g. [22]).

Overfitting basically refers to the poor generalization of the networks due to fitting the noise in the data (e.g. [23]). Furthermore, the trained network might not minimize the error on the training data set because it has uncontrolled excess dynamics capability or because the training data itself is corrupted with noise [23]. The representation capability of a neural network is determined by its size (number of neurons). If networks are too large they can find many solutions which fit the training set data exactly, but which contain high frequency dynamics is not present in the underlying function. When the data

is corrupted with noise a second form of overfitting occurs. Here the data itself contain high frequencies not present in the underlying function, with the result that minimizing the error on the data set will result in the networks fitting the noise.

Many researchers concentrated on how to increase the robustness of the neural network models either by improving the learning algorithm performance or by improving the generalisation capability of the models. However, single neural networks sometimes lack robustness when the data is insufficient especially when dealing with real world data due to the fact that the robustness of the network is related to the representativeness of the training data [3]. Single neural networks sometimes suffer badly when applied to unseen data where some neural network might fail to deliver the correct result due to the network training converged to undesired local minima, overfitting, or noise in the data (e.g. [24, 23]). Then multiple neural networks are proposed by some researchers to enhance model robustness. This paper is the continuation of the previous review regarding single and multiple neural network modeling techniques at SomChe2007 [25].

MULTIPLE NEURAL NETWORKS MODELS

As mentioned by Willis *et al* [26], more accurate representation of the processes are required to ensure good process control performance especially in Advance Process Control. Therefore neural network models must be robust or stable when they are applied to new (unseen) data.

Even though single neural network models are very powerful non-linear modeling tools, noises in the input data sometimes cause the model to overfit [23]. Overfitting and under fitting is the main problem in developing neural network models. In overfitting, the error on the training data set is driven to a very small value, but when applied to unseen data, the network errors are large and the generalization capability of the neural network is poor. While under fitting is due to that the neural network itself cannot cope with or fails to capture the relationship within the complex data [27]. A single neural network model can be described as a neural network model that utilizes only one neural network model

to represent the system to be modeled. This method however always exhibits some glitches as mention earlier where the model fails to properly represent the function. A single neural network model can be depicted as in Figure 1, which shows a three layer feedforward neural network.

Therefore a lot of techniques have been introduced to improve the generalization capability of neural network models like regularization techniques (e.g. [22, 23, 28]), Bayesian Learning (e.g. [8, 29]) and also by using the parsimonious networks structure [30]. The most exceptional model for this approach is network pruning techniques and sequential orthogonal training techniques. A sequential orthogonal training techniques gradually builds up a neural network model and avoids unnecessarily large networks structure [14]. The idea of multiple neural networks came up from Wolpert [31] where he described about stacked generalisation which is a technique for combining different representations to improve the overall prediction performance. It can also be described as an architecture of network consisting of several sub-models and a mechanism which combines the outputs of these sub-models [32].

Bootstrap Re-sampling

Bootstrap re-sampling or bootstrap technique was first introduce in 1979 as a computer based method for estimating the standard error of empirical distribution [33]. In neural networks bootstrap basically relates or deals with the sampling to create random data sets for training and testing. By creating an equal number of bad and good data sampling, it actually improve the generalization ability because it helps the identification of the characteristic of the scarce class [33]. Zhang [30] demonstrates that sampling by bootstrap does actually increase the robustness of the model and he came up with BAGNET or bootstrap aggregation neural networks [30, 34]. Figure 2 represents bootstrap re-sampling with replacement. In this particular realization, data sample 2 was sampled twice but data sample 6 was not sampled.

Stacked Neural Networks

Zhang [30] mentioned that the individual neural networks are trained using different training data sets and/or from different initial weights, then combined. Instead of choosing the best neural network model among the individual networks, all the neural networks are combined. Sridhar *et al* [35] described the outline of the stacked neural network on how to design and implement the stacked generalization techniques. Wolpert [31] described in detail how the stacked network works. In a glance a stacked neural network model, as shown in Figure 3, contains several networks developed from the original training data set and are referred to as the level-0 models. Then the original data set forms several sub-sets of data during the actual training. Afterward, a level-1 data model needs to be created and is trained using the prediction of level-0 models. The outputs of level-0 models are combined using a level-1 model.

There are several types of multiple neural networks but the underlying ideas are basically similar and the main difference is on how to create the sub-models. Two major types of multiple neural networks are described here.

The first category is multiple model neural networks (e.g. [36, 37]). The training data are totally different in building the individual networks which can be built using different inputs in different regions of operation. The idea of this approach is to adapt different information by using different inputs, and by combining this information a better prediction can be obtained (e.g. [24, 32]). The learning algorithm in each network can also be different and can be supervised or unsupervised methods. Another multiple model approach is introduced by Jacobs *et al* [38] by using the 'mixture of local expert'. Then, Jordan and Jacobs [39] came up with the hierarchical mixture of neural networks. In this case they basically discuss about the supervised learning algorithm and how the divide and conquer method works.

Some examples of multiple model applications are in the field of pattern recognition where different models represent different image classification (e.g. [40, 41, 42]).

Medical application of multiple models is presented by Jerebko *et al* [43] where different classifications of polyps as single neural network models using different inputs are combined and better prediction rate is obtained. It has also been used in other medical fields like in diagnosis application and in detecting the lung cancer [44, 45]. Multiple models have also been applied in time series forecasting [37]. In this case each model forecasts a different time series prediction or prediction horizon and this reduces the recursive prediction promoted to reducing the recursive error occurred in the long range prediction. It also shows that the multiple network model performs better than single networks.

The second category is to creating multiple models using the same training data but re-sampled or partitioned using particular algorithms (e.g. [46, 47]). There are three main algorithms being used to re-sample or partition the training data which are bagging or bootstrap (e.g. [34, 47, 48, 49]), adaboost (e.g. [50, 51]) and randomisation (e.g. [52]). The motivation of creating those different inputs or partitions is to create the effective network ensembles [53]. The bootstrap or bagging basically refers to replication of a training data set where the bootstrap algorithm re-samples the original training data set. Some of the data samples may occur several times, and other may not occur in the sample at all. The individual training sets are independent and the neural networks can be trained in parallel.

Adaboost or 'adaptive boosting' on the other hand constructs a composite classifier by sequentially training classifiers while putting more and more emphasis on certain patterns [51]. The probability distribution over the original training data was maintained in this approach where the network is trained with respect to this distribution. In other words the networks are dependent to each other, while randomization just randomly selects the original training data in each training data and each network can be trained in parallel.

Each technique has it own capabilities or advantages in some applications. For example bagging or bootstrap can generate diverse networks when the base learning algorithm is unstable in that small changes in the training data set cause large changes in the learned

classifiers while boosting can result in less instability. Boosting or Adaboost can make larger changes in the training set like placing large weights on the training set. Based on the experiment conducted by Dietterich [52] randomization method can give quite good performance when the noise level is low in the networks but bagging is still much better when high level of noise is introduced in the networks.

Another method for creating a 'good' ensemble is by adjusting the individual neural networks themselves like varying the set of initial random weights, varying the topology of the networks and also by varying the learning algorithms in the networks (e.g. [53].

The development of computer capability promoted the development of multiple neural networks. Application of multiple neural networks will grow rapidly and become an important component of future research. This is also due to the various neural networks used and combining neural networks is one of the methods for improving the neural network model performance.

COMBINATION OF MULTIPLE NEURAL NETWORKS

Figure 4 delivers a fundamental view of combination of multiple neural networks. Most of the combinations of networks are based on linear combination (e.g. [24,35, 54, 55]). Sharkey [53] also described the methods of combination, which are ensembles, the input data and also modular decomposition methods. Combining the networks improves the generalization capability of the neural networks models in such a way that it guards against the failure of individual components networks. This is because that some of the neural networks will fail to deliver the result or output prediction due to limited training data set (e.g. [23, 24]). In other words, combining a set of imperfect estimators (networks) can be thought of as a way of managing the recognized limitation of the individual estimators, each component is known to make errors, but they are combined in such a way as to minimize the effect of these errors.

Methods for combining multiple networks reported in literature can be divided into linear and nonlinear combinations. The common linear combination is averaging and weighted averaging. The linear combination of multiple outputs is to create a single output as a final prediction. In weighted averaging, individual network outputs are multiplied by appropriate weights and then combined to give the final model prediction. Weighted averaging includes PCR and MLR approaches. Zhang [46] used PCR approach to select the combination weights. Another combination scheme is by Wolpert [31] and it combines the networks with weights that vary over the feature space. The output from a set of level 0 generalisers are used as the input to level 1 generaliser, which is trained to produce the appropriate output.

Nonlinear combination techniques, include Demspter-Shafers belief based method [40], majority voting (e.g. [56]), and also Bayesian model averaging. The Demspter-Shafers belief based method is quite complex and it have to deal with the uncertainty and ignorance of the classifiers. This approach is usually used in model classification or pattern recognition when each network or model represents a character of the image, same as the majority voting combination for example in handwriting recognition (e.g. [57]).

Selective combination of networks has also been proposed. The objective behind selective combination is to reduce the number of shared failure among networks. There are a number of methods on how to select proper networks for combination. For example, Perrone and Copper [58] suggest a heuristics selection method whereby the population of trained networks are ordered in terms of increasing mean squared error and only those with lower sum of squared errors are selected for combination. Hashem [24] also came up with a method which is combination of two alternative selection algorithms: colinearity analysis and cross-validation. The majority voting is one of the selective combination methods where selection is based on the majority of the classifiers or networks that give a 'true' value to the actual image or pattern. Other selective combination approach is by selecting networks that are less correlated before combination using correlation co-efficient analysis (e.g. [59]). This idea is related to the

finding by Rogova [40] that the better result of the combination output is not necessarily based on the combination of 'good' individual networks and combining less accurate and less correlated networks might have a better prediction output. Genetics algorithms can also be used in selecting the networks as what have been done by Wu *et al* [60].

MULTIPLE NEURAL NETWORKS IN MODELING AND CONTROL APPLICATIONS

Single neural network applications in modeling and control

As mentioned earlier in the previous section that single neural networks have been widely used not only in the engineering filed but also in other applications like in remote sensing (e.g.[61]), transportation, power system (e.g. [62]), medicine (e.g. [63, 64]), telecommunication, banking and also application in robotics and vision techniques [65]. The growing interest in applying single neural networks are due to the computing system that growth rapidly which enable the behavior of the complex system to be modeled and predicted accurately. Furthermore the characteristic of neural network models themselves that learn from examples rather than having to program the complex system also contributed the application of the models. The architecture of single neural networks vary from multilayer perceptron to radial basis function (RBF) and also recurrent neural networks models (e.g. [66]).

Most of the applications of neural networks in chemical engineering are concentrated on the modeling and control of chemical processes using multilayer perceptron networks. The common systems used in the chemical processes are distillation columns, and reactor systems (continuous stirred tank reactor (CSTR), bioreactor, and neutralizing reactor). These processes are usually very nonlinear and nonlinear models have to be developed. Currently, applications of single neural networks in process modeling and control are quite significant in industry especially in model based predictive control (MBPC) (e.g. [67, 68]) and this is due to the ability of neural networks in modeling nonlinear processes (e.g. [69]).

In process modeling, single neural networks have been applied in numerous applications, for example, Aldrich and Slater [70] model the fractional hold-up and drop size in a reactor, Xiong and Jutan [68] developed a model to predict the heat released by a chemical reactor as well as Aziz *et al* [71]. Other research in chemical reaction in CSTR for examples was done by Shaw *et al* [69] where single neural networks have been used to model the reactor temperature and the result was quite convincing. In bioprocess Lobanov *et al* [72] developed a model where single neural networks are used as a biosensor to predict the glucose and ethanol in certain range of substrate. Scheffer and Filho [73] applied single neural networks with the extended Kalman filter in the training to predict the production of the penicillin in a batch process. Other applications of neural networks were reported by Lennox *et al* [74] where single neural networks have been used to model a vitrification process using real world data.

In process control, there have been many applications of single neural networks and they can be classified into three major categories of control: model predictive control, inverse-model based control and adaptive control. For example Willis *et al* [26] implemented model predictive control in a CSTR using single neural network models to control the output concentration. Zhan and Ishida [75] implemented the multi-step-ahead prediction model in NMPC to control the product concentration in a CSTR. Chen and Yea [67] also implemented a multi-step-ahead prediction model using a single neural network in a CSTR neutralization process and the control performance was obtained. While in fed-batch processes, single neural networks have also been used in predicting the future values of the process output for optimization. Kovarova-Kovar *et al* [76] applied a single neural network in NMPC to optimize the production of the riboflavin in fed-batch processes. Other implementations of NMPC used recurrent neural networks, for example, Zamareno and Vega [77] applied recurrent neural network based NMPC to a very nonlinear suphitation process. Zhang and Morris [78, 79] implemented GPC using recurrent neuro-fuzzy model to model and control the tank level in a conic tank and pH in a neutralization process in CSTR.

The most popular control approach in inverse model based strategy is internal model control (IMC). Shaw *et al* [69] use dynamic feedforward neural networks (FANN) and recurrent neural networks (RNN) to model the temperature of a reactor. Hunt and Sbarbaro [80] utilized the IMC approach to control the pH in CSTR. On the other hand, Hussain [81] modified the IMC model using a single neural network to include an adaptive scheme using sliding windows in a fermentation process. In indirect adaptive control schemes, neural networks are used to identify an unknown nonlinear plant online. For examples Calise *et al* [82] implemented the adaptive control in the van der Pol oscillator and the controller performed well. Lightbody and Irwin [83] used a neural network in parallel with a fixed gain linear controller in direct model-reference adaptive control configuration to control the product concentration in a CSTR. Boslovic and Narendra [84] applied both the FANN and RBF in adaptive control schemes for a baker yeast fermentation process.

From the above paragraphs, it is hard to refuse the broadness applications of single neural networks. Instead of enormous number of applications, there are still some drawbacks that should be avoided in order to achieve maximum model accuracy and robustness and in fact there are ways if not to make them totally vanished but to suppress them. Combining neural networks is declared to be the way to suppress the drawbacks of single neural networks or it is known as multiple neural networks. A review of multiple neural networks applications in process modeling and control is presented next.

Multiple neural network applications in modeling and control

Multiple neural network applications in control especially in NMPC are quite new compared to applications of single neural networks (e.g. [85]). Chen and Narendra [36] implemented what they call intelligent control where they applied multiple models on their controller. They designed the controller based on the different models and models can be switched when appropriates. Multiple neural networks have also been applied in adaptive control where the weighted sum of the multiple neural networks is used to approximate the system nonlinearity of the given task [86]. It is shown that multiple

neural network models performed better than conventional artificial neural networks. Zhang [87] proposed a multiple neural network based reliable optimal control strategy for a batch polymerization process. This technique shows some good result in the simulation study. He also introduced multiple neural networks using bootstrap re-sampling technique to predict polymer quality in batch polymerization reactor. He mentioned that neural networks trained on different bootstrap re-sampled data sets would be more dissimilar than those trained on the same training data. When trained with bootstrap re-sampled data set, different neural networks will perform differently in different regions of the input space. Although these neural networks are correlated since they intend to model the same relationship, the independent elements among these models can be discovered through principal component analysis. Neural network prediction confidence bounds can also be obtained using the bootstrap technique. Model prediction confidence bounds give process operators extra information on how confident a particular prediction is [46]. Process operators can accept or reject a neural network prediction based upon the estimated confidence bounds. He also employed principal component regression (PCR) to determine the appropriate weights for the combined neural networks.

Sridhar et al. [88] further approves this matter by modeling chemical processes using stacked neural networks. The stacked neural networks have been applied and evaluated for three example problems including the dynamic modeling of a nonlinear chemical process. As expected, this method never failed to give a convincing result. Another contribution by Sridhar et al. [55] in proving the superiority of multiple neural networks is by using the stacked neural networks together with information theoretic stacking (ITS) algorithm. This algorithm was used to combine neural network models. The ITS algorithm identifies and combines useful models regardless of the nature of their relationship to the actual output. This method was utilized in three examples including a dynamic process modeling problem.

Eikens and Karim [32] implemented multiple neural network models through multiple models "division" using linear combination. They presented a flexible framework which allows the integration of the other model paradigms. These models were applied in

process identification of a fermentation process. Three different methods for constructing multiple neural network models are employed. They are prior knowledge, unsupervised learning and gating neural network. As expected, the results always in a satisfactory condition where improved modeling performance can be seen through the results. Jazayeri-Rad [15] also has emerged with an idea of using multiple neural networks together with nonlinear model predictive control for modeling a chemical plant. Two examples were tested using this model that was a simple MIMO system and also a multi-component distillation column. Simulation results demonstrate the ability of the proposed strategy to outperform the MPC algorithms based on the linear model of the plant.

By implementing multiple neural networks in control application especially in the batch polymerization case, the relationship between batch recipes and polymerization trajectories can be learnt avoiding the development of an intricate polymerization kinetic model.

The difficult-to-measure variables such as molecular weight are related to certain easy-to-measure variables such as temperatures in the reactor. With this relationship, inferential estimation of these difficult-to-measure variables can be obtained from the measurements of the easy-to-measure variables. Empirical models can be developed from process operation data. Since polymerization processes are highly non-linear processes, non-linear empirical models should be developed. One of the advantages of multiple neural network based modeling is that a complex non-linear process model can be developed from process data only [46].

Therefore, the problem such as numerical integration for a large number of complex differential equation can be avoided [30]. Another application of multiple neural network is the estimation of impurities and fouling in batch polymerization reactors [89]. In this paper, Zhang et al. introduced two approaches where first approach an inverse neural network model of the polymer process is constructed and the initial reaction conditions are predicted while in the second approach a neural network is used to model the dynamic behavior of the polymer process. The inverse model was developed using multiple neural

network model i.e. stacked neural networks. From the trajectory of the polymerization process, the neural network model estimates the effective initial initiator weight and the effective heat transfer coefficient. The amount of impurities was calculated to be the difference between the gross initial initiator weight and the estimated effective initial initiator weight. The amount of fouling was defined as the difference between the nominal and estimated heat transfer coefficient.

Meanwhile in the second approach, a dynamic neural network was used to predict the polymerization trajectory from the initial conditions. Impurities and fouling were detected when the predicted trajectory deviates from the observed trajectory. The amount of impurities and fouling was then estimated using an optimization procedure which minimizes the difference between the predicted trajectory and the observed trajectory. The predicted trajectories are then compared with the on-line measurements of conversion and coolant temperatures [89].

Ahmad and Zhang [90] propose a network combination method using data fusion technique. In this paper, neural networks are also built based on bootstrap re-sampling of original data and are combined using data fusion technique. A proper model is selected at each sampling time using Bayesian inference approach. In the proposed approach, multi-sensor data fusion is applied in decision level identity fusion to the combination of multiple neural networks [90]. In essence, it is Bayesian Combination Predictor (BCP) with some modification to identity declaration in data fusion technique. The proposed method has been applied in modeling of reactant concentration in irreversible exothermic reaction process, pH neutralization process and real world data for water discharge in Langat River and they are proved to be better compare to averaging of all networks and combining all networks using Bayesian combination method. The architecture of Bayesian Inference is depicted in Figure 5.

Bayesian selective combination of multiple neural networks was applied by Ahmad and Zhang [91] for improving long-range predictions in nonlinear process modeling. Instead of using fixed combination weights, the probability of a particular network being the true

model is used as the combination weight for combining the networks. Selective combination aims to achieve maximum generalization capability by combining selected individual networks. The results demonstrate that the proposed techniques unbelievably improve model generalization and perform better than aggregating all the individual networks.

Fault diagnosis becomes one of the popular research areas recently due to its vitality in running and operating a good and safe plant. Multiple neural networks have been used as one of the eminent tools for pattern recognition as fault diagnosis is one of the branches in pattern recognition "tree". Once again Zhang [92] took a full advantage of multiple neural networks to develop a model for improving online fault diagnosis through information fusion. By employing multiple neural networks instead of single neural networks, the system can trigger earlier and more reliable warning plus earlier diagnosis for the occurrence of incipient fault. In his paper, multiple neural networks are developed and their diagnosis results are combined to give the overall results. In order to develop a diverse range of individual networks, each of them is trained on a replication of the original training data generated through bootstrap re-sampling with replacement [92]. Three combination methods have been applied to the model, averaging and weighted averaging, major voting and also modified major voting. According to the study, modified major voting combination scheme give the best performance for the system.

Zhang [93] has introduced a new method of controlling batch polymerization process by using batch to batch control together with stacked neural networks for modeling the system. Towards the mission to overcome the difficulties of developing mechanistic model, stacked neural network models are developed from the process operational data. Batch process possesses a nature of repetitive process and it is in fact where the idea of using batch to batch controlling method as a new and improved method using the information from current and previous batch run. This is the way it works. The neural network model is linearized around the current batch and based on the linearized model the control policy for the next batch is modified to minimize the control errors at the end of the next batch [93]. This procedure is repeated from batch to batch. Application to a

simulated batch polymerization reactor demonstrates that the proposed method can enhance process performance from batch to batch in the presence of model plant mismatches and unknown disturbances.

Tian et al. [94] employed hybrid stacked recurrent neural network model for a batch polymerization process. They claimed that the hybrid model contains a simplified mechanistic model that does not consider the gel effect and stacked recurrent neural networks. Stacked recurrent neural networks on the other hand were built to characterize the gel effect which was perceived as one of the most difficult parts of polymerization modeling. The results were compared with a best-single-network-based hybrid model. It was proved that control policy based on the hybrid stacked recurrent neural network model performed reliably on the real process. Meanwhile, Perrone and Cooper [58] utilized a neural network ensemble via GEM estimator on the NIST OCR database. By applying averaging in functional space, they triumphantly constructed a neural network model which is guaranteed to have improved performance. Hashem [24] in his paper proposed optimal linear combinations (OLC) of neural networks. He adduced that combining the trained neural networks may help integrate the knowledge acquired by the components networks thus improve model accuracy. He also discussed about the harmful collinearity which can deteriorate the model's performance. His optimal linear combinations of neural networks had been tested on various algorithms and certified to significantly improve model accuracy.

The fact that they have surpassed single neural networks performance has been stamped as a new evolution in neural network application. Multiple neural networks have been utilized not only in modeling and controlling chemical engineering related processes but in any other processes and systems such as biology, agriculture, hydrology, etc.

Wanga *et al* [95] applied multiple neural networks for prediction of membrane protein types based on pseudoamino acid. He mentioned that they are several identifiers that have been developed such as support vector machine (SVM), covariant discriminant (CD), artificial neural network (ANN) and *k*-nearest neighbor (KNN) classifier but still the way

they operate are basically individual. In view of this, stacked generalization has been chosen as the method for classification task. Stacking approach can combine several different types of classifiers through a meta-classifier to maximize the generalization accuracy and it is anticipated to be able to improve identification quality of the protein classes.

Sharkey [96] is one of the eminent name in multiple neural networks has come up with an idea of applying multiple neural networks for a fault diagnosis of a diesel engine. This system was designed to provide an early warning of combustion-related faults in a diesel engine. He used four different sets of data, NEF, NE, NF and NE for training purpose. Modules corresponding to these different sets of data were then assembled. The combined modules were then created the multiple neural networks system capable of providing a solution to complete fault diagnosis problem. The system is shown to outperform a decision-tree algorithm and a human expert; comparisons which show the complexity of the required discrimination and it is depicted in Figure 6.

Jia and Culver [97] applied bootstrap neural networks in synthetic flow generation. Hydrological calibration of mechanistic watershed simulation models often requires several years of continuous flow data. Unfortunately, historical flow information is highly limited for many ungauged or recently gauged watersheds. Synthetic flow generation methods could be used to extend the available flow records at data-limited watersheds and to create a statistically reasonable synthetic flow series as a target for hydrological calibration. The bootstrap method is used to estimate the generalization errors of neural networks with different structures and to construct the confidence intervals for each flow prediction. In a Total Maximum Daily Load (TMDL) study, a continuous mechanistic watershed simulation model can play a key role, providing a means to describe the relationships between pollutant sources, load allocation plans, and water quality

However, some important issues related to this synthetic flow generator require further investigation. First, the accuracy of the generated synthetic flows is a major concern, and

a more rigorous understanding of the uncertainty of the synthetic flows will help the analyst to estimate the uncertainty of the calibrated model parameters. Second, under the condition of a small data set, the cross-validation approach may not be the best approach for the modeling of ANNs. To address the above problems, this study investigates a bootstrapped ANN (BANN) approach for synthetic flow generation. Based on the result of the research, bootstrap neural networks outperform the other models such as maintenance of variance extension (MOVE) and modified drainage area ratio (DAR). The summary of multiple neural networks application in various fields can be seen in Table 1.

CONCLUSIONS

From day to day, a continuously increasing number of people interested into joining the exciting research about neural networks have been one of the causes of broad , extensive and continuing application of neural networks in many chemical processes, both modeling and control. There are myriad applications nowadays that are using neural networks as part of their modeling and control process. These are due to convincing results for both research and real application using neural networks as the tool in modeling and control applications. Moreover, neural networks also posses an ability of embedding into other control scheme such as fuzzy, inverse model and adaptive control so that they can perform better than using solely fuzzy or inverse model or adaptive control. Such circumstance allows us to venture into a real broad application of neural networks both for neural networks and neural networks along with the other control schemes. Despite of their enormous advantages, neural networks never been "born" perfectly. Drawbacks such as overfitting, underfitting and also problem on how to improve the generalization capability of neural networks have forced us to seek for a better solution. Multiple neural networks seem to be the best solution for the time being. Stacked and bootstrap methods are the premier ways to apply multiple neural networks in modeling and controlling dynamic systems which proved to be effective to overcome the drawbacks of single neural networks. As a conclusion, these reviews were also revealing several other points about multiple neural networks application such as:

1. Lack of data during training stage can deteriorate the performance of neural network model but can be alleviate by applying multiple neural networks.
2. Using bootstrap technique, the confidence bound of neural networks can be obtained.
3. Multiple neural networks model can overcome the problem of developing complex and uncertain mechanistic model to represent complex processes.

Acknowledgements

This work was supported by the Universiti Sains Malaysia (USM) for funding the work under short term grant 6035253 and 6035182.

REFERENCES

[1] Hertz J., A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.

[2] Haykin S., *Neural Networks*, Macmillan College Publicating Company, 1994.

[3] Bishop C., *Neural Networks for Pattern Recognition*, Clarendon Press, 1995.

[4] Grossberg S., *Neural Networks and Natural Intelligence*, MIT Press, 1988.

[5] Hinton G. E., How Neural Networks Learn from Experience, 1992, *Scientific American*, 267, 144-151.

[6] English E. M., Stacked Generalisation and Simulated Evolution, 1996, *BioSystem,* 39, 3-18.

[7] Chen K., L. Xu, H. Chi, Improved Learning Algorithm for Mixture of Expert in Multiclass Classification, 1999, *Neural Networks,* 12, 1229 – 1252.

[8] Crucianu M., R. Bone, J. P. Asselin de Beauville, Bayesian Learning for Re-current Networks, 2001, *Neurocomputing*, 6, 235-242.

[9] Srinivas G., M. R. Kabuka, Design of Supervised Classifiers Using Boolean Neural Network, 1995, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(12), 1239-1246.

[10] Seong-Whan L., Off-line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Networks, 1996, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(6), 648-652.

[11] Chen W., P. D. Gader, H. Shi, Improved Dynamic Programming based Handwritten word Recognition using Optimal Order Statistic, *In Proceeding of the International Conference Statistical and Stochastic Methods in Image Processing II*, 1997, 245-256.

[12] Baig A.R., R. Séguier, G. Vaucher, A Spatio-temporal Neural Network Applied to Visual Speech Recognition, *In Proceedings of International Conference on Artificial Neural Networks*, 1999, 797- 802.

[13] Furlanello C., D. Giuliani, S. Merler, E. Trentin in *Model Selection of Combined Neural Networks Nets for Speech Recognition, Combining Artificial Neural Nets Ensemble and Modular* (Eds: Amanda J. C. Sharkey), Springer Publication London, 1999.

[14] Zhang J., A. J. Morris, E. B. Martin, Long-term Prediction Models Based on Mixed Order Locally Recurrent Neural Networks, 1998a, *Computers and Chemical Engineering*, 22(7-8), 1051-1063.

[15] Jazayeri-Rad H., The Nonlinear Model-Predictive Control of a Chemical Plant Using Multiple Neural Networks, 2004, *Neural Computing and Applications,* 13, 2-15.

[16] Fletcher D., E. Goss, Forecasting with Neural Networks: An Application Using Bankruptcy Data, 1993, *Information and Management*, 24, 159-167.

[17] Desai V.S., R. Bharati, The Efficiency of Neural Networks in Predicting Returns on Stock and Bond Indices, 1998, *Decision Sciences*, 29(2), 405-425(1998).

[18] Larsson L., S. Krol, K. Lagemann, NeNEB-An Application Adjustable Single Chips Neural Networks Processor for Mobile Real Time Image Processing, *International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing,* 1996, 154-163.

[19] Girosi F., M. Jones, T. Poggio, Regularisation Theory and Neural Networks Architecture, 1995, *Neural Computation,* 7, 219-269(1995).

[20] Morgan N., H. Bourlard in *Generalisation and Parameter Estimation in Feedforward Nets: Some Experiments* (Ed: D. S. Touretzkey), In Advances in Neural Information Processing System 2, San Mateo CA, 1990, pp. 630-637.

[21] Ohbayashi M., K. Hirasawa, K. Toshimitsu, J. Murata, J. Hu, Robust Control for Non-linear System by Universal Learning Networks Considering Fuzzy Criterion and Second Order Derivatives, *In Proceeding IEEE World Congress on Computational Intelligence,* 1998.

[22] Caruana R., S. Lawrence, C. Lee Giles, Overfitting in Neural Networks: Backpropagation, Conjugate Gradient and Early Stopping, 2000, *Neural Information Processing System,* 13, 402-408.

[23] Mc Loone S., G. Irwin, Improving Neural Networks Training Solution Using Regularisation, 2001, *Neurocomnputing*, 37, 71-90(2001).

[24] Hashem S., Optimal Linear Combination, 1997, *Neural Networks*, 10(4), 599-614.

[25] Ahmad Z., Mat Noor R. A., A Review of Neural Network Modeling Technique: Multiple and Single Neural Networks, *In Proceeding of Symposium of Malaysian Chemical Engineer (SomChe),* 2007.

[26] Willis M. J., C. Di Massimo, G. A. Montaque, M. T. Tham, A. J. Morris, Artificial Neural networks in Process Estimation and Control, 1992, *Automatica*, 28(6), 1181-1187.

[27] Guyon X., J. Yao, On the Underfitting and Overfitting Sets of Model Chosen by Order Selection Criteria, 1999, *Journal of Multivariate Analysis*, 70, 221-249.

[28] Hagiwara K., K. Kuno, Regularisation Learning and Early Stopping in Linear Networks, 2000, *International Joint Conference on Neural Networks (IJCNN)*, 511-516.

[29] Radford M. N., Bayesian Learning for Neural Networks, 1997, *Statistics*, 118-183.

[30] Zhang J., A. J. Morris, E. B. Martin, C. Kiparissides, Prediction of Polymer Quality in Batch Polymerization Reactors Using Robust Neural Networks, 1998b, *Chemical Engineering Journal,* 69, 135-143.

[31] Wolpert D. H., Stacked Generalisation, 1992, *Neural Networks,* 5, 241-259.

[32] Eikens B., M. N. Karim, Process Identification with Multiple Neural Networks Models, 1999, *International Journal of Control,* 72(7/8), 576-590.

[33] Dupret G., M. Koda, Theory and Methodology Bootstrap Re-sampling for Unbalanced Data in Supervised Learning, 2001, *European Journal of Research,* 134, 141-156.

[34] Zhang J., Developing Robust Non-linear Models Through Bootstrap Aggregated Neural Networks, 1999a, *Neurocomputing,* 25, 3-113.

[35] Sridhar D. V., E. B. Bartlett, R. C. Seagrave, Process Modelling Using Stacked Neural Networks, 1996, *AIChe Journal*, 42(9), 2529-2539.

[36] Chen L., K. S. Narendra, Intelligent Control using Multiple Neural Networks and Multiple Models, 2001, *Automatica*, 37, 1245-1255.

[37] Nguyen H. H., C. W. Chan, Multiple Neural Networks for a Long Term Time Series Forecast, 2004, *Neural Computation and Applications,* 13, 90-98.

[38] Jacobs R. A., M. I. Jordan, S. J. Nowlan, G. E. Hinton, Adaptive Mixture of Local Expert, 1991, *Neural Computation*, 3, 79-87.

[39] Jordan M. I., R. A. Jacobs, Hierarchical Mixtures of Expert and the EM Algorithm, 1994, *Neural Computation*, 6, 191-214.

[40] Rogova G., Combining the Results of Several Neural Networks Classifiers, 1994, *Neural Networks*, 7(5), 777-781.

[41] Giacinto G., F. Roli, Design an Effective Neural Networks Ensembles for Image Classification Purposes, 2001, *Image and Vision Computing*, 19, 699-707.

[42] Cho S. B., J. H. Lee, Learning Neural Networks Ensembles for Practical Text Classification, 2003, *Lectures Notes in Computer Science*, 2690, 1032-1036.

[43] Jerebko A. K., J. D. Malley, M. Franaszek, R. M. Summers, Multiple Neural Networks Classification Schemes for Detection of Colonic Polyps in CT Colonography Data Set, 2003, *Academic Radiology,* 10(2), 154-160.

[44] Hayashi Y., R. Setiono, Combining Neural Network Prediction for Medical Diagnosis, 2002, *Computer in Biology and Medicine, 32*, 237-246.

[45] Zhou Z. H., Y. B. Jiang, S. F. Chen, Lung Cancer Cell Identification Based on Artificial Neural Networks Ensembles, 2002, *Artificial Intelligent in Medicine*, 24, 26-36.

[46] Zhang J., Inferential Estimation of Polymer Quality Using Bootstrap Aggregated Neural Networks, 1999b, *Neural Networks*, 12, 927-938.

[47] Cunningham P., J. Carney, S. Jacob, Stability Problem with Artificial Neural Networks and the Ensembles Solutions, 2000, *Artificial Intelligence in Medicine*, 20, 217-225.

[48] Brieman L., Bagging Predictors (Tech.Rep.No.421), Department of Statistic University of California at Berkeley, USA, 1994.

[49] Wehrens R., H. Putter, L. M. C. Buyden, the Bootstrap: A Tutorial, 2000, *Chemometrics and Intelligent Laboratory System*, 54, 35-52.

[50] Freund Y. and R. E. Schapire, Experiment with a New Boosting Algorithm, *In Proceeding 13th International Conference on Machine Learning*, 1996.

[51] Schwenk H. and Y. Bengio, Boosting Neural Networks, 2000, *Neural Computation*, 12, 1869-1887.

[52] Dietterich T. G., An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, 2000, *Machine Learning*, 40(2), 139 – 157.

[53] Sharkey A. J. C., *Multi Nets System*, (Ed: A. J. C. Sharkey), In Combining Artificial Neural Nets Ensemble and Modular, London, Springer Publication, 1999.

[54] Hashem S., *Treating Harmful Collinearity in Neural networks Ensembles*, Combining Artificial Neural Nets Ensemble and Modular, (Ed: Amanda J.C Sharkey), Springer Publication London, 1999.

[55] Sridhar D. V., E. B. Bartlett, R. C. Seagrave, An Information Theoretic Approach for Combining Neural Network Process Models, 1999, *Neural Networks*, 12, 915-926.

[56] Battiti R., M. Colla, Democracy in Neural Net: Voting Schemes for Classification, 1997, Neural *Networks*, 7(4), 691-707.

[57] Xu L., A. Krzyzak, C. Y. Suen, Methods for Combining Multiple Classifiers and their Application to Handwriting Recognition, 1992, *IEEE Transaction on System, Man, and Cybernetics*, 22(3), 418-435.

[58] Perrone M. P., L. N. Cooper, *When Networks Disagree: Ensembles Methods for Hybrid Neural Networks*, (Ed: R. J. Mammone), In Artificial Neural Networks for Speech and Vision, London, Chapman and Hall, 1993; pp. 126-142.

[59] Ahmad Z. and J. Zhang, Selective Combination of Multiple Neural Networks for Improving Long Range Prediction in Nonlinear Process Modelling through Correlation Co-efficient Analysis, *In Proceeding International Conference on Robotics, Vision, Information and Signal Processing*, 2003.

[60] Wu J. X., Z. H. Zhou, Z. Q. Chen, Ensembles of GA based Selective Neural Networks Ensembles, *In Proceeding 8th International Conference on Neural Information Processing*, 2001.

[61] Hussein M. A., Review of the Application of Neural Networks in Chemical Process Control-Simulation and Implementation, 1999, *Artificial Intelligence in Engineering*, 13, 55-68.

[62] Kiartzis S., A. Kehagias, A. Bakirtzis, A. Petridis, Short Term Load Forecasting Using a Bayesian Combination Method, 1997, *Electrical Power and Energy System*, 19(3), 171-177.

[63] Lo S. C. B., J. S. J. Lin, M. T. Freedman, S. K. Mun, Application of Artificial Neural networks to Medical Image Pattern Recognition: Detection of Clustered Microcalcification on Mammograms and Lung Cancer on Chest Radiographs, 1998, *Journal of VLSI Signal Processing*, 18, 263-274.

[64] Brameier M. and W. Banzhaf, A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining, 2001, *IEEE Transaction on Evolutionary Computation,* 5(1), 17-26.

[65] Pham D. T. and X. Liu, Neural Networks for Identification, *Prediction and Control*, Springer, 1995.

[66] Hagan M. T., H. B. Demuth, O. De Jesus, An Introduction to the used of Neural Networks in Control System, 2002, *International Journal of Robust and Nonlinear Control*, 12, 959-985.

[67] Chen J. and Y. Yea, Neural Network-Based Predictive Control for Multivariables Processes, 2002, *Chemical Engineering and Communication*, 189, 866-894.

[68] Xiong Q. and A. Jutan, Grey-Modelling and Control of Chemical Processes, 2002, *Chemical Engineering Science*, 57, 1027-1039.

[69] Shaw A. M., F. J. Doyle III, J. S. Schwaber, A Dynamic Neural Networks Approach to Nonlinear Process Modelling, 1997, *Computer and Chemical Engineering*, 21(4), 371-385.

[70] Aldrich A., M. J. Slater, *Simulation of Liquid-Liqud Extraction Data with Artificial Neural Networks*, (Eds: I. M. Mujtaba and M. A. Hussain), In Application of Neural Networks and Other Learning Technologies in Process Engineering, London, Imperial College Press, 2001; pp. 3-22.

[71] Aziz N., I. M. Mujtaba, M. A. Hussain, *Set Point Tracking in Batch Reactors: Use of PID and Generic Model Control with Neural Networks*, (Eds: I. M. Mujtaba and M. A. Hussain), In Application of Neural Networks and Other Learning Technologies in Process Engineering. London, Imperial College Press, 2001, 217-241.

[72] Lobanov A.V., I. A. Borisov, S. H. Gordon, R. V. Greene, T. D. Leathers, A. N. Reshetilov, Analysis of Ethanol-Glucose Mixtures by Two Microbial Sensors: Applications of Chemometrics and Artificial Neural networks for Data Processing, 2001, *Biosensor and Bioelectronics*, 16, 1001-1007.

[73] Scheffer R. and M. Filho, *Process Identification of a Fed-batch Penicillin Production Process-Training with the Extended Kalman Filter*, (Eds: I. M. Mujtaba and M. A. Hussain), In Application of Neural Networks and Other Learning Technologies in Process Engineering. London, Imperial College Press, 2001, pp. 99-118.

[74] Lennox B., P. Rutherford, G. A. Montaque, C. Haughin, Case Study Investigating the Application of Neural Networks for Process Modelling and Condition Monitoring, 1998, *Computers and Chemical Engineering*, 22(11), 573-1579.

[75] Zhan J. and M. Ishida, The Multi-Step Predictive Control of Nonlinear SISO Processes with a Neural Models Predictive Control (NMPC) Method, 1997, *Computer and Chemical Engineering*, 21(2), 201-210.

[76] Kovarova-Kovar K., S. Gehlen, A. Kunze, T. Keller, R. von Daniken, M. Kolb, A. P. G. M van Loon, Application of Model-Predictive Control Based on Artificial Neural Networks to Optimize the Fed-Batch Process for Riboflavin Production, 2000, *Journal of Biotechnology*, 79, 39-52.

[77] Zamareno J. M. and P. Vega, Neural Predictive Control Application to a Highly Nonlinear System, 1999, *Engineering Applications of Artificial Intelligence*, 12, 149-158.

[78] Zhang J. and A. J. Morris, Recurrent Neuro-Fuzzy Networks for Nonlinear Process Modelling, 1999, *IEEE Transaction of Neural Networks*, 10(2), 313-326.

[79] Zhang J. and A. J. Morris, Long Range Predictive Control of Nonlinear Processes Based on Recurrent Neuro-Fuzzy Network Models, 2000, *Neural Computation and Applications*, 9, 50-59.

[80] Hunt K. J. and D. Sbarbaro, *Studies in Neural Networks Based Control*, (Eds: K. Warwick, G. W. Irwin and K. J. Hunt), In Neural Networks for Control and System, London, Peter Peregrinus Ltd, 1992, pp. 96-122.

[81] Hussain M. A., *Adaptive Inverse Model Control of a Continuous Fermentation Process using Neural Networks*, (Eds: I. M. Mujtaba and M. A. Hussain), In Application of Neural Networks and Other Learning Technologies in Process Engineering, London, Imperial College Press, 2001, pp. 199-216.

[82] Calise A. J., N. Hovakimyan, M. Idan, Adaptive Output Feedback Control of Nonlinear System using Neural Networks, 2001, *Automatica*, 37, 1201-1211.

[83] Lightbody G. and G. W. Irwin, Direct Neural Model Reference Adaptive Control, *In Proceeding IEE on Control Theory Application*, 1995.

[84] Boslovic J. D. and K. S. Narendra, Comparison of Linear, Nonlinear and Neural Networks Based Adaptive Controller for a Class of Fed-Batch Fermentation Process, 1995, *Automatica*, 31(6), 817-840.

[85] Chen L. and K. S. Narendra, Nonlinear Adaptive Control using Neural Networks and Multiple Models, 2001, *Automatica*, 37, 1245-1255.

[86] Lee C. Y. and J. J. Lee, Adaptive Control for Uncertain Nonlinear System Based on Multiple Neural Networks, 2004, *IEEE Transaction on System, Man, and Cybernetics*, 34(1), 325-333.

[87] Zhang J., A Reliable Neural Network Model Based Optimal Control Strategy for a Batch Polymerization Reactor, 2004, *Ind. Eng. Chem. Res*, 43, 1030-1038.

[88] Sridhar D. V., Seagrave R. C., Bartlett E. B., Process Modeling Using Stacked Neural Networks, 1996, *AIChe Journal*, 42(9), 2529-2539.

[89] Zhang J., A. J. Morris, E. B. Martin, C. Kiparissides, Estimation of Impurities and Fouling in Batch Polymerization Reactors through the Application of Neural Networks, 1999c, *Computers and Chemical Engineering*, 23, 301-314.

[90] Ahmad Z. and Zhang J., Combination of Multiple Neural Networks Using Data Fusion Techniques For Enhanced Nonlinear Process Modeling, 2005, *Computers and Chemical Engineering,* 30, 295–308.

[91] Ahmad Z. and Zhang J., Bayesian Selective Combination of Multiple Neural Networks for Improving Long-Range Predictions In Nonlinear Process Modelling, 2005, *Neural Computing and Applications*, 14, 78-87.

[92] Zhang J., Improved Online fault Diagnosis through Information Fusion in Multiple Neural Networks, 2006, *Computers and Chemical Engineering*, 30, 558–571.

[93] Zhang J., Batch-To-Batch Optimal Control of a Batch Polymerisation Process Based On Stacked Neural Network Models, 2008, *Chemical Engineering Science*, 63, 1273-1281.

[94] Tian Y., Zhang J., Morris J., Modeling and Optimal Control of a Batch Polymerization Reactor Using a Hybrid Stacked Recurrent Neural Network Model, 2001, *Ind. Eng. Chem. Res.*, 40, 4525 – 4535.

[95] Wanga S. Q., Yanga J., Chou K. C., Using Stacked Generalization to Predict Membrane Protein Types Based On Pseudo-Amino Acid Composition, 2006, *Journal of Theoretical Biology*, 242, 941–946.

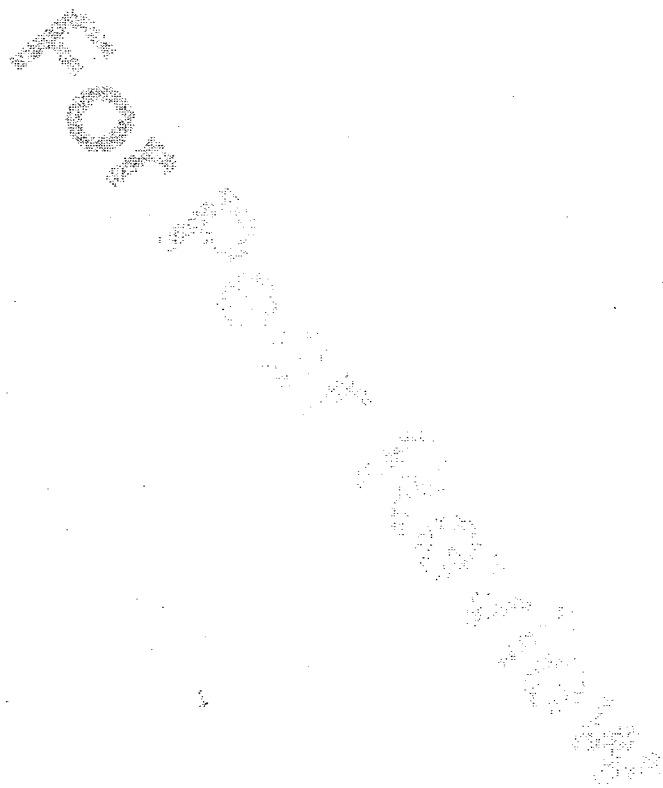
[96] Sharkey A. J. C., Chandroth G. O., Sharkey N. E., A Multi-Net System for the Fault Diagnosis of a Diesel Engine, 2000, *Neural Computing and Applications*, 9, 152-160.

[97] Jia Y. and Culver T. B., Bootstrapped Artificial Neural Networks for Synthetic Flow Generation with a Small Data Sample, 2006, *Journal of Hydrology*, 331, 580– 590.

Table 1. Summary of Multiple Neural Networks Applications

Author	Year	Multiple Neural Networks		Combination	System
		Multiple Models	Data Re-sampling		
Perrone and Cooper	1992	Yes		Linear Combination (Averaging)	Pattern Recognition (NIST OCR database)
Rogova	1994	Yes		Nonlinear Combination (Dempster-Shafer)	Pattern Recognition (NIST OCR database)
Sridhar et al.	1996	Yes		Linear Combination (Stacked Generalization)	Exothermic reaction in CSTR
Hashem	1997	Yes		Linear Combunation (Weight Averaging)	Multiple algorithms
Eikens and Karim	1999	Yes		Linear Combination (Weight Averaging)	Fermentation Process
Sridhar et al.	1999	Yes		Nonlinear Combination (ITS algorithm)	Exothermic reaction in CSTR

Zhang	1999		Yes		Batch Polymerization Reactor
Zhang	1999	Yes		Neural Network	Batch Polymerization Reactor
Sharkey et al.	2000	Yes			Pattern Recognition (Fault Diagnosis for Diesel Engine)
Tian et al.	2001	Yes		Linear Combination (Weight Averaging)	Batch Polymerization Reactor
Chen and Narendra	2001	Yes		Linear Model Neural Network	Chemical Plant
Jazayeri-Rad	2004	Yes			Distillation Column
Zhang	2004		Yes	Linear combination	Batch Polymerization Reactor
Ahmad and Zhang	2005		Yes	Linear combination	pH Neutralization Process, Reactant Concentration in An Irreversible Exothermic Reaction Process
Ahmad and Zhang	2005		Yes	Linear combination	Exothermic Reaction Process, pH Neutralization Process, River's Water Discharge
Zhang	2006		Yes	Linear combination	Online Fault Diagnosis
Wanga et al.	2006	Yes		Neural Network	Pattern Recognition (Protein Types Identification)
Jia and Culver	2006		Yes		Pattern Recognition (Synthetic Flow Generation)
Zhang	2007	Yes		Linear Combination	Batch Polymerization Reactor (Batch-to-batch control method)



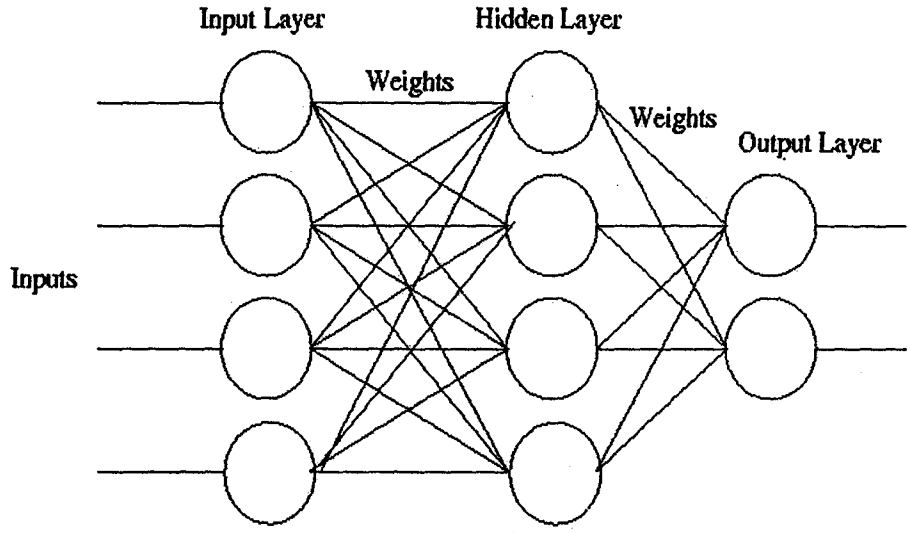


Figure 1. A Typical Single Neural Network

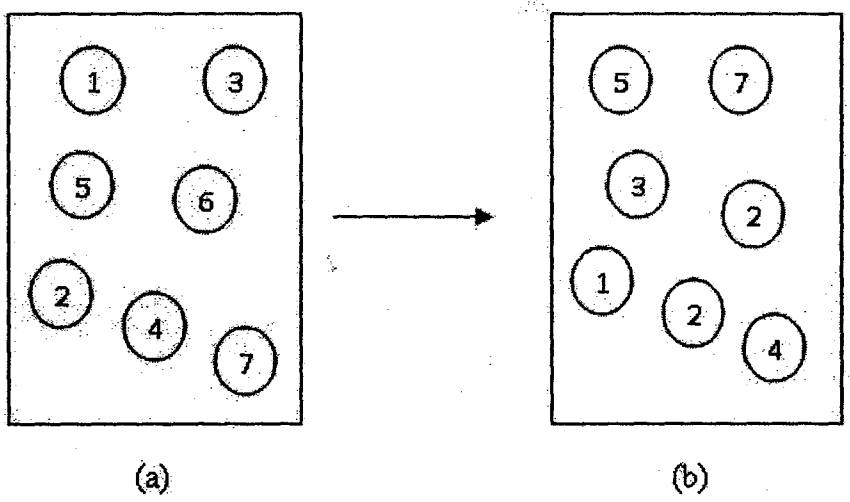


Figure 2. (a) Numbers represent data for sampling (b) data are randomly create using bootstrap method

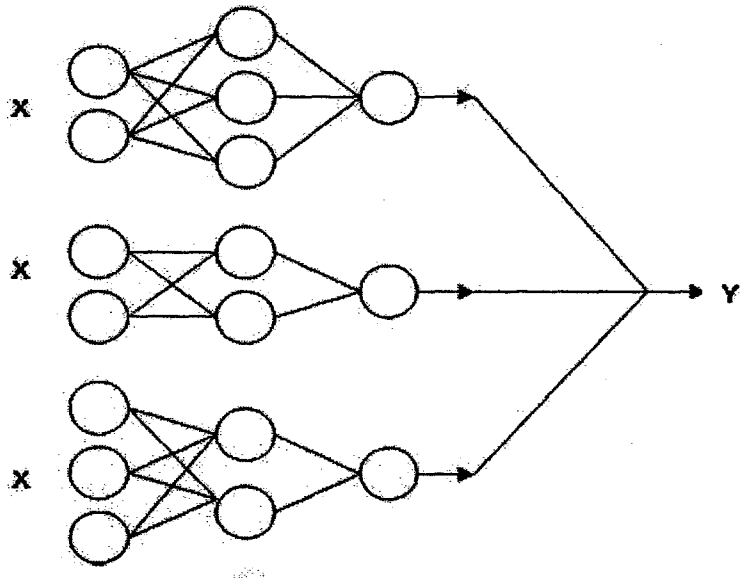


Figure 3. A Stacked Neural Networks

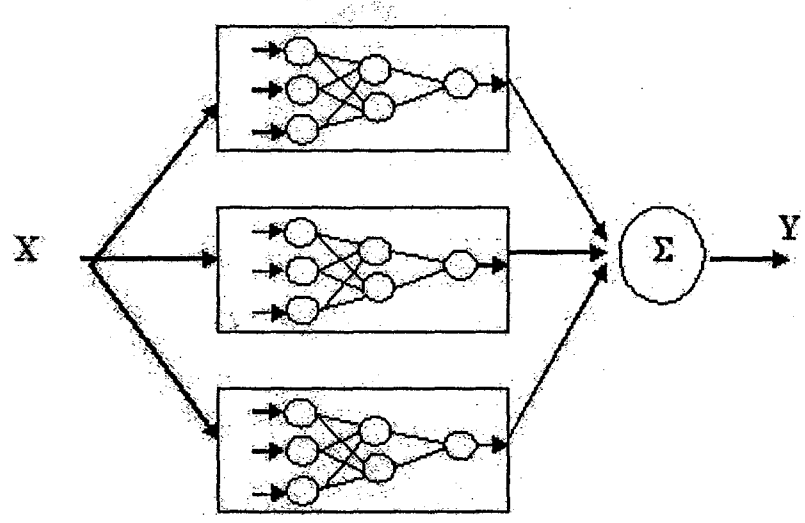


Figure 4. Combining Multiple Neural Networks

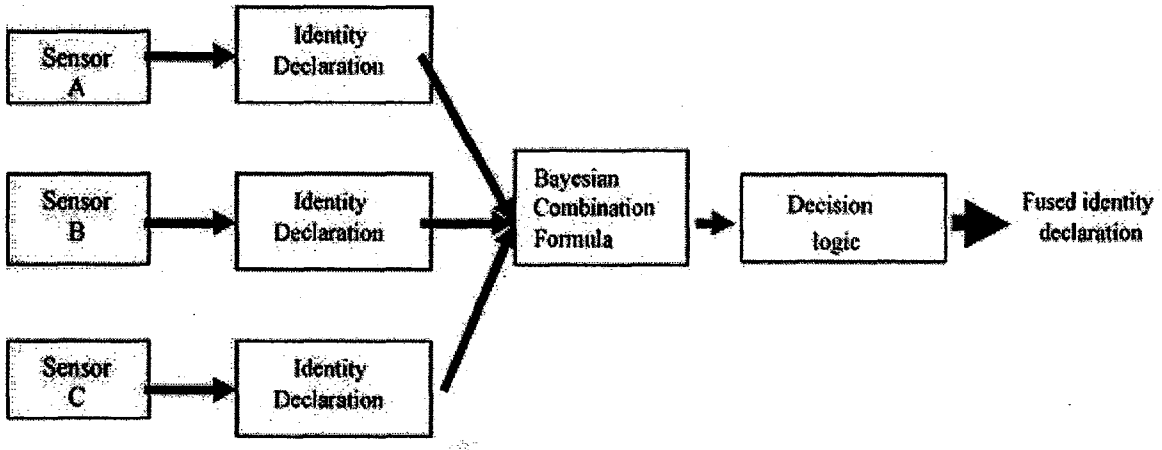


Figure 5. Architecture of Bayesian Inference in Data Fusion

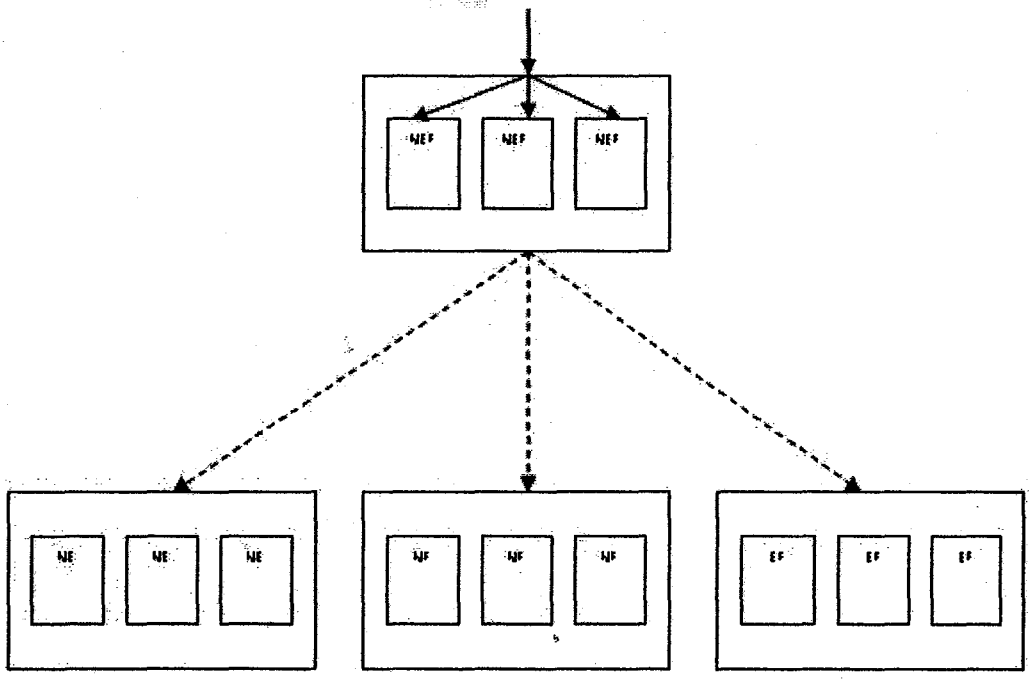


Figure 6. A Multi-Net System for Fault Diagnosis

FCh_07

IMPROVING SINGLE NEURAL NETWORK MODEL FOR REAL PH NEUTRALIZATION PROCESS USING BOOSTRAP RE-SAMPLING TECHNIQUE

Zainal Ahmad[1], Rabiatul 'Adawiah Mat Noor[2]
[1,2]Process Control Group,
School of Chemical Engineering, Engineering Campus, University Sains Malaysia (USM), Seri Ampangan, 14300, Nibong Tebal, Penang, Malaysia
Email: chzahmad@eng.usm.my

ABSTRACT

A single neural network model developed from a limited amount of data usually lacks robustness. Neural network model robustness can be enhanced by using re-sampling technique like boostrap during training of the networks. There are several approaches for re-sampling the data but in this paper boostrap re-sampling technique is employed. Comparisons of these methods on using original data without re-sampling are carried out in this paper and apply to pH neutralisation process which is a non-linear dynamic system. It is shown that training using re-sampling data generally improve model performance compare to using the original data.

Keywords: *boostrap re-sampling, neural networks, nonlinear modelling. pH neutralisation process*

INTRODUCTION

Artificial neural networks have been increasingly used in developing non-linear models in industry and model robustness is one of the main criteria that need to be considered when judging the performance of neural network models [1]. Model robustness is primarily related to the learning or training methods and the amount and representativeness of the training data [1; 2]. Even though neural networks have a significant capability in representing non-linear functions, inconsistency of accuracy still seems to be a problem where a neural network model cannot cope or perform well when it is applied to new unseen data. Furthermore, advanced process control and supervision of industrial processes require accurate process models promoting investigations in the robustness of neural networks models. Lack of robustness in neural network models is basically due to the over fitting and poor generalisation of the models (e.g. [3; 4]). Therefore, a lot of researchers have been interested and concentrated on how over fitting can be alleviated by improving the learning algorithms or by combining multiple neural networks (e.g.[3; 5; 6]). In view of improving the robustness of neural network models a lot of techniques have been developed like regularisation and the early stopping method (e.g.[3]). Reference [7] implemented the universal learning rule with second order derivatives to increase the robustness in neural network models. Among those approaches, re-sampling the data while training the network is quite promising enhancing the performance of single neural network. There are several methods in re-sampling the data like bagging but bootstrap re-samples technique is the most convincing in this case study where it actually the re-sampling of the original training data [5; 8; 9]. In this paper modelling of the real pH neutralisation process is implemented.

The pH control is very important in many processes. For examples, in wastewater treatment plant, the cell growth rate and the accurate stabilization of pH at an optimal level often determines the efficiency of the bioprocess. The regulation and control of a pH process is a typical problem found in a variety of industries including wastewater treatment, pharmaceuticals, biotechnology and chemical processing. It is a nontrivial task arising from the nonlinearity of the titration process. Therefore, controlling the pH at certain region or set point is very important. On the other hand, in chemical processes, pH neutralization is not easy to control due to the fast and quite complicated reaction [10; 11]. In terms of modelling, one of the disadvantages of pH neutralization is the difficulty of obtaining a rigorous mechanistic model of the process, which accounts for several important operating factors such as the flow rate of the influent stream, the flow rate of the titrating stream, the concentration of the influent stream, the concentration of the titrating stream, the concentration of the acid solution, and the volume of the mixture in the CSTR [12]. This is particularly true when knowledge about the process is initially vague or if the process is so complex that the resulting equations cannot be solved. Therefore modelling the pH is very challenging and a neural network is one of the options.

ISBN 978- 983-41705-8-5

Process modelling is an area where neural networks configurations and structures have been considered as alternative modelling techniques, particularly in cases where reliable mechanistic models cannot be obtained [13-15] where this is due to the complexity and difficulty in control, the model based control is come to the picture. As mention in [12], to be successful in implementing the control strategy for this system, the pH control system must contain two main features: (i) reliable estimation of the process nonlinearity and (ii) a nonlinear compensation and control. In this aspect the neural networks capabilities are utilized.

However, single neural networks sometimes lack robustness when the data is insufficient especially when dealing with real world data due to the fact that the robustness of the network is related to the representativeness of the training data [16]. Single neural networks sometimes suffer badly when applied to unseen data where some neural network might fail to deliver the correct result due to the network training converged to undesired local minima, over fitting or noise in the data (e.g. [4]). Therefore the re-sampling the original data in siingle neural networks using boostrap approach is implemented in this paper with the aim of enhancing the single neural network robustness.

BOOSTRAP RE-SAMPLING TECHNIQUE

Bootstrap application or bootstrap technique was first introduce in 1979 as a computer based method for estimating the standard error of empirical distribution [17]. In neural networks bootstrap basically relate or deals with the sampling to create random data sets for training and testing. By creating an equal number of bad and good data sampling, it actually improve the generalisation ability because it helps the identification of the characteristic of the scarce class [17]. Zhang [18] demonstrates that sampling by bootstrap does actually increase the robustness of the model and he came up with BAGNET or bootstrap aggregation neural networks [5; 18].

There are three main algorithms being used to re-sample or partition the training data which are bagging or bootstrap (e.g. [5; 1921], adaboost (e.g. [22; 23] and randomisation (e.g. [9]. The motivation of creating those different inputs or partitions is to create the effective network ensembles [24]. The bootstrap or bagging basically refers to replication of a training data set where the bootstrap algorithm re-samples the original training data set. Some of the data samples may occur several times, and other may not occur in the sample at all. The individual training sets are independent and the neural networks can be trained in parallel.

Adaboost or 'adaptive boosting' on the other hand constructs a composite classifier by sequentially training classifier while putting more and more emphasis on certain patterns [22]. The probability distribution over the original training data was maintained in this approach where the network is trained with respect to this distribution. In other words the networks are dependent to each other, while randomisation just randomly selects the original training data in each training data and each network can be trained parallel.

Each technique has it own capabilities or advantages in some application like bagging or bootstrap can generate diverse networks when the base learning algorithm is unstable in that small changes in the training data set cause large changes in the learned classifiers while boosting can result in less instability. Boosting or Adaboost can make larger changes in the training set like placing large weights on the training set. Based on the experiment conducted by Dietterich [9] randomisation method can give quite good performance when the noise level is low in the networks but bagging is still much better when high level of noise is introduced in the networks. But in this paper, the boostrap re-sampling technique is proposed.

As shown in Figure 1 and Figure 2, there is a different between the original data and the data after applying the boostrap re-sampling techniques. The data was re-sampled after arranging the data point into discrete time function for input and output which will be discussed in the next section.

ISBN 978- 983-41705-8-5 ©2007 FEIIC

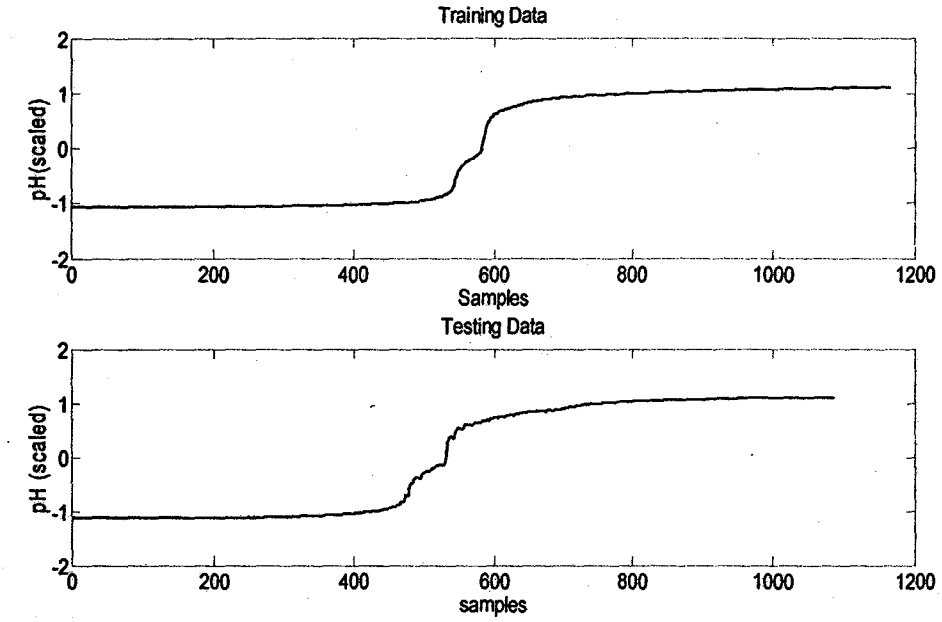


Figure 1. Original training and testing data for pH (scale)

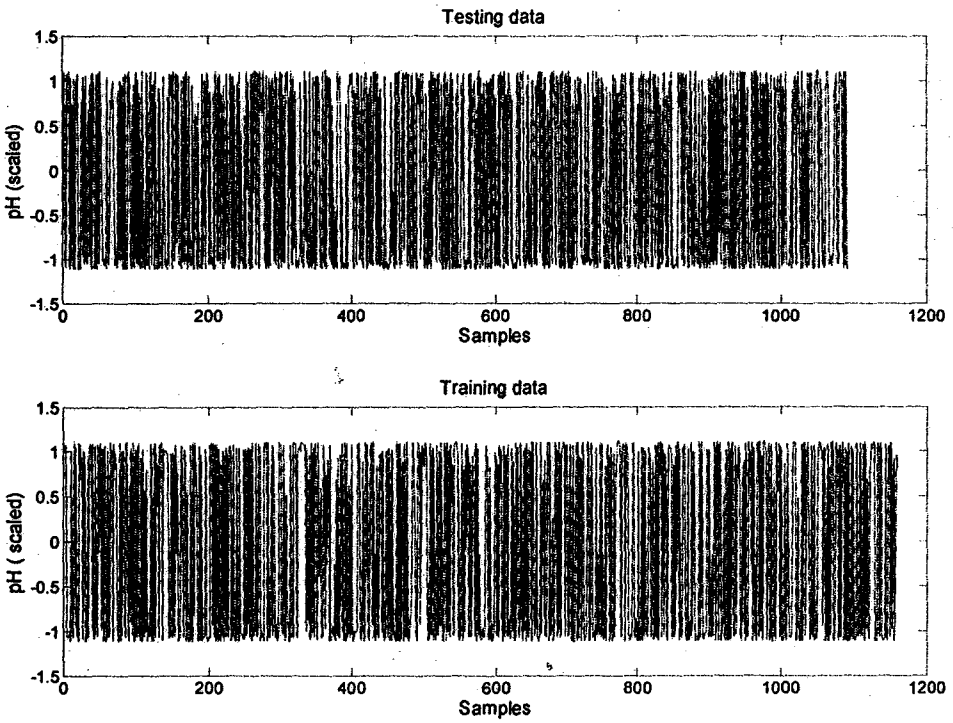


Figure 2. Training and testing data for pH (scale) after re-sampling using boostrap.

ISBN 978- 983-41705-8-5 ©2007 FEIIC

RESULTS AND DISCUSSIONS

The experimental data employed for modelling was obtained from a pH neutralization rig shown in Figure 3. A feed sodium hydroxide (NaOH) solution is fed to the CSTR by a diaphragm pump (metering pump). At the same time, a feed sulphuric acid (H_2SO_4) solution is fed to the CSTR by a diaphragm pump (masterflex pump). A stream leaves the CSTR is called neutralization effluent of the H_2SO_4 and NaOH solution. These NaOH stream and effluent stream pass through a pH sensor to measure its pH values. In this case study, 20 networks with fixed identical structure were developed from boostrap re-samples of the original training and testing data. In re-sampling the training and testing data using bootstrap re-sampling techniques, the training and testing was already in discrete time function, therefore by re-sampling discrete time function it's not effect the sequence of input-output mapping of the prediction.

Then the individual networks were trained by the Levenberg-Marquardt optimisation algorithm with regularisation and "early stopping". All weights and biases were randomly initialised in the range from -0.1 to 0.1. The individual networks are single hidden layer feed forward neural networks. Hidden neurons use the logarithmic sigmoid activation function whereas output layer neurons use the linear activation function. Instead of selecting a single neural network model, a combination of several neural network models is implemented to improve the accuracy and robustness of the prediction models.

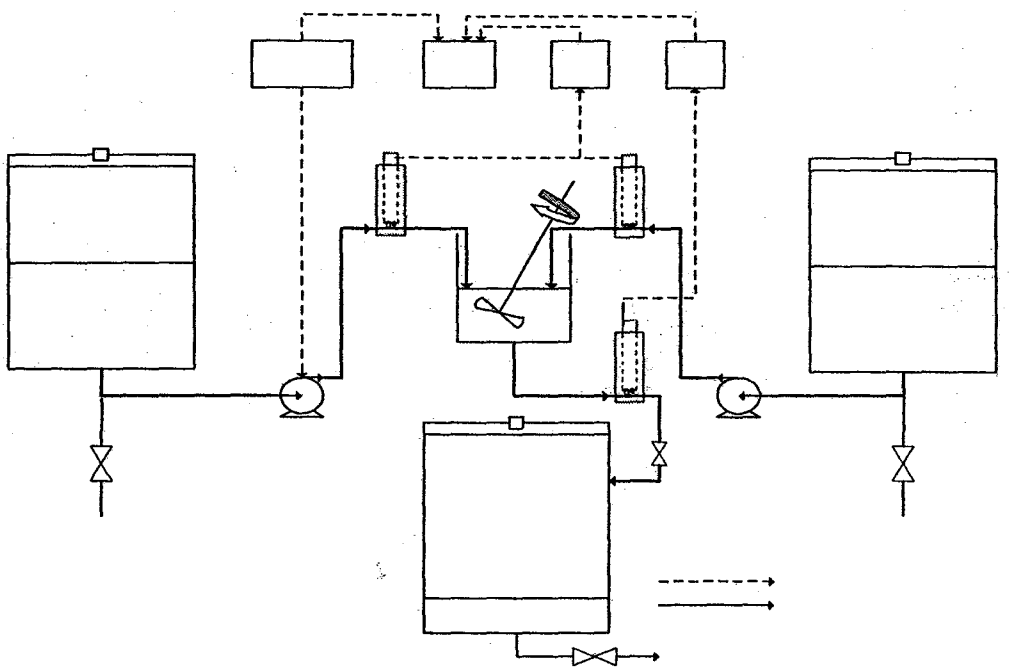


Figure 3. Advanced pH control schematic diagram

There were fours strokes percent for the metering pump during the data generation. The stroke percentages are 40, 60, 90 and 100 respectively. While the stroke length percent and stroke per minute for the masterflex pump are constant at 20 percent during the experiment. The other parameter such as NaOH concentration, H_2SO_4 concentration and H_2SO_4 stream stroke are also remain constant. The duration of each manipulated variables percent changes was 4 min. The process was allowed to reach steady state for perfect mixing during the first three minutes followed by pH evaluation of the effluent for the next one minutes. The pH value for the effluent was then obtained automatically through the pH sensor and the signals transmitted by the pH transmitter to the recorder and it have recorded in every two seconds. Then, the data generated from the experimental rig were divided in to training, testing and validation where in this case study the training data is based on the data taken from the strokes of 40 %, testing data from strokes 100 % and the remaining data is for validation.

pHIC pHR pHE 2

This case study apply a one-step-ahead predictions approach where , the process output at time (t-1), y(t-1), is used as a model input to predict the process output at time t, y(t), as follows:

ISBN 978- 983-41705-8-5 ©2007 FEIIC

$$\hat{y}(t) = f[y(t-1), u1(t-1), u2(t-1), \qquad (1)$$

where u1(t-1) and u2(t-1)is the process input at time (t-1) which is the acid flow and the pump strokes, $\hat{y}(t)$ is the predicted process output (pH) at time t , the lags for this model is 1 for both input and output.

Initially, the network was trained using all 1166 data points based on the 40% stroke of the masterflex pump for single and multiple neural networks. By using the LM optimization method, the training stopped after 100 iterations with the sum square error SSE value of 0.0392 and the correlation coefficient R-square equal to 1.00. The trained network was simulated by feeding it with all of the 40 percent stroke data. Then, the model was tested using 100 percent stroke data which contains 1086 data points. The testing also stopped after 100 iterations with the sum square error SSE value of 0.6935 and the correlation coefficient R-square equal to 0.9994. Figure 4 presents a plot of the pH value for both network outputs (predicted pH value) and the targets (actual pH value) versus the data points for single neural networks for training and testing data.

The model has been validated using 60% and 90% stroke data which contain around 1000 data points in each set. The validation data will determine whether the generalization capability of the model developed using 40% and 100% data for training and testing is acceptable.

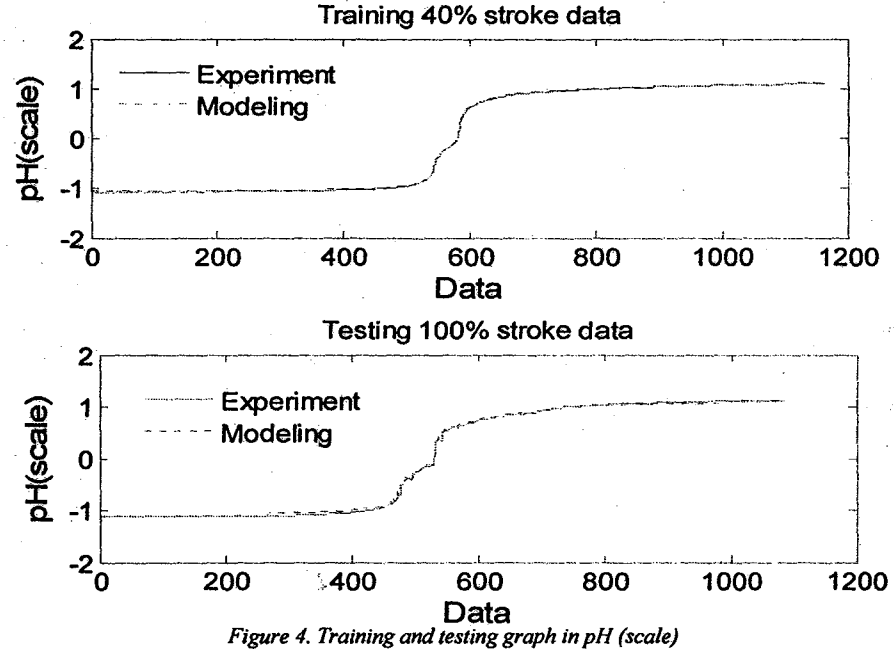


Figure 4. Training and testing graph in pH (scale)

Figure 5 and Figure 6 shows the model and actual output in the validation data for single neural networks (SNN) using original training and testing data. It clearly seen that the single neural networks was performed quite well. The predicted model output showed quite the same as the experiment data, but there is some errors occurred at the low pH region as well as at the end of the high region and also at the transition between the low region and middle region. This might be due to the transition of the pH especially from low region to higher region where the neutralization process was very fast, small changes in the input (acid flow) give a lot of affect to the process.

Then re-sampling technique using boostrap approach is applied and the result was shown in Figure 7 and Figure 8 for 60 % and 90 % stroke data respectively. It clearly seen that from Figures 6 and 7, single neural network prediction is significantly better than single neural networked using original data generated previously. The predicted and the experiment value can be seen exactly matching for both data.

ISBN 978- 983-41705-8-5 ©2007 FEIIC

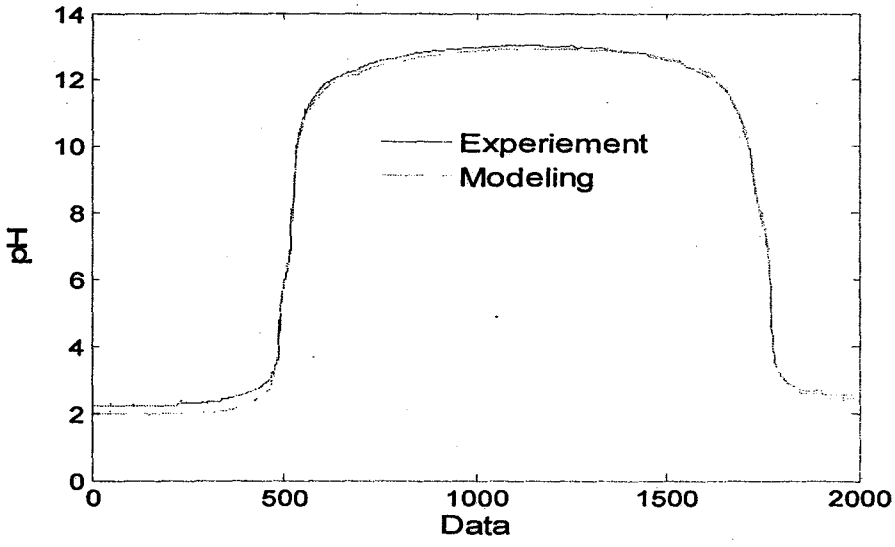


Figure 5. Validation output for 60 % stroke data

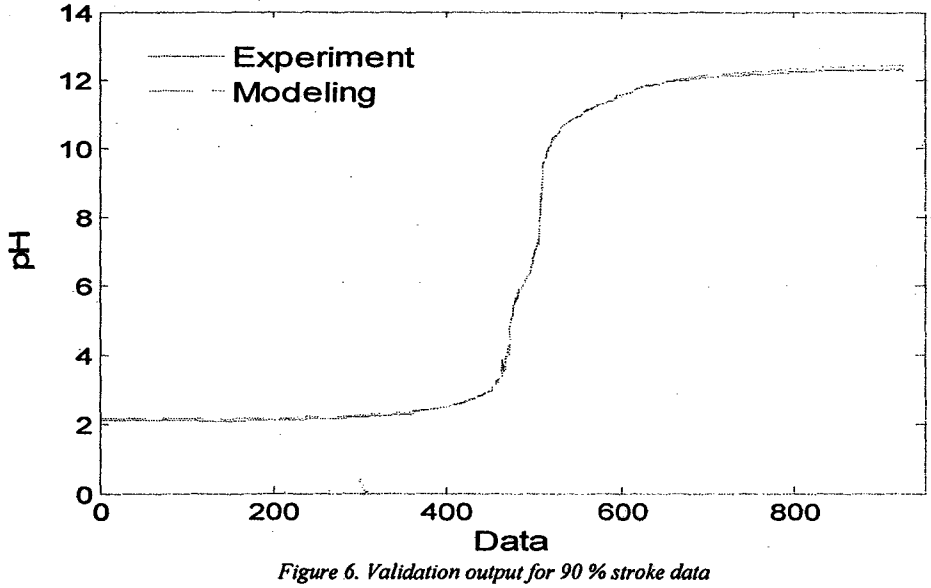


Figure 6. Validation output for 90 % stroke data

In order to test further the performance of the model, statistical analysis was carried out which is sum square error (SSE), mean square error (MSSE) and relative correlation R-square analysis. The overall statistical analysis result of SSE, MSSE and relative correlation R-squared shown in the Table 1. It is clearly shown in Table 1 that the SSE and the MSSE is quite small, the relative correlation (R-square) is 1 for re-sampling SNN while in original SNN prediction is slightly lower than 1. Meanwhile for SSE and MSSE, the original SNN produced higher value compare to re-sampling SNN. It is shown that the re-sampling SNN model can predict significantly well even though using real process data compare to original SNN.

ISBN 978- 983-41705-8-5 ©2007 FEIIC

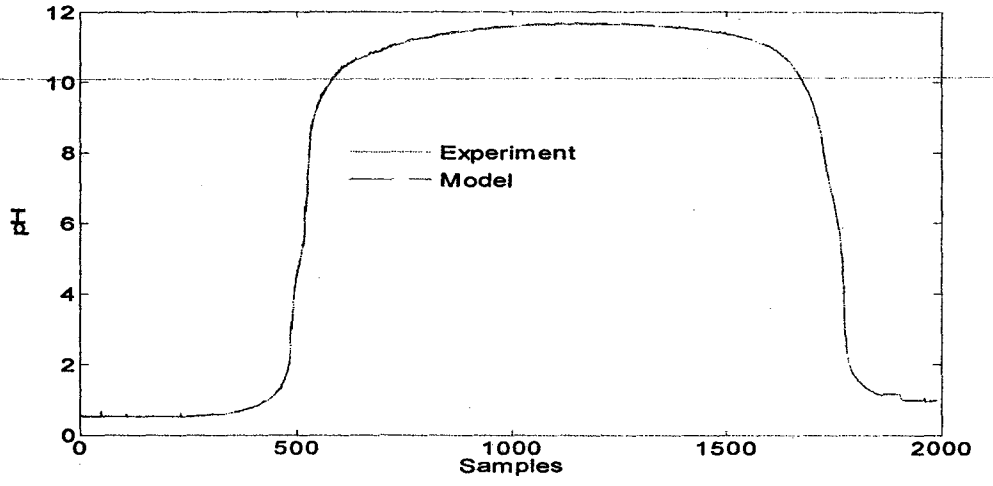


Figure 7. Validation output for re-sampling SNN for 60 % stroke data

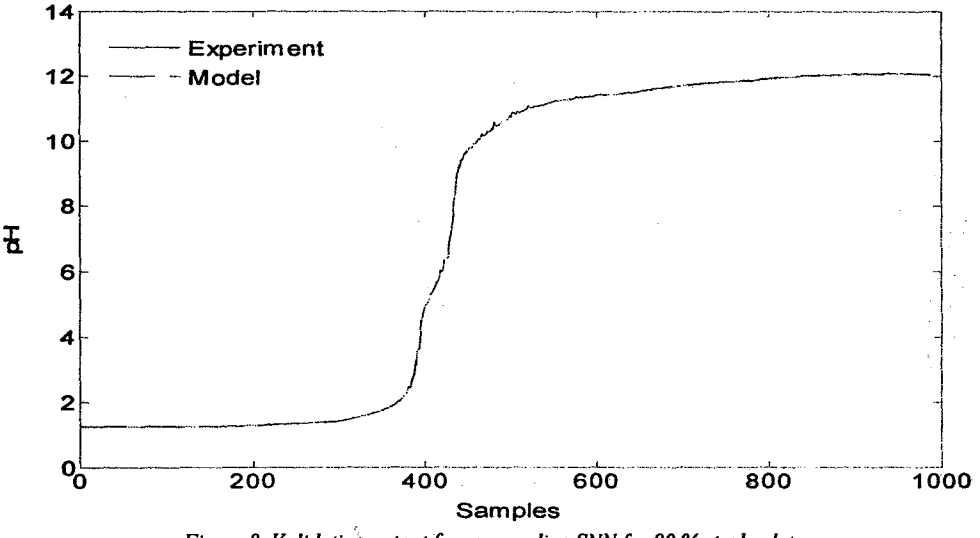


Figure 8. Validation output for re-sampling SNN for 90 % stroke data

Table 1: Result of the output based on the single neural networks application on the validation data.

Data	SSETv		MSSETv		RsquareTv	
	Original SNN	Re-sampling SNN	Original SNN	Re-sampling SNN	Original SNN	Re-sampling SNN
60	2.6757	0.0988	1.3e-3	4.9652e-005	0.9977	1.0000
90	0.4584	0.0442	4.64E-04	4.4782e-005	0.9996	1.0000

ISBN 978- 983-41705-8-5 ©2007 FEIIC

CONCLUSIONS

A single neural network (SNN) was developed to model the performance of a pH neutralization process using experimental data, which was subjected to a series of different stroke percent for sodium hydroxide stream. The inputs to the network were the sodium hydroxide stream flow rate and metering pump percent stroke, and the output was the pH values of the effluent. The Levenberg–Marquardt optimization technique was used together with the 'early stopping' and regularisation methods to improve the robustness of the network.

Application to the real pH neutralization process shows re-sampling neural networks (SNN) increased the robustness of the models compared to original single neural network (SNN). The SSE is decreased as well as the increment of R-square analysis compare to original single neural networks in all validation data. The result for re-sampling neural networks was consistent especially in R-square and it's concluded that re-sampling neural networks can significantly produced a better models.

ACKNOWLEDGEMENT

The authors would like to thank University Sains Malaysia (USM) for financially support this research project through short term grant 6035182.

REFERENCES

[1] Willis M J, C Di Massimo, G A Montaque, M T Tham and A J Morris.(1992). Artificial Neural networks in Process Estimation and Control,Automatica 28 (6) (pp: 1181-1187).
[2] Zhang J.(2001). Developing Robust Neural Network Models by Using Both Dynamic and Static Process Operating Data,Ind.Eng.Chem.Res 40 (pp: 234-241).
[3] Mc Loone S and G Irwin.(2001). Improving Neural Networks Training Solution Using Regularisation,Neurocomnputing 37 (pp: 71-90).
[4] Caruana R, S Lawrence and C Lee Giles.(2000). Overfitting in Neural Networks: Backpropagation, Conjugate Gradient and Early Stopping,Neural Information Processing System 13 (pp: 402-408).
[5] Zhang J.(1999). Developing Robust Non-linear Models Through Bootstrap Aggregated Neural Networks,Neurocomputing 25 (pp: 93-113.).
[6] Guyon X and J Yao.(1999). On the Underfitting and Overfitting Sets of Model Chosen by Order Selection Criteria,Journal of Multivariate Analysis 70 (pp: 221-249).
[7] Hagiwara K and K Kuno.(2000). Regularisation Learning and Early Stopping in Linear Networks,International Joint Conference on Neural Networks (IJCNN 2000) (pp: 511-516).
[8] Franke J and M H Neumann.(2000). Bootstrapping Neural Networks,Neural Computation 12 (pp: 1929–1949).
[9] Dietterich T G.(2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization,Machine Learning 40 (2) (pp: 139 - 157).
[10] Chan H C and C C Yu.(1995). Autotuning of gain scheduled pH control: an experimental study,Ind. Eng. Chem. Res. 34
[11] Costello D J.(1994). Evaluation of model based control techniques for a buffered acid-base reaction system,Trans. IChemI, 72 (Part A)
[12] Loh A R, K O Looi and K F Fong.(1995). Neural network modelling and control strategies for a pH process,Journal of Process Control 5 (6) (pp: 355-362).
[13] Chen L and K S Narendra.(2003). Inttelligent Control using Multiple Neural Networks and Multiple Models,Automatica 37 (pp: 1245-1255).
[14] Nguyen H H and C W Chan.(2004). Multiple Neural Networks for a Long Term Time Series Forecast,Neural Computation and Application 13 (pp: 90-98).
[15] Chen L, S K Nguanga, C X Dong and L X Mei.(2004). Modelling and Optimization of Fed-batch Fermentation Processes using Dynamic Neural Networks and Genetic Algorithms,Biochemical Engineering Journal 22 (pp: 51-61).
[16] Zhang J.(2002) Sequential Training of Bootstrap Aggregated Neural Networks for Nonlinear Systems Modelling. In the Proceeding American Control Conference.
[17] Dupret G and M Koda.(2001). Theory and Methodology Bootstrap re-sampling for Unbalanced Data in Supervised Learning,European Journal of Research 134 (pp: 141-156).

[18] Zhang J, A J Morris, E B Martin and C Kiparissides.(1998). Prediction of Polymer Quality in Batch Polymerisation Reactors Using Robust Neural Networks,Chemical Engineering Journal 69 (pp: 135-143).

[19] Brieman L.(1994). Bagging Predictors (Tech.Rep.No.421),Department of Statistic University of California at Berkeley, USA.

[20] Cunningham P, J Carney and S Jacob.(2000). Stability Problem with Artificial Neural Networks and The Ensembles Solutions,Artificial Intelligence in Medicine 20 (pp: 217-225).

[21] Wehrens R, H Putter and L M C Buyden.(2000). The bootstrap: a tutorial,Chemometrics and Intelligent Laboratory System 54 (pp: 35-52).

[22] Schwenk H and Y Bengio.(2000). Boosting Neural Networks,Neural Computation 12 (pp: 1869-1887).

[23] Freund Y and R E Schapire.(1996) Experiment with a new Boosting Algorithm. In the Proceeding 13th International Conference on Machine Learning.

[24] Sharkey A J C. (1999).Multi Nets System. A. J. C. Sharkey (Ed).In Combining Artificial Neural Nets Ensemble and Modular. London, Springer Publication.

ISBN 978- 983-41705-8-5

Modeling of real pH Neutralization Process using Multiple Neural Networks (MNN) Combination Technique

Zainal Ahmad* and Fairuoze Roslin
Process Control Group,
School of Chemical Engineering, Engineering Campus, University Sains Malaysia (USM), Seri Ampangan, 14300, Nibong Tebal, Penang, Malaysia
Tel : +604-593 7788 ext 6462, Fax : +604 594 1013
Email: chzahmad@eng.usm.my, fairuo_ze@yahoo.com

Abstract

Combining multiple neural networks appears to be a very promising approach in improving neural network generalisation since it is very difficult, if not impossible, to develop a perfect single neural network (SNN) especially when dealing with a real time data. Therefore, in this paper, two feedforward neural networks model technique are developed to predict the performance of a pH neutralization process, which uses a sulphuric acid as the acidic stream and sodium hydroxide aques as the bes stream. The technique involves combining multiple neural networks (MNN) and single neural network (SNN). The Levenberg–Marquardt (LM) optimization technique was employed for training the NN for both techniques. Application results demonstrate that the proposed multiple neural networks (MNN) combination techniques significantly improve model generalisation compared to single neural network (SNN) models.

Keywords: Neural networks, Multiple Neural Networks, simple averaging, nonlinear process modeling.

1. Introduction

The pH control is very important in many processes. For examples, in wastewater treatment plant, the cell growth rate and the accurate stabilization of pH at an optimal level often determines the efficiency of the bioprocess. The regulation and control of a pH process is a typical problem found in a variety of industries including wastewater treatment, pharmaceuticals, biotechnology and chemical processing. It is a nontrivial task arising from the nonlinearity of the titration process. Therefore, controlling the pH at certain region or set point is very important. On the other hand, in chemical processes, pH neutralization is not easy to control due to the fast and quite complicated reaction [1,2]. In terms of modeling, one of the disadvantages of pH neutralization is the difficulty of obtaining a rigorous mechanistic model of the process, which accounts for several important operating factors such as the flow rate of the influent stream, the flow rate of the titrating stream, the concentration of the influent stream, the concentration of the titrating stream, the concentration of the acid solution, and the volume of the mixture in the CSTR [3]. This is particularly true when knowledge about the process is initially vague or if the process is so complex that the resulting equations cannot be solved. Therefore modeling the pH is very challenging and a neural network is one of the options.

Process modeling is an area where neural networks configurations and structures have been considered as alternative modeling techniques, particularly in cases where reliable mechanistic models cannot be obtained [4–9] where this is due to the complexity and difficulty in control, the model based control is come to the picture. As mention in [1], to be successful in implementing the control strategy for this system, the pH control system must contain two main features: (i) reliable estimation of the process nonlinearity and (ii) a nonlinear compensation and control. In this aspect the neural networks capabilities are utilized.

Why neural network? Artificial neural networks have been shown to be able to approximate any continuous non-linear functions and have been used to build data base empirical models for non-linear processes [10]. Hence what is a neural network? According to [11].

* corresponding authors

'A neural network is a massive parallel-distributed processor that has a natural capability for storing experiential knowledge and making it available for use. It resembles the brain in two respects knowledge is acquired by the networks through a learning process. Interneuron connection strengths known as synaptic weights are used to store the knowledge'

Furthermore, the main advantage of neural network based process models is that they are easy to build. This feature is particularly useful when modelling complicated processes where detailed mechanistic models are difficult to develop. However a critical shortcoming of neural networks is that they often lack robustness unless a proper network training and validation procedure is used. Robustness of the model can be defined as one of the baseline to judge the performance of the neural network models and it is really related to the learning or training classes as what Bishop [12] described:

'The importance of neural networks in this context is that they offer very special powerful and very general framework for representing non-linear mappings from several input variables to several output variables, where the form of the mapping is governed by a number of adjustable parameters.'

Therefore a lot of techniques have been introduced to improve the generalisation capability of neural network models like regularisation techniques [e.g.13,14,15] Bayesian Learning [e.g. 16,17] and also by using the parsimonious networks structure [18]. The most exceptional model for this approach is network pruning techniques and sequential orthogonal training techniques. A sequential orthogonal training techniques gradually builds up a neural network model and avoids unnecessarily large networks structure [19,20].

However, single neural networks sometimes lack robustness when the data is insufficient especially when dealing with real world data due to the fact that the robustness of the network is related to the representativeness of the training data [12]. Single neural networks sometimes suffer badly when applied to unseen data where some neural network might fail to deliver the correct result due to the network training converged to undesired local minima, overfitting or noise in the data [e.g. 21,22]. Therefore the combination of multiple neural networks using simple averaging approach is implemented in this paper with the aim of enhancing the single neural network robustness.

2. Multiple Neural Networks

The idea of multiple neural networks came up from Wolpert [23] where he described about stacked generalisation which is a technique for combining different representations to improve the overall prediction performance. It can also be described as architecture of network consisting of several sub-models and a mechanism which combines the outputs of these sub-models [24]. There are several types of multiple neural networks but the underlying ideas are basically similar and the main difference is on how to create the sub-models as shown in Figure 1 and combined those output to get a single output.

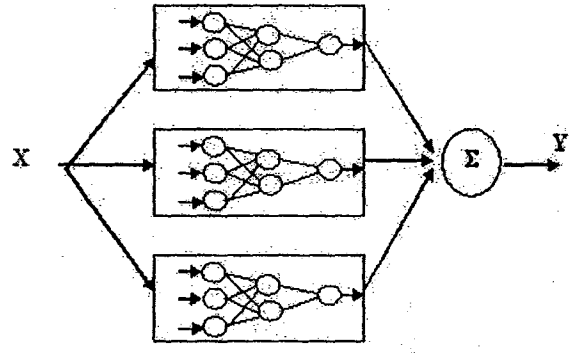


Figure 1. Combining multiple neural networks

Methods of combining multiple networks in current literature can be divided into linear and nonlinear combinations. The common linear combination is averaging and weighted averaging. The linear combination of multiple outputs is to create a single output as a final prediction. In weighted averaging, individual network outputs are multiplied by appropriate weights and then combined to give the final model prediction. Weighted averaging includes PCR and MLR approaches. Zhang [18] used PCR approach to select the combination weights. Another combination scheme is by Wolpert [23] which combines the networks with weights that vary over the feature space. The output from a set of level 0 generaliser are used as the input to level 1 generaliser, which is trained to produce the appropriate output.

Nonlinear combination techniques, include Demspter-Shafers belief based method [25], majority voting [e.g. 26], and also Bayesian model averaging. The Demspter-Shafers belief based method is quite complex and it have to deal with the uncertainty and ignorance of the classifiers. This approach is usually used in model classification or pattern recognition

when each network or model represents a character of the image, same as the majority voting combination for example in handwritten recognition [27]. For this paper, simple averaging combination technique is employ to get a final single output. This method is the most common method in combining several model outputs with the weights fixed as shown below:

$$\hat{Y} = w_1\hat{y}_1 + w_2\hat{y}_2 + \ldots + w_n\hat{y}_n \quad (1)$$

where $\hat{y}_i$ is the network prediction from the *i*th network, n is the number of networks to be combined, $\hat{Y}$ is the final prediction output, and $w_i = 1/n$ is the weight for combining the *i*th network. In this paper the number of network to be combined is 20. In this approach all the networks have the same contribution to the final prediction output even though some of the networks might have better predictions then others.

3. Case study: pH Neutralization Process

The experimental data employed for modeling was obtained from a pH neutralization rig shown in Figure 2. A feed sodium hydroxide (NaOH) solution is fed to the CSTR by a diaphragm pump (metering pump). At the same time, a feed sulphuric acid (H_2SO_4) solution is fed to the CSTR by a diaphragm pump (masterflex pump). A stream leaves the CSTR is called neutralization effluent of the H_2SO_4 and NaOH solution. These NaOH stream and effluent stream passes through a pH sensor to measure its pH values. In this case study, 20 networks with fixed identical structure were developed from boostrap re-samples of the original training and testing data. In re-sampling the training and testing data using bootstrap re-sampling techniques, the training and testing was already in discrete time function, therefore by re-sampling discrete time function it's not effect the sequence of input-output mapping of the prediction.

Then the individual networks were trained by the Levenberg-Marquardt optimisation algorithm with regularisation and "early stopping". All weights and biases were randomly initialised in the range from –0.1 to 0.1. The individual networks are single hidden layer feed forward neural networks. Hidden neurons use the logarithmic sigmoid activation function whereas output layer neurons use the linear activation function. Instead of selecting a single neural network model, a combination of several neural network models is implemented to improve the accuracy and robustness of the prediction models.

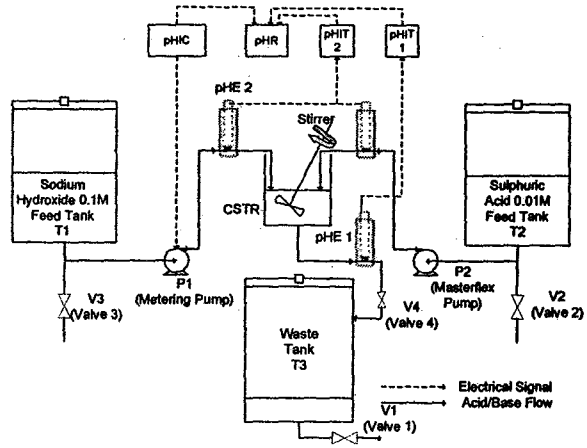


Figure 2. Advanced pH control schematic diagram

There were fours strokes percent for the metering pump during the data generation. The stroke percentages are 40, 60, 90 and 100 respectively. While the stroke length percent and stroke per minute for the masterflex pump are constant at 20 percent during the experiment. The other parameter such as NaOH concentration, H_2SO_4 concentration and H_2SO_4 stream stroke are also remain constant. The duration of each manipulated variables percent changes was 4 min. The process was allowed to reach steady state for perfect mixing during the first three minutes followed by pH evaluation of the effluent for the next one minutes. The pH value for the effluent was then obtained automatically through the pH sensor and the signals transmitted by the pH transmitter to the recorder and it have recorded in every two seconds. Then, the data generated from the experimental rig were divided in to training, testing and validation where in this case study the training data is based on the data taken from the strokes of 40 %, testing data from strokes 100 % and the remaining data is for validation.

This case study apply a one-step-ahead predictions approach where , the process output at time (t-1), y(t-1), is used as a model input to predict the process output at time t, y(t), as follows:

$$\hat{y}(t) = f[y(t-1), u1(t-1), u2(t-1), \quad (2)$$

where u1(t-1) and u2(t-1)is the process input at time (t-1) which is the acid flow and the pump strokes, $\hat{y}(t)$ is the predicted process output (pH) at time t , the lags for this model is 1 for both input and output.

4.0 Results and discussion

Initially, the network was trained using all 1166 data points based on the 40 % stroke of the masterflex pump for single and multiple neural networks. By using the LM optimization method, the training stopped after 100 iterations with the sum square error SSE value of 0.0392 and the correlation coefficient R-square equal to 1.00. The trained network was simulated by feeding it with all of the 40 percent stroke data. Then, the model was tested using 100 percent stroke data which contains 1086 data points.

The testing also stopped after 100 iterations with the sum square error SSE value of 0.6935 and the correlation coefficient R-square equal to 0.9994. Figure 3 presents a plot of the pH value for both network outputs (predicted pH value) and the targets (actual pH value) versus the data points for single neural networks and assumption has been made that by duplicating this individual network using bootstrap re-sampling method, the multiple neural networks model will perform as closed as possible to this model or better after combination. In this case, all predicted points are close to the actual, which means that the network has learned the input–output mappings with a good degree of accuracy.

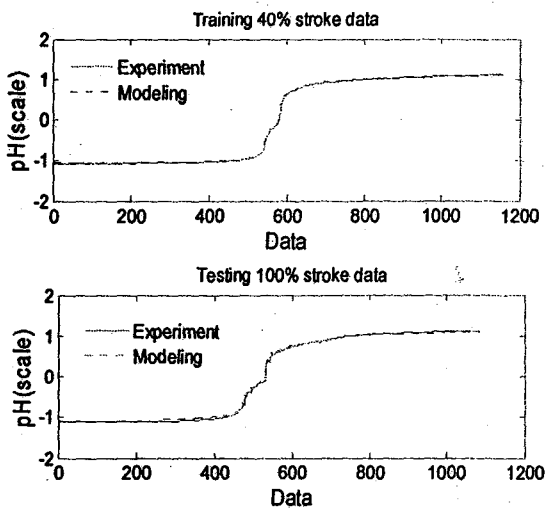


Figure 3. Training and testing graph in pH (scale)

The model has been validated using 60 % and 90 % stroke data which contain around 1000 data points in each set. The validation data will determined whether the generalization capability of the model developed using 40 % and 100 % data for training and testing is acceptable.

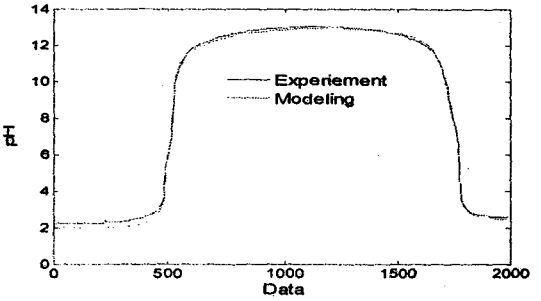


Figure 4. Validation output for 60 % stroke data

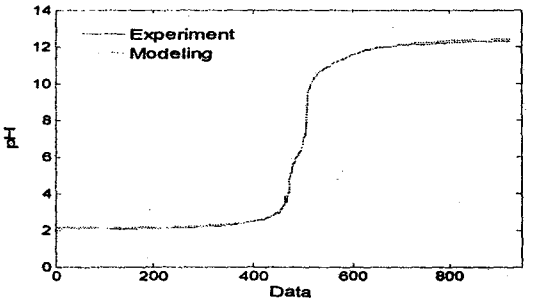


Figure 5. Validation output for 90 % stroke data

Figure 4 and Figure 5 shows the model and actual output in the validation data for single neural networks (SNN). It clearly seen that the single neural networks was performed quite well. The predicted model output showed quite the same as the experiment data, but there is some errors occurred at the low pH region as well as at the end of the high region and also at the transition between the low region and middle region. This might be due to the transition of the pH especially from low region to higher region where the neutralization process was very fast, small changes in the input (acid flow) give a lot of affect to the process.

Then multiple neural networks (MNN) combination approach is applied and the result was shown in Figure 6 and Figure 7 for 60 % and 90 % stroke data respectively. It clearly seen that from Figure 6 and Figure 7, multiple neural networks prediction is significantly better than single neural networked. The predicted and the experiment value can be seen exactly matching for both data.

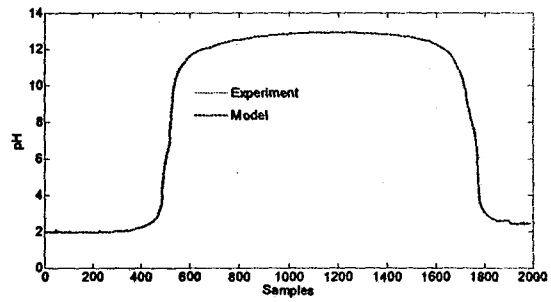


Figure 7. Multiple Neural Networks validation output for 60 % stroke data

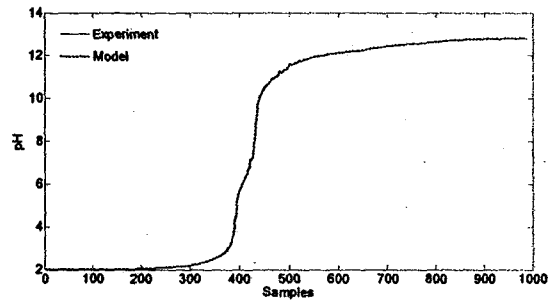


Figure 8. Multiple Neural Networks validation output for 90 % stroke data

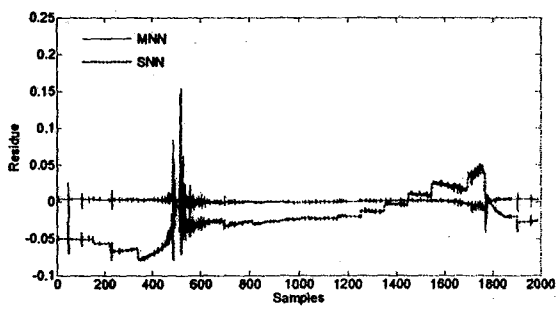


Figure 9. Residue for multiple neural networks (MNN) and single neural network (SNN) prediction for 60 % stroke data

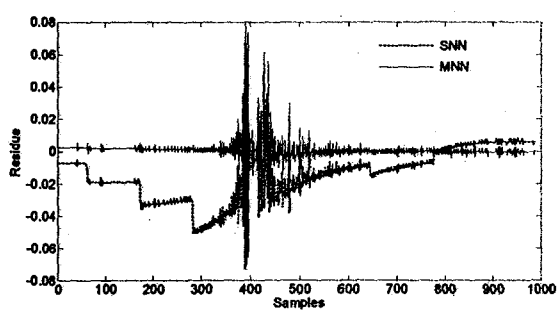


Figure 10. Residue for multiple neural networks (MNN) and single neural network (SNN) prediction for 90 % stroke data

The performance of MNN combination is encouraging especially based on the residue analysis which is shown in Figure 9 and Figure 10. The residue is constant for MNN but for SNN is quite inconsistent especially in the transition of low and upper region. This contributed to the large number of SSE for SNN prediction.

In order to test further the performance of the model, statistical analysis was carried out which is sum square error (SSE), mean square error (MSSE) and relative correlation R-square analysis as well as residue analysis.

The overall statistical analysis result of SSE, MSSE and relative correlation R-square shown in the Table 1 and Table 2. It is clearly shown in Table 1 that the SSE and the MSSE is quite small and in Table 2, the relative correlation (R-square) is nearly to 1 for MNN while in SNN prediction, it's slightly larger for SSE and MSSE. It is shown that the MNN combination model can predict significantly well even though using real process data.

Table 1. Result of the output based on the single and multiple neural networks application on the validation data.

Data	SSETv		MSSETv	
	SNN	MNN	SNN	MNN
60	2.6757	0.0880	0.0013	4.4234e-005
90	0.4584	0.0458	4.64E-04	4.6383e-005

Table 2. Result of the output based on the single and multiple neural networks application on the validation data for R-square.

Data	RsquareTv	
	SNN	MNN
60	0.9977	0.9999
90	0.9996	1.0000

5. Conclusion

A multiple neural network (MNN) was developed to model the performance of a pH neutralization process using experimental data, which was subjected to a series of different stroke percent for sodium hydroxide stream. The inputs to the network were the sodium hydroxide stream flow rate and metering pump percent stroke, and the output was the pH values of the effluent. The Levenberg–Marquardt optimization technique was used together with the 'early stopping' and regularisation methods to improve the robustness of the network.

Application to the real pH neutralization process shows that combining multiple neural networks (MNN) increased the robustness of neural network models compared to single neural network (SNN). The SSE is decreased as well as the increment of R-square analysis compare to single neural networks in all validation data. The result for multiple neural networks combination was consistent especially in residue analysis as well as in R-square and it's concluded that combining multiple neural networks can significantly produced a better models.

Acknowledgement

The authors would like to thank University Sains Malaysia (USM) for financially support this research project using short term grant 6035182.

References

[1]. Chan, H C and C. C. Yu., "Autotuning of gain scheduled pH control: an experimental study", *Ind. Eng. Chem. Res.*, vol. 34, 1995.

[2]. Costello, D J., "Evaluation of model based control techniques for a buffered acid-base reaction system", *Trans. IChemI*, vol. 72. Part. A, January 1994.

[3]. Loh, A R, K. O. Looi and K. F. Fong, "Neural network modelling and control strategies for a pH process", *J. Proc. Control*. Vol. 5, 1995, No. 6, pp. 355-362.

[4]. Nascimento, C A O, R. Giudici, R. Guardani, "Neural network based approach for optimization of industrial chemical processes", *Comp. Chem. Eng.* 24 (9–10) (2000) 2303–2314.

[5]. Delegrange-Vincent,N, C. Cabassud, M. Cabassud, L. Durand-Bourlier, J.M. Laine, "Neural networks for long term prediction of fouling and backwash efficiency in ultrafiltration for drinking water prediction", Desalination 131 (1–3) (2000) 353–362.

[6]. Willis, M J, G.A. Montague, C. Di Massimo, M.T. Tham, A.J. Morris, *Artificial neural networks in process estimation and control*, Automatica 28 (6) (1992) 1181–1187.

[7]. Pigram, G.M, T.R. Macdonald, *Use of neural network models to predict industrial bioreactor effluent quality*, Environ. Sci. Technol. 35 (1) (2001) 157–162.

[8]. Mirzai, A.R, J.R. Leigh, "An overview of the applications of neural networks in process engineering", *Comput. Control Eng. J.* 3 (5) (1992) 105–112.

[9]. Karjala, T.W, D.M. Himmelblau, "Dynamic data rectification by recurrent neural networks vs. traditional methods", *AIChE J.* 40 (11) (1994) 1865–1875.

[10]. Hertz J, Krogh A and Palmer R G., *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA, 1991.

[11]. Haykin S. *Neural Networks*, Macmillan College Publicating Company, 1994

[12]. Bishop C., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995

[13]. Caruana R, Lawrence S and Lee Giles C, *Overfitting in Neural Networks: Backpropagation, Conjugate Gradient and Early Stopping.* Neural Information Processing System , 2000, 13: 402-408.

[14]. Hagiwara K and Kuno K., "Regularisation Learning and Early Stopping in Linear Networks". *International Joint Conference on Neural Networks* (IJCNN 2000): 511-516.

[15]. McLoone S and Irwin G., "Improving Neural Networks Training Solution Using Regularisation. Neurocomnputing", 2001, 37: 71-90.

[16]. Crucianu M, Bone R and Asselin de Beauville J P., *Bayesian Learning for Re-current Networks. Neurocomputing* , 2001, 36: 235-242.

[17]. Radford M N., *Bayesian Learning for Neural Networks. Statistics*, 1997, 118-183.

[18]. Zhang J, Morris A J, Martin E B and Kiparissides C., "Prediction of Polymer Quality in Batch Polymerisation Reactors Using Robust Neural Networks". *Chemical Engineering Journal*, 1998, 69:135-143.

[19]. Zhang J and Morris A J, *A Sequential Learning Approach for Single Hidden Layer Neural Networks.* Neural Networks, 1998, 11(1): 65-80.

[20]. Zhang J. "Sequential Training of Bootstrap Aggregated Neural Networks for Nonlinear Systems Modelling". In the Proceedings of *American Control Conference* (ACC 2002), 1:531-536.

[21]. Hashem S., *Optimal Linear Combination.* Neural Networks, 1997, 10(4):599-614.

[22]. McLoone S and Irwin G., *Improving Neural Networks Training Solution Using Regularisation. Neurocomnputing* , 2001,37: 71-90

[23]. Wolpert D H., *Stacked Generalisation.* Neural networks, 1992,5:241-259.

[24]. Eikens B and Karim M N., *Process* "Identification with Multiple Neural Networks Models". *International Journal of Control* 72(7/8), 1999, 72(7/8):576-590.

[25]. Rogova G., *Combining the Result of Several Neural Networks Classifiers.* Neural Network, 1994, 7(5):777-781.

[26]. Battiti R and Colla M., *Democracy in Neural Net: Voting Schemes for Classification*. Neural Networks, 1997, 7(4):691-707.
[27]. Xu L, Krzyzak A and Suen C Y. "Methods for Combining Multiple Classifiers and their Application to Handwriting Recognition". *IEEE* Transaction on System, Man, and Cybernetics, 1992, 22(3):418-435.

A REVIEW OF NEURAL NETWORK MODELING TECHNIQUE: MULTIPLE AND SINGLE NEURAL NETWORKS

Zainal Ahmad* and Rabiatul 'Adawiah Mat Noor

School of Chemical Engineering, University Sains Malaysia, Engineering Campus,
Seri Ampangan, 14300 Nibong Tebal, Seberang Perai Selatan,
Pulau Pinang, Malaysia.
*Corresponding author. Phone: +604-5937788 ext 6462, Fax: +604-5941013
E-mail: chzahmad@eng.usm.my

ABSTRACT

Neural networks are a very powerful tool for modeling proved by their capability of modeling extremely complex functions and processes. They are very useful in modeling non-linear models that linear models are incapable of. Other features of neural networks that gain interest among researchers are the ease of developing models using neural networks and the fact that they learn through examples. They collect data and carry out an algorithm from the data then automatically learn the structure of the data. The fact that they gain interest among researchers has created a phenomenal atmosphere of research in neural networks field. Therefore, this paper tries to venture into this phenomenal atmosphere of research by reviewing some of the crucial aspects in neural networks research field. One of the shortcomings of neural networks is their lack of robustness and the fact that they need proper network training and validation procedure. Drive by the latter, this paper also reveals the way of improving the robustness of neural networks. Incidentally, employing neural networks scheme as a tool in modeling or process control may be the best way to drag the system to almost perfect circumstance but applying the correct neural networks scheme to the system is one of a very fundamental aspects that need to be mulled. Single neural networks and multiple neural networks can be very useful to both modeling and process control applications. Single neural networks scheme may not give reliable result as perfect neural networks are literally difficult or almost impossible to develop. Thus, combining individual neural networks is the method to elevate the reliability of neural networks by combining their results to give the overall results. The combined neural networks are known as multiple neural networks.

Keywords: Neural Networks, Multiple Neural Networks, Non-linear Process Modeling, Process Control

1. INTRODUCTION

Neural networks have been used for more than fifty years. They started in 1940's where McCulloch and Pitts [Mc Culloch and Pitts, 1943] introduced the idea of studying the computational capabilities of networks composed of simple models of neurons. Hebb [Hebb, 1949] introduced the idea of his unsupervised learning rules which became the root of other development of neural networks. Hebb also came up with the ideas of the reinforcing association between those neurons that are active at the same time.

Neural networks are related to the basic principle of brain [Patterson, 1996] and try to mimic how brain works. They have been developed since 1940 after World War 2 when industrialization was growing rapidly. Neural networks are generally structured in layers of which all the neurons are connected between the adjacent layers. Common layers that build neural networks are input, hidden and output layers. Hidden layer comprises of activation function that converts input to nonlinear output. A typical neural networks structure can be depicted in Figure 1.

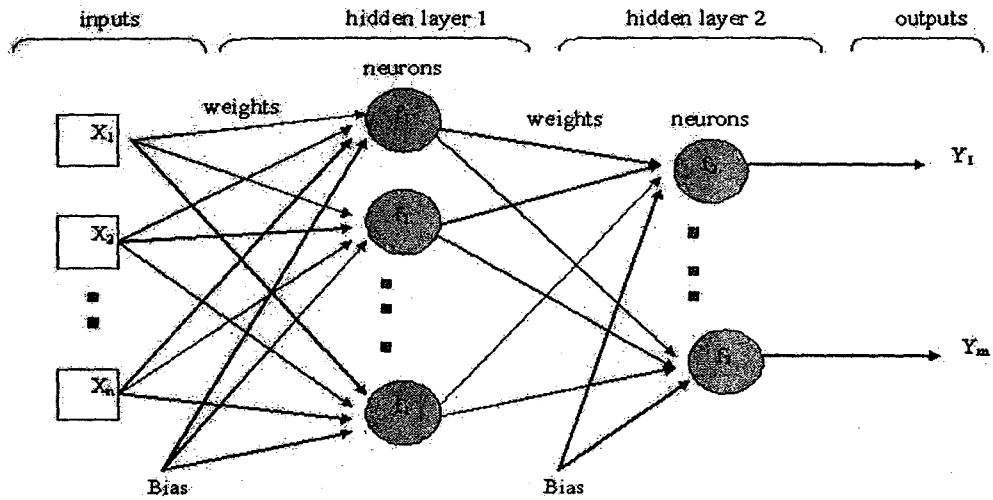


Figure 1: A typical structure of neural networks

After sometimes, neural networks evolved one step ahead by Rosenblat's research [Rosenblat, 1958] in the late 50's where the study of a new concept of neural networks called 'perceptron' also known as 'multilayer feedforwards networks' in today's neural networks' term. Rosenblat also introduced various adaptation rules which include a stochastic technique or what called today as 'backpropagation'. Minsky and Papert [Minsky and Papert, 1967] showed the mathematical limitation of the perceptron and also about the problem behind the perceptron techniques.

During 1970's, a lot of researchers carried out analysis on neural networks especially Grossberg [Grossberg, 1972 and 1973] who attempted to produce differential equation models of various conditioning phenomena. Then, Kohonen [Kohonen, 1982] came up with the idea of feature extraction and clustering which is more to unsupervised learning methods.

In 1980's, neural networks' research field started to show a distinction among the researches. Neural networks research scope can be separated into two major areas. One is by Hopfield [Hopfield, 1982] on the design of associate memories, and later the solution of optimization problems, using the special type of recurrent networks. The other is development based on Rosenblat's ideas on feedforward neural networks, by using differentiable or what we call a 'sigmoidal' activation function. Differentiability makes it possible to employ steepest descent training on the weight (parameter) space, in order to find the neural networks that compute the desire function or interpolate at the desire values. The term 'backpropagation' came up from this ideas and it involves computing the gradients of an error criterion with respect to parameters, via the chain rule, and 'propagating backwards' the vector that correspond to the error at the network output.

Onward in the 1990's there were a lot of researches going on in neural networks like the works of Wolpert [Wolpert, 1992], Sharkey [Sharkey, 1999], Hashem [Hashem, 1997], Sridhar *et al.* [Sridhar *et al.*, 1996] and more. Most of the researches are concentrating on how to increase the robustness of the neural network models either by improving the learning algorithm performance or by improving the generalization capability of the models.

These chronicles have benefit many people since neural networks have become a powerful tool for both modeling and process control applications. Neural networks have been employed in myriad applications such as pattern recognition, speech production, real-time control e.g. [Zhang *et al.*, 1998a; Jayazeri-Rad, 2004], business (e.g. [Fletcher. and Goss, 1993; Desai and Bharati, 1998] and also signal processing (e.g. [Larsson *et al.*, 1996]. In pattern recognition for instance neural networks have been used broadly especially in automated recognition of spoken words (e.g. [Baig *et al.*, 1999; Furlanello *et al.*, 1999], fingerprints identification, handwritten text and moving targets on a static background (e.g. [Srinivas and Kabuka, 1995; Seong-Whan, 1996; Chen *et al.*, 1997]. Even in geology, people tend to applied neural networks scheme as a system to distinguish layers on a bed-to-bed basis so that the complete system can find spatio-temporal eruption from stratigraphic patterns [Bursik and Rogova, 2006].

Neural networks also have been used in manufacturing area where a model was developed to predict fine pitch stencil-printing quality in surface mount assembly. Such model was developed to overcome soldering defect problems that can attribute to solder paste stencil printing process. Neural network model comes as the savior to improve the quality of solder-paste stencil-printing [Yang *et al.*, 2004]. Other than their wide scope of implementation, neural networks also can be combined or embedded into other control schemes such as predictive control, inverse-model-based control and adaptive control [Hussain, 1998].

There are so many evidences to prove that neural networks are technically superior to their competitors. Neural networks are also easy to develop and they learn by themselves through examples. Such advantages have attracted researchers to explore into neural networks fields and successfully make use of them into diverse areas. Despite of the fact that they are superior to their competitors, neural networks also suffer from glitches such as their lack of robustness. Even though neural network models are very powerful non-linear modeling tools, noises in the input data sometimes cause the model over fitting [Mc Loone and Irwin, 2001]. Over fitting and under fitting is the main problem in developing neural network model.

Due to this defect, researchers have come up with some methods to enhance their robustness. Regulation (e.g. [Girosi *et al.*, 1995] and the early stopping method (e.g. [Morgan and Bourlard, 1990] are among the anonymous methods that are suggested. Meanwhile, Ohbayashi [Ohbayashi *et al.*, 1998] implemented the universal learning rule and second order derivatives to increase the robustness in neural network models.

2.0 SINGLE NEURAL NETWORKS

As mentioned earlier in the previous section that neural networks have been widely used not only in engineering filed but also in other applications like in remote sensing (e.g.[Hussein, 1999]), transportation, power system (e.g. [Kiartzis *et al.*, 1997], medicine (e.g. [Lo *et al.*, 1998; Brameier. and Banzhaf, 2001], telecommunication, banking and also application in robotics and vision techniques [Pham and Liu, 1995]. The growing interest in applying neural networks are due to the computing system that growth rapidly which enable the behavior of the complex system to be modeled and predicted accurately. Furthermore, the characteristic of neural network models themselves that learn from examples rather than having to program the complex system also contributed the application of the models. Single neural networks are widely employed both modeling and process control applications. The architecture of single neural networks vary from multilayer perceptron to radial basis function (RBF) and also recurrent neural networks models (e.g. [Hagan *et al.*, 2002].

2.1 GENERAL APPLICATIONS OF SINGLE NEURAL NETWORKS

Most of the applications of neural networks are concentrated on the modeling and control of chemical processes using multilayer perceptron networks. The common systems used in the chemical processes are distillation columns, and reactor systems (continuous stirred tank reactor (CSTR), bioreactor, and neutralizing reactor). These processes are usually very nonlinear and nonlinear models have to be developed. Currently, applications of single neural networks in process modeling and control are quite significant in industry especially in model based predictive control (MBPC) (e.g. [Chen and Yea, 2002; Xiong and Jutan, 2002]) and this is due to the ability of neural networks in modeling nonlinear processes (e.g. [Shaw *et al.*, 1997]).

In process modeling Aldrich and Slater [Aldrich. and Slater, 2001] model the fractional hold-up and drop size in the reactor using single neural networks and the results are quite good. Single neural networks also have been used to predict the heat released by a chemical reactor as developed by Xiong and Jutan [Xiong and Jutan, 2002] as well as Aziz *et al* [Aziz *et al.*, 2001]. The prediction of heat released was compared to the actual heat released in the reactor and the single neural network model performed quite well and promoted to the significant performance on the model predictive control. Other research in chemical reaction in CSTR for examples was done by Shaw *et al* [Shaw *et al.*, 1997] where single neural networks have been used to model the reactor temperature and the result was quite convincing.

Single neural network has also been used to model complex systems in bioprocess, for example by Lobanov *et al* [Lobanov *et al.*, 2001] where single neural networks are used as a biosensor to predict the

glucose and ethanol in certain range of substrate and the accuracy of the estimation was quite good. Scheffer and Filho [Scheffer. and Filho, 2001] apply single neural networks with the extended Kalman filter in the training to predict the production of the penicillin in a batch process. It is shown that the single neural network predictions were quite good even in the real data.

Other applications of neural networks were reported by Lennox *et al* [Lennox *et al.*, 1998] where single neural networks have been used to model a vitrification process using real world data. This process was very nonlinear and single neural networks performed well in predicting and monitoring the vitrification process and at the same time can be employed as a detector to detect any failure in the process.

3.0 MULTIPLE NEURAL NETWORKS

The idea of combined neural networks also known as multiple neural networks was introduce by Wolpert [Wolpert, 1992] where he described about stacked generalization which is a technique for combining different representations to improve the overall prediction performance. It can also be described as an architecture of network consisting of several sub-models and a mechanism which combines the outputs of these sub-models [Eikens. and Karim, 1999]. A typical combined neural network can be depicted in Figure 2 below.

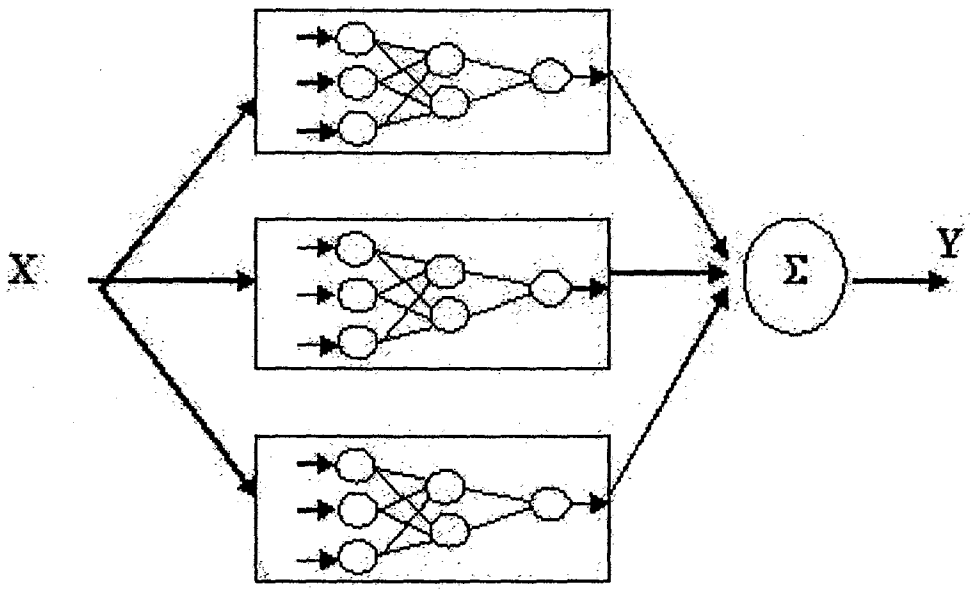


Figure 2: A combined neural networks

There are several types of multiple neural networks but basically they lie underneath the same objective. They made distinction on how to create their sub-model. There are mainly two types of multiple neural networks. The first category is where the training data are totally different in building the individual networks which can be built using different inputs in different regions of operation (e.g. [Chen and Narendra, 2003; Nguyen and Chan, 2004]. The idea of this approach is to adapt different information by using different inputs, and by combining this information a better prediction can be obtained (e.g. [Hashem, 1997; Eikens. and Karim, 1999]. Other multi model approach are introduced by Jacobs [Jacobs *et al.*, 1991] by using the 'mixture of local expert'. Then, Jordan and Jacobs [Jordan and Jacobs, 1994] came up with the hierarchical mixture of neural networks. In this case they basically discuss about the supervised learning algorithm and how the divide and conquer method works. Other is stacked neural networks where it describes as the individual neural networks and trained using different training data sets and /or from different initial weights, then combined [Zhang *et al.*, 1998b]. Instead of choosing the best neural network model among the networks, all the neural networks are combined.

The second category is to creating multiple models using the same training data but re-sampled or partitioned using particular algorithms (e.g. [Zhang, 1999b; Cunningham *et al.*, 2000]. There are three main algorithms being used to re-sample or partition the training data which are bagging or bootstrap (e.g. [Brieman, 994; Zhang, 1999a; Cunningham *et al.*, 2000; Wehrens *et al.*, 2000], adaboost (e.g. [Freund and Schapire, 1996; Schwenk and Bengio, 2000] and randomisation (e.g. [Dietterich, 2000]. The motivation of creating those different inputs or partitions is to create the effective network ensembles [Sharkey, 1999]. Bootstrap application or bootstrap technique was first introduce in 1979 as a computer based method for estimating the standard error of empirical distribution [Dupret. and Koda, 2001]. In neural networks, bootstrap basically relates or deals with the sampling to create random data sets for training and testing. The bootstrap or bagging basically refers to replication of a training data set where the bootstrap algorithm re-samples the original training data set. Some of the data samples may occur several times, and other may not occur in the sample at all. The individual training sets are independent and the neural networks can be trained in parallel.

Adaboost or 'adaptive boosting' on the other hand constructs a composite classifier by sequentially training classifier while putting more and more emphasis on certain patterns [Schwenk and Bengio, 2000]. The probability distribution over the original training data was maintained in this approach where the network is trained with respect to this distribution. In other words the networks are dependent to each other, while randomisation just randomly selects the original training data in each training data and each network can be trained parallel.

3.1 GENERAL APPLICATIONS OF MULTIPLE NEURAL NETWORKS

Multiple neural networks can contribute to almost the same applications as single neural networks. They have been applied in many fields such as in pattern recognition where different models represent different image classification (e.g. [Rogova, 1994; Giacinto and Roli, 2001; Cho and Lee, 2003]. Medical application of multi models is presented by Jerebko *et al* [Jerebko *et al.*, 2003] where different classifications of polyps as single neural network models using different inputs are combined and better prediction rate is obtained. It has also been used in other medical fields like in diagnosis application and in detecting the lung cancer ([Hayashi and Setiono, 2002; Zhou *et al.*, 2002]. Multiple models have also been applied in time series forecasting [Nguyen and Chan, 2004]. In this case each model forecasts a difference time series prediction or prediction horizon and this reduces the recursive prediction promoted to reducing the recursive error occurred in the long range prediction.

3.2 MULTIPLE NEURAL NETWORKS VS. SINGLE NEURAL NETWORKS

Instead of broad implementation of single neural networks, people prone to find better solution with hope that it will attain better results. Research in neural networks area is one of the continuously developed in searching better and better and eventually best solution to gain best results. Lately, researchers discovered that combined individual neural networks improve robustness of the models. The continuous development of computer and it affiliates also seems to spark the development in multiple neural networks. They have been employed ubiquitously in many applications. The fact that they are superior compare to single neural networks also seems to be acknowledged by many people. In fault diagnosis for instance, multiple neural networks already gave convincing results compare to single neural networks where they detected the fault faster than single neural networks [Zhang, 2006]. In fact, they also give the accurate results which is desirable in any filed of applications.

The fact that multiple neural networks give better results compare to single neural networks has spawned the idea of improving robustness of multiple models. As stated in previous chapter, robustness in one of the crucial aspects in developing a 'successful' neural networks model. Overfitting and undefitting are the major problem in applying neural networks. Robustness is technically referred to generalization capability of a neural network model. Therefore, quite a number of techniques have been developed to overcome this defect. Regularization and Bayesian method are among common techniques in improving neural networks' robustness. Despite of these techniques, a combined neural network also has been approved as one of the way to improve overall performance of neural network model [Wolpert, 1992].

Combining the networks improves the generalization capability of the neural networks models in such a way that it guards against the failure of individual components networks. This is because that some of the neural networks will fail to deliver the result or output prediction due to limited training data set (e.g. [Hashem, 1997; Mc Loone and Irwin, 2001]. Many techniques for combining neural networks have been developed since the results are very convincing. Researchers tend to venture into area where methods of improving the combination of neural networks are devised.

Dempster-Shafer belief method is one of the methods used to combine complex model. Selective combination of neural networks has been proposed and seems to attract researchers based on the fact that it reduces the number of shared failure among networks. Other selective combination methods also has been introduced by Perrone and Copper [Perrone and Cooper, 1993] using a heuristic selection method, combining two alternative selection algorithms [Hashem, 1997] and Rogova, Zhang, Ahmad share the same idea by combining less correlated networks to enhance neural networks' output [Rogova, 1994; Ahmad and Zhang, 2003].

4.0 CONCLUSION

Based on the latter section, multiple neural networks are indeed getting popular and most of their application it gives better results as well as improving model's robustness. Even though single neural networks have been proved to be a powerful tool in both modeling and control applications, people will never satisfied on current technology if they can find and successfully develop a better technology in order to improve current technology's performance. This willingness has spawned a brand new technology based on combination of neural networks that in fact give better results and has been benefited many people in this area. Multiple neural networks can be said as the ultimate achievement in this neural networks research field where continuously improvement is welcomed as technology needs a fresh and reliable idea to ease human's life.

ACKNOWLEDGEMENT

This work is supported by University Science Malaysia through short term grant 6035182.

REFERENCES

Ahmad, Z. and Zhang, J. (2003), Selective Combination of Multiple Neural Networks for Improving Long Range Prediction in Nonlinear Process Modelling through Correlation Co-efficient Analysis, in: Proceeding International Conference on Robotics, Vision, Information and Signal Processing.

Aldrich, A. and Slater, M. J. (2001), Simulation of Liquid-Liquid Extraction Data with Artificial Neural Networks. I. M. Mujtaba and M. A. Hussain (Ed). In Application of Neural Networks and Other Learning Technologies in Process Engineering. London, Imperial College Press, 3-22.

Aziz, N., Mujtaba I. M. and Hussain, A. M. (2001), Set Point Tracking in Batch Reactors: Use of PID and Generic Model Control with Neural Networks. I. M. Mujtaba and M. A. Hussain (Ed).In Application of Neural Networks and Other Learning Technologies in Process Engineering. London, Imperial College Press, 217-241.

Baig, A. R., Séguier, R. and Vaucher, G. (1999), A Spatio-temporal Neural Network Applied to Visual Speech Recognition, Proceedings of International Conference on Artificial Neural Networks, 797- 802.

Brameier, M. and Banzhaf, W. (2001)A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining, IEEE Transaction on Evolutionary Computation 5(1), 17-26.

Bursik, M. and Rogova, G., (2006), Use Of Neural Networks and Decision Fusion for Lithostratigraphic Correlation with Sparse Data, Mono-Inyo Craters, California, Computers & Geosciences 32 (2006), 1564–157.

Chen, J and Yea, Y. (2002), Neural Network-Based Predictive Control for Multivariables Processes, Chemical Engineering and Communication 189 (2002), 866-894.

Chen, W., Gader P. D. and Shi H. (1999), Improved Dynamic Programming based Handwritten word Recognition using Optimal Order Statistic, Proceeding of the International Conference Statistical and Stochastic Methods in Image Processing II (1997), 245-256.

Cho, S. B. and J. H. Lee, (2003), Learning Neural Networks Ensembles for Practical Text Classification, Lectures Notes in Computer Science 2690 (2003), 1032-1036.

Cunningham P, J Carney and S Jacob. (2000),Stability Problem with Artificial Neural Networks and the Ensembles Solutions, Artificial Intelligence in Medicine 20 (2000),217-225.

Desai V S and R Bharati. (1998),The Efficiency of Neural Networks in Predicting Returns on Stock and Bond Indices, Decision Sciences 29 (2) (1998) 405-425.

Dietterich T G. (2000), An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, Machine Learning 40 (2), (2000) 139 - 157.

Dupret G and M Koda, (2001), Theory and Methodology Bootstrap re-sampling for Unbalanced Data in Supervised Learning, European Journal of Research 134 (2001), 141-156.

Eikens B and M N Karim, (1999), Process Identification with Multiple Neural Networks Models, International Journal of Control 72 (No.7/8) (1999), 576-590.

Fletcher D and E Goss, (1993), Forecasting with Neural Networks: An Application Using Bankruptcy Data, Information and Management 24 (1993), 159-167.

Freund Y and R E Schapire, (1996), Experiment with a new Boosting Algorithm. Proceeding of 13th International Conference on Machine Learning.

Furlanello C, D Giuliani, S Merler and E Trentin, (1999), Model Selection of Combined Neural Networks Nets for Speech Recognition, Combining Artificial Neural Nets Ensemble and Modular, Amanda J.C Sharkey (Ed) Springer Publication London.

Giacinto G and F Roli, (2001), Design an Effective Neural Networks Ensembles for Image Classification Purposes, Image and Vision Computing 19 (2001), 699-707.

Girosi F, M Jones and T Poggio, (1995), Regularisation Theory and Neural Networks Architecture, Neural Computation 7 (1995), 219-269.

Grossberg S., (1972), Neural Expectation: Cerebellar and Retinal Analogs of Cells Fired by Learnable or Unlearned Pattern Classes, Kybernetik 10 (1972) 49-57.

Grossberg S. (1973), Contour Enhancement, Short-term Memory, and Constancies in Reverberating Neural Networks, Studies in Applied Mathematics 52 (1973) 217-257.

Hagan M T, H B Demuth and O De Jesus (2002), An Introduction to the used of Neural Networks in Control System, International Journal of Robust and Nonlinear Control 12 (2002), 959-985.

Hashem S., (1997), Optimal Linear Combination, Neural Networks 10 (4), 599-614.

Hayashi Y and R Setiono, (2002), Combining Neural Network Prediction for Medical Diagnosis, Computer in Biology and Medicine 32 (2002), 237-246.

Hebb D O., (1949), The Organisation of Behaviour,New York, Wiley.

Hopfield J J., (1982), Neural Networks and Physical System with Emergent Collective Computational Abilities, Proceedings of the National Academy of Science 79 (1982), 2554-2558.

Hussain M. A. (2001), Adaptive Inverse Model Control of a Continuous Fermentation Process using Neural Networks. I. M. Mujtaba and M. A. Hussain (Ed).In Application of Neural Networks and Other Learning Technologies in Process Engineering. London, Imperial College Press, 199-216

Hussein M A., (1999), Review of the Application of Neural Networks in Chemical Process Control-Simulation and Implementation, Artificial Intelligence in Engineering 13 (1999), 55-68.

Jayazeri-Rad H., (2004), The Nonlinear Model Predictive Control of Chemical Plant using Multiple Neural Networks, Neural Computation and Application 13 (2004), 2-15.

Jerebko A K, J D Malley, M Franaszek and R M Summers, (2003), Multiple Neural Networks Classification Schemes for Detection of Colonic Polyps in CT Colonography Data Set, Academic Radiology 10 (2), 154-160.

Jordan M I and R A Jacobs, (1994), Hierarchical Mixtures of Expert and the EM Algorithm, Neural Computation 6 (1994), 191-214.

Kiartzis S, A Kehagias, A Bakirtzis and A Petridis, (1997), Short Term Load Forecasting Using a Bayesian Combination Method,Electrical Power and Energy System 19 (3), 171-177.

Kohonen T., (1982), Self-organised Formation of Topologically Correct Feature Map, Biological Cybernetics 43 (1982), 59-69.

Lobanov A V, I A Borisov, S H Gordon, R V Greene, T D Leathers and A N Reshetilov, (2001), Analysis of Ethanol-Glucose Mixtures by Two Microbial Sensors: Applications of Chemometrics

and Artificial Neural networks for data Processing, Biosensor and Bioelectronics 16 (2001), 1001-1007.

Mc Loone S and G Irwin, (2001), Improving Neural Networks Training Solution Using Regularisation,Neurocomnputing 37 (2001), 71-90.

Nguyen H H and C W Chan, (2004), Multiple Neural Networks for a Long Term Time Series Forecast,Neural Computation and Application 13 (2004) 90-98.

Ohbayashi M, K Hirasawa, K Toshimitsu, J Murata and J Hu, (1998), Robust Control for Nonlinear System by Universal Learning Networks Considering Fuzzy Criterion and Second Order Derivatives, Proceeding of IEEE World Congress on Computational Intelligence.

Patterson D W., (1996), Artificial Neural Networks Theory and Application, Prentice Hall.

Perrone M P and L N Cooper, (1993), When Networks Disagree: Ensembles Methods for Hybrid Neural Networks. R. J. Mammone (Ed), In Artificial Neural Networks for Speech and Vision. London, Chapman and Hall, 126-142.

Pham D T and X Liu, (1995), Neural Networks for Identification, Prediction and Control,Springer.

Rogova G., (1994), Combining the Result of Several Neural Networks Classifiers,Neural Network 7 (5), 777-781.

Rosenblat F., (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review 65 (1958), 386-408.

Schwenk H and Y Bengio, (2000), Boosting Neural Networks, Neural Computation 12 (2000), 1869-1887.

Sharkey A J C. (1999), Multi Nets System, A. J. C. Sharkey (Ed). In Combining Artificial Neural Nets Ensemble and Modular. London, Springer Publication.

Shaw A M, F J Doyle III and J S Schwaber (1997), A Dynamic Neural Networks Approach to Nonlinear Process Modelling,Computer and Chemical Engineering 21 (4), 371-385.

Sridhar D V, E B Bartlett and R C Seagrave, (1999), An Information Theoretic Approach for Combining Neural Network Process Models, Neural Networks 12 (1999), 915-926.

Sridhar D V, E B Bartlett and R C Seagrave, (1996), Process Modelling Using Stacked Neural Networks, AIChe Journal 42 (9), 2529-2539.

Srinivas G and M R Kabuka, (1995), Design of Supervised Classifiers Using Boolean Neural Networks, IEEE Transaction on Pattern Analysis and Machine Intelligence 17 (12), 1239-1246.

Wolpert D H (1992), Stacked Generalisation, Neural networks 5 (1992), 241-259.

Zhang J and M Ishida, (1997), The Multi-step Predictive Control of Nonlinear SISO Processes with a Neural Models Predictive Control (NMPC) Method, Computer and Chemical Engineering 21 (2), 201-210.

Zhang J., (2001), Developing Robust Neural Network Models by Using Both Dynamic and Static Process Operating Data, Ind. Eng. Chem. Res 40 (2001), 234-241.

Zhang J., (1999), Developing Robust Non-linear Models Through Bootstrap Aggregated Neural Networks, Neurocomputing 25 (1999a), 93-113.

Zhang J., (1999), Inferential Estimation of Polymer Quality using Bootstrap Aggregated Neural Networks, Neural Networks 12 (1999b), 927-938.

Zhang J., (2004), A Reliable Neural Network Model Based Optimal Control Strategy for a Batch Polymerization Reactor, Ind. Eng. Chem. Res. 43 (2004), 1030-1038.

Zhang J and A J Morris, (2000), Long Range Predictive Control of Nonlinear Processes Based on Recurrent Neuro-Fuzzy Network Models, Neural Computation and Application 9 (2000), 50-59.

Zhang J., (2006), Improved On-Line Process Fault Diagnosis through Information Fusion in Multiple Neural Networks, Computer and Chemical Engineering, 30 (2006), 558-571.

Improving Multi Step-Ahead Model Prediction through Backward Elimination Method in Multiple Neural Networks Combination

Zainal Ahmad* and Rabiatul 'Adawiah Mat Noor

[1]School of Chemical Engineering, University Sains Malaysia, Engineering Campus, Seri Ampangan,

14300 Nibong Tebal, Penang, Malaysia

E-mail: chzahmad@eng.usm.my

Abstract: Combining multiple neural networks appears to be a very promising approach in improving neural network generalization since it is very difficult, if not impossible, to develop a perfect single neural network. In this paper, individual networks are developed from bootstrap re-samples of the original training and testing data sets. Instead of combining all the developed networks, this paper proposed backward elimination. In backward elimination, all the individual networks are initially aggregated and some of the individual networks are then gradually eliminated until the aggregated network error on the original training and testing data sets cannot be further reduced. The proposed techniques are applied to nonlinear process modeling and application results demonstrate that the proposed techniques can significantly improve model performance better than aggregating all the individual networks.

Keywords : NMPC, multiple neural networks, nonlinear process, feedforward neural networks

1. Introduction

Artificial neural networks have been increasingly used in developing nonlinear models in industry and model robustness is one of the main criteria that need to be considered when judging the performance of neural network models [1, 2]. Model robustness is primarily related to the learning or training methods and the amount and representativeness of training data [3]. Even though neural networks have a significant capability in representing nonlinear functions, inconsistency of accuracy still seems to be a problem where neural network models may not perform well when applied to unseen data. Furthermore, advanced process control and supervision of industrial processes require accurate process models promoting investigations on the robustness of neural network models [4]. Lack of robustness in neural network models is basically due to the over-fitting and poor generalisation of the models (e.g. [5]). Therefore, many researchers have been investigating on how over-fitting can be alleviated through improving network learning algorithms or through combining multiple imperfect neural networks (e.g. [6-11]). In view of improving network learning algorithms, a number of techniques have been developed like regularisation and early stopping method (e.g. [12, 13]). Ohbayashi *et al.* [14] implemented a universal learning rule with second order derivatives to increase the robustness in neural network models.

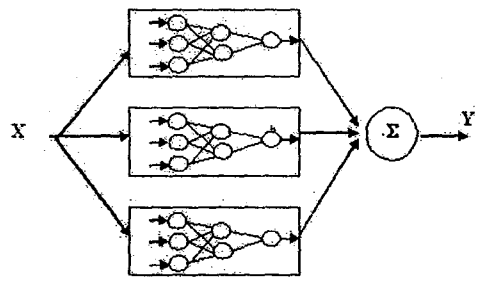


Fig. 1 An aggregated neural network

* Corresponding author

Among those approaches for improving neural network generalisation, the combination of multiple neural networks seems to be very effective. Fig. 1 shows how multiple neural networks are combined. The individual networks in Fig. 1 model the same relationship and are developed from different data sets and/or different training algorithms. They can also have different structures. Instead of choosing the single "best" neural network model, all the individual neural networks are combined. There are a number of methods in combining the networks like stacked neural network and bootstrap aggregated network where multiple networks are created on bootstrap re-samples of the original training data [8, 15-19]. The main objective of this approach is to improve the generalization capability of the neural network models in such a way that it will guard against the failure of individual component networks. This is because of the fact that some of the neural networks will fail to deliver the correct results or output predictions due to network training converged to undesired local minima, over-fitting of noise in the data, or the limited training data set (e.g. [6, 20]).

In most of the reported works on aggregating multiple neural networks, all the developed individual networks are combined. However, some neural networks may not contribute to improving model prediction performance when combined with other networks. This could be due to several reasons, such as these networks severely over-fit the data or the information captured by these networks has already been represented by other networks included in the aggregated network. Excluding these networks could further improve the generalisation capability of the aggregated network. Perrone and Cooper [21] suggests a heuristics selection method whereby the trained networks are ordered in terms of increasing mean squared errors (MSE) and only those with lower MSE are included in combination. However, combining these networks with lower MSE may not significantly improve model generalisation since these networks can be severely correlated. In this paper, backward elimination (BE) methods in statistical regression [12] are proposed for selective combination of neural networks. The paper is organized as follows. Section 2 presents BE selective combination methods for aggregating multiple neural networks. Section 3 presents the case study to test the proposed techniques. Some results and discussions on the case study are given in Section 4. Finally, the last section concludes this paper.

2. Selective Combination of Multiple Neural Networks

Suppose that neural network models are to be developed from the data set $\{X, Y\}$, where $X \in R^{N \times p}$ is the input data, $Y \in R^{N \times q}$ is the output data, N is the number of samples, p is the number of input variables, and q is the number of output variables. To develop an aggregated neural network model containing n individual networks, the original data set can be re-sampled using bootstrap re-sampling with replacement to form n replications of the original data set [15]. The n replications can be denoted as $\{X_{(1)}, Y_{(1)}\}$, $\{X_{(2)}, Y_{(2)}\}$, ..., $\{X_{(n)}, Y_{(n)}\}$, where $X_{(i)} \in R^{N \times p}$, $Y_{(i)} \in R^{N \times q}$, $i=1, 2, \ldots, n$. A neural network model can be developed on each of these replications, which can be partitioned into a training data set and a testing data set if cross-validation is used in network training and network structure selection. If the predictions of these n networks on the original data set are denoted as $\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}n$, then the sum of squared errors (SSE) of the ith network can be calculated as

$$SSE_i = \text{trace}[(Y - \hat{Y}_i)^T (Y - \hat{Y}_i)] \tag{1}$$

For the sake of simplicity in illustration, the simple average method is used in combining the selected networks. If all n networks are combined, then the aggregated network output is:

$$\hat{Y} = \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_i \tag{2}$$

2.1 Backward Elimination

The BE approach begins with the aggregated neural network containing all the individual networks and removes one network at a time until the SSE on the training and testing data cannot be further reduced. The network deleted at each step is such selected that its deletion results in the largest reduction in the aggregated network SSE on the training and testing data. The BE method is summarised as follows:

Step 1 Generate n replications of the original data set using bootstrap re-sampling, $\{X_{(1)}, Y_{(1)}\}$, $\{X_{(2)}, Y_{(2)}\}$, ..., $\{X_{(n)}, Y_{(n)}\}$, and develop a neural network on each replication. Denote the prediction of the ith network on the original data set as $\hat{Y}i$. Calculate the SSE of these networks on the original data using Eq (1).

Step 2 Set j=1 and denote I as a set containing the indices of the networks currently included in the aggregated network and I=[1, 2, ..., n]. Denote J as a set containing the indices of the networks currently deleted from the aggregated network and J=[], i.e. J is initially empty. Denote $\hat{Y}_{a,j}$ and SSE(j) as, respectively, the predictions and SSE of the aggregated network at stage j.

$$SSE(j) = \text{trace}[(\frac{1}{n}\sum_{i\in I}\hat{Y}_i - Y)^T(\frac{1}{n}\sum_{i\in I}\hat{Y}_i - Y)]$$

Step 3 If n-j=0, then go to Step 5;

else
j=j+1
for i∈I

$$\hat{Y}_{a,j}^{(i)} = \frac{1}{n-j}\sum_{l\in I-i}\hat{Y}_l$$

end

$$k = \arg\min_{i\in I}\text{trace}[(\hat{Y}_{a,j}^{(i)} - Y)^T(\hat{Y}_{a,j}^{(i)} - Y)]$$

$$SSE(j) = \text{trace}[(\hat{Y}_{a,j}^{(k)} - Y)^T(\hat{Y}_{a,j}^{(k)} - Y)]$$

Step 4 If SSE(j)≥SSE(j-1), then go to Step 5;
else
I=I - k (i.e. remove k from set I)
J=[J, k]
go to Step 3.

Step 5 Stop

2. Case Study

The case study chosen is pH neutralization process. The neutralization process takes place in a CSTR and there are two input streams to the CSTR as shown in Figure 1. One is acetic acid of concentration C_1 at flow rate F_1 and the other is sodium hydroxide of concentration C_2 at flow rate F_2 [23]. The mathematical equations of the CSTR can be found in reference [23]. To generate training, testing and validation data, multi-level random perturbations were added to the flow rate of acetic acid while other inputs to the reactor were kept constant.

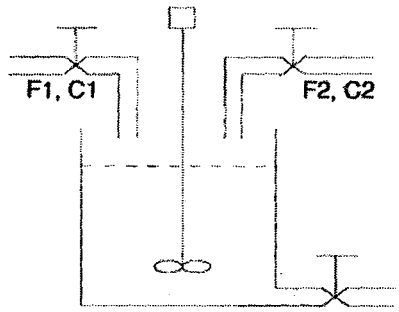


Fig. 2 CSTR for pH neutralisation process

The pH measurements were corrupted with normally distributed random noise with zero mean and a standard deviation of 0.2. The dynamic model representing the neutralization process is of the form:

$$\hat{y}(t) = f[\hat{y}(t-1), \hat{y}(t-2), u(t-1), u(t-2)] \tag{3}$$

where $\hat{y}(t)$ is the pH prediction in the reactor at time t and $u(t)$ is the acid flow rates at time t.

For long range predictions or multi-step-ahead predictions, the current and past model predictions are used to predict the future values of the model outputs:

$$\hat{y}(t) = f[\hat{y}(t-1), \hat{y}(t-2), ..., \hat{y}(t-n), u(t-1), u(t-2), ...u(t-m)] \tag{4}$$

where the model prediction, $\hat{y}(t-1)$ to $\hat{y}(t-n)$, are used in place of the process outputs, $y(t-1)$ to $y(t-n)$, to predict $\hat{y}(t)$ as shown for pH prediction in Eq (3).

In this case study, 20 networks with fixed identical structure and 20 networks with various structures were developed and the individual networks were trained by the Levenberg-Marquardt optimisation algorithm with regularisation and "early stopping". The individual networks are single hidden layer feed forward neural networks. Hidden neurons use the sigmoid activation function whereas output layer neurons use the linear activation function. To cope with different magnitudes in the input and output data, all the data were scale to zero mean and unit standard deviation. The data for neural network model building need to be divided into: 1). Training data (for network training); 2). Testing data (for cross-validation based network structure selection and early stopping); and 3). Unseen validation data (for evaluation of the final selected model). In networks with fixed structure, the network structures, i.e. the number of hidden neurons, were determined through cross validation. Single hidden layer neural networks with different numbers of hidden neurons were trained on the training data and tested on the testing data. The network with the lowest SSE on the testing data was considered as having the best network topology. In assessing the developed models, SSE on the unseen validation data is used as the performance criterion.

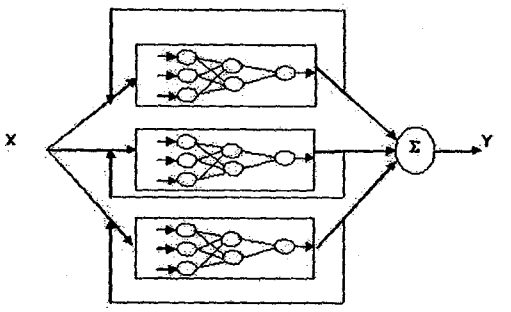


Fig. 3 Long range prediction with feedback before the combination of individual networks

Accurate long range predictions are much more difficult to achieve than accurate one-step-ahead predictions due to the accumulation of the errors in the recursive predictions [22]. To obtain long range predictions from an aggregated network, two types of network output feedback schemes can be used but only feedback as shown in Fig. 3 is used. This is equivalent to combining the long range predictions of individual networks. To test the performance of the proposed selective combination schemes, the following combination schemes are investigated:

Median	: Median of the individual networks;
Average	: Average of all networks;
BE	: Average of selected networks using the BE method.

In order to further study the capability of the proposed method, five additional runs with different initial network weights were carried out. These different initial weights were generated using different seeds in the MATLAB random number generator and applying different scaling factors to the generated random numbers.

4. Result and Discussion

It is well known that the dynamics of pH is highly nonlinear. In this case study 20 networks with fixed number of hidden neurons (5) and 20 networks with varying number of hidden neurons (between 1 and 10) were developed. Again in the fixed structure, the number of hidden neurons was determined through cross validation. Fig. 4 shows the long range prediction performance of individual neural networks. It can be seen from Fig. 4 that the individual networks give inconsistent long range prediction performance on the training and testing data and on the unseen validation data. For example in Fig. 4 shows that network number 14 among the networks with various structures gives the worst performance on the training and testing data. However, its performance on the unseen validation data is quite good. This demonstrates the non-robust nature of individual networks.

Fig. 5 shows the SSE of long range predictions from aggregated neural networks with various structures. The aggregated networks under selective combination scheme give quite consistent prediction performance on the training and testing data and on the unseen validation data. This patent was also observed for the fixed structure.

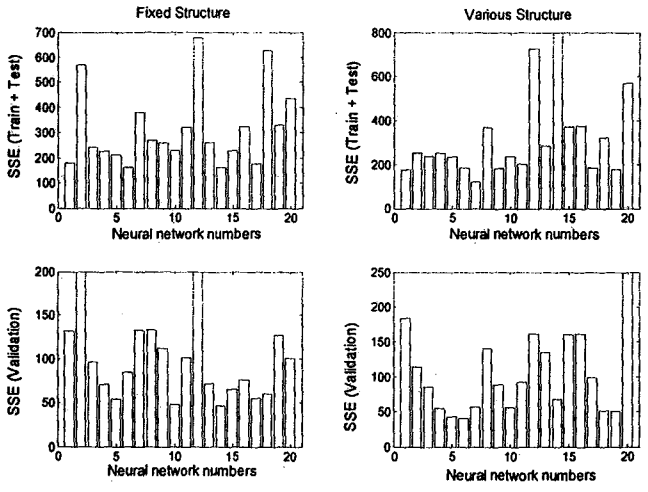


Fig. 4 SSE of long range predictions from individual neural networks in pH neutralization process

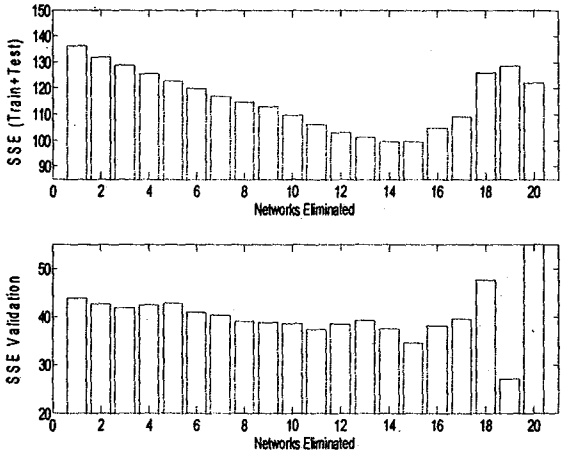


Fig. 5 SSE from aggregated neural networks with various structures in pH neutralization process

Table 1 gives the SSE on the unseen validation data of different combination schemes. It can be seen that the worse one of BE selective combination schemes gives better performance than combining all the networks and the median of individual networks. In the BE selection methods 5 networks (networks 1, 6, 11, 14, and 17) and 7 networks (networks 1, 5, 7, 11, 17, 18, and 20) were combined for fixed and various structures. The median of the individual network SSE on the unseen validation data for fixed and various structures are 90.44 and 90.52 respectively.

Table 1. Overall Results for pH Neutralisation Process

Combination schemes			SSE on validation data
Median	Fixed structure		90.44
	Various structures		90.52
Average	Feedback before combination	Fixed structure	57.31
		Various structures	43.84
BE	Feedback before combination	Fixed structure	41.77
		Various structures	37.44

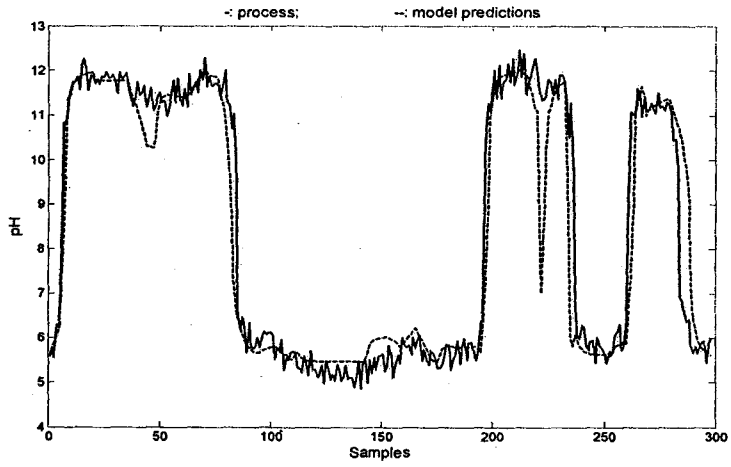


Fig. 6 Long range predictions from the best aggregated neural network combination

Table 2 Mean and Standard Deviation When Varying the Parameter in Neural Network Modeling

Combination schemes			Mean	Std Deviation
Median	Fixed structure		93.48	3.52
	Various structures		94.43	4.38
Average	Feedback before combination	Fixed structure	59.16	4.99
		Various structures	51.36	4.41
BE	Feedback before combination	Fixed structure	50.47	3.14
		Various structures	38.37	1.29

The best combination scheme in this case is "BE with fixed structures with feedback before combination" with an SSE of 37.44 on the unseen validation data. Fig. 6 shows the long range predictions from this aggregated neural network. Lastly, the initial parameter was change in order to test whether the proposed methods can get a consistent result even though some of the condition is different. The result is quite consistent for BE selection method where the mean and standard deviation is smaller compare to median and averaging methods as shown in Table 2.

5. Conclusions

Backward elimination methods for the selective combination of multiple neural networks are proposed in this paper in order to improve the model generalization performance. In the BE method, initially all individual networks are included in the aggregated network. Individual networks are then eliminated one at a time from the aggregated network until the aggregated network error on the original training and testing data cannot be further reduced. BE selective combination methods have shown their superiority compared to the combination of all networks and the median in this case study.

Acknowledgements: The author would like to thank USM(grant no : 6035182)) for its continuous support and Dr Jie Zhang from Newcastle University, UK for providing the additional information on the Matlab™ simulation.

References

[1]. D. H. Wolpert. 1995. Stacked Generalization. Neural Networks 5: 241-259.

[2]. Heartz J A, Krogh A, and Palmer R G. 1991. Introduction to the Theory of Neural Network Computation. Redwood City, CA: Addison-Wesley: 1991

[3]. Bishop C, 1995. Neural Networks for Pattern Recognition. Oxford: Clarendon Press.

[4]. Zhang J. 2001. Developing Robust Neural Network Models by Using Both Dynamic and Static Process Operating Data. Ind.Eng.Chem.Res 40: 234-241.

[5]. Caruana R, Lawrence S and Giles Lee C. 2000. Overfitting in Neural Networks: Backpropagation, Conjugate Gradient and Early Stopping. Neural Information Processing System 13: 402-408.

[6]. Hashem S. 1994. Optimal Linear Combination. Neural Networks 10(4): 599-614.

[7]. Sharkey A J C. 1999. Combining Artificial Neural Nets Ensemble and Modular in Multi Nets System. Sharkey A J C Ed. Springer Publication.

[8]. Sridhar D V, Bartlett E B and Seagrave R C. 1999. An Information Theoretic Approach for Combining Neural Network Process Models. Neural Networks 12: 915-926.

[9]. Sridhar D V, Bartlett E B and Seagrave R C. 1996. Process Modelling Using Stacked Neural Networks. AIChe Journal 42(9): 2529-2539.

[10].Zhang J. 1999. Inferential Estimation of Polymer Quality using Bootstrap Aggregated Neural Networks. Neural Networks 12: 927-938.

[11]. Laar P D. 2000. Input selection Based on an Ensembles. Neurocomputing 34: 227-238.

[12].Hagiwara K and Kuno K. 2000. Regularization Learning and Early Stopping in Linear Networks. In proceeding: International Joint Conference on Neural Networks (IJCNN 2000): 511-516.

[13].Morgan N and Bourlard H. 1990. Generalization and Parameter Estimation in Feedforward Nets: Some Experiments. In Advances in Neural Information Processing System 2, Touretzkey D S Ed. San Mateo CA: 630-637.

[14].Ohbayashi M. 1998. Robust Control for Non-linear System by Universal Learning Networks Considering Fuzzy Criterion and Second Order Derivatives. In Proceeding: 1998 IEEE World Congress on Computational Intelligence: 968-973.

[15].Zhang J. 1999. Developing Robust Non-linear Models through Bootstrap Aggregated Neural Networks. Neurocomputing 25: 93-113.

[16].Zhang J. 1998. Prediction of Polymer Quality in Batch Polymerization Reactors Using Robust Neural Networks. Chemical Engineering Journal 69: 135-143.

[17].Brieman L. 1994. Bagging Predictors (Tech.Rep.No.421). Department of Statistic University of California at Berkeley, USA.

[18].Franke J and Neumann M H. 2000. Bootstrapping Neural Networks. Neural Computation 12: 1929–1949.

[19].Dietterich T G. 2000. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Machine Learning 40(2): 139 - 157.

[20].Mc Loone S and Irwin G. 2001. Improving Neural Networks Training Solution Using Regularization. Neurocomnputing 37: 71-90.

[21].Perrone M P and Cooper L N. 1993. When Networks Disagree: Ensembles Methods for Hybrid Neural Networks. In Artificial Neural Networks for Speech and Vision. Mammone R J Ed. Chapman and Hall. London : 126-142.

[22].Zhang J, Morris A J and Martin E B. 1998. Long-term Prediction Models Based on Mixed Order Locally Recurrent Neural Networks. Computers and Chemical Engineering 22(7-8): 1051-1063.

[23].Mc Avoy T J, Hsu E and Lowenthal S. 1972. Dynamics of pH in Controlled Stirred Tank Reactor. Ind. Chem. Process Des. Dev 11: 68-70.

Studying the Effect of CSTR Capacity and Stirring Rate in Controlling pH Neutralization Process in WWT Plant

Zainal Ahmad[*,1] and Sharlinda Salim Sachithananthan[2]

[1,2]School of Chemical Engineering, University Sains Malaysia, Engineering Campus,
Seri Ampangan, 14300 Nibong Tebal, Seberang Perai Selatan,
Pulau Pinang, Malaysia.
*Corresponding author. Phone: +604-5995999 ext 6462, Fax: +604-5941013
E-mail: chzahmad@eng.usm.my

Abstract

This paper focused on factors that improved the performance of PID controller in controlling pH neutralization process which is frequently used in treating wastewater. The performance of PID controller at different tank capacities and stirring rates had been studied in order to find the optimum tank capacity and stirring rate in pH controlling system. Strong acids, 0.01M H_2SO_4 and strong base, 0.1M NaOH were react in different tank capacities at constant stirring speed followed by using different stirring speeds at constant tank capacity. The PID controller performance was based on set points tracking and disturbance rejection. In the largest tank, the controller showed an excellent set point tracking with minimum oscillations and shortest time in rejecting disturbance. An almost the same result were displayed by using the highest stirring rate while poor performances in controller response were observed in a lowest tank capacity and stirring rate. Thus, it can be concluded that larger tank dampen out oscillations and reduced the effect of disturbance while at a higher stirring rate, it gave a complete mixing and thus accurate measurement of pH which improved significantly the controller performance.

Keywords: pH control, PID controller, CSTR, capacities, stirring rates

1.0 Introduction

The pH control is very important in many processes. In the wastewater treatment and the cell growth rate the accurate stabilization of pH at an optimal level often determines the efficiency of the bioprocesses. According to nowadays legislation the restrictions on effluent discharges are very tight and pH control in these discharges becomes more important. Heavy metals must be recovered and one frequently employed method is to control pH to minimize the solubility of the metals. The most common pH process is the neutralization of an acidic or a basic waste stream for the next reasons: (i) to prevent corrosion (ii) to protect the aquatic life and the human welfare according to ecological low, (iii) as a preliminary treatment in bioprocess (iv) to provide neutral pH water for recycle.

On the chemical process side pH neutralization is a very fast and simple reaction. However it was recognized as a very difficult control problem which arises mainly from the strong process

nonlinearity (the process gain can change ten or more times) and the time varying properties due to frequent load and/or component concentration changes. Several strategies have been proposed for a nonlinear pH control. Lee and Choi (2000) propose a simple nonlinear adaptive control system for pH processes which uses the in-line mixer to control the pH value in the stirred-tank reactor. Regunath and Kadirkamanathan (2001) had presented a fuzzy non-uniform grid scheduling approach for controlling pH neutralization process while Faanes and Skogestad (2004) addresses control related design issues, such as tank sizes and number of tanks, for neutralization plants. Go´mez *et al.* (2004) presented a Wiener model identification and predictive control of a pH neutralisation process. Akesson *et al.* (2005) studied the computational issues of model predictive control (MPC) of nonlinear sampled-data systems and also neural network approximation of nonlinear model predictive controllers. Syafiie *et al.* (2007) applied model-free learning control (MFLC), based on reinforcement learning algorithms and hierarchical reinforcement learning. Altınten (2007) studied pH control of a neutralization process by using generalized predictive control (GPC) method. The below researches were focused on treating wastewaster by controlling pH.

Ishak et.all. (2001) study the dynamics and control of a semibatch wastewater neutralization process in modeling and simulation with a digital PI control algorithm was used as the controller, and the control simulation was performed in Matlab's Simulink environment. The control studies were done to include the effect of changes in process dead time, base concentrations and base flowrates to the controllability of the semibatch system. From the simulation study, it was found that an increase in process dead time would result in process instability while, an increase in base concentration and flow rate would result in faster neutralization time. For a given set of condition, the process dead time gave no effect to the volume of wastewater accumulated in the tank. Naohiro (2003) investigate the effectiveness of oxidation-reduction potential (ORP), pH and dissolved oxygen as parameters for indicating denitrification followed by nitrification in sequencing batch reactors (SBRs) for swine wastewater treatment. With a low C/N ratio, and using a suitable C/N ratio adjustment control, ORP and pH could be used as monitoring and control parameters in both the anoxic and oxic phases for practical swine wastewater treatment. Most of the studies on the effect of pH on enhanced biological phosphorous removal were conducted with the acetate wastewater, and the pH was controlled during the entire anaerobic and aerobic stages but Yan Liu (2006) investigated the influence of anaerobic initial pH control, which will be more practical than the entire process pH control strategy, on enhanced biological phosphorus removal from wastewater containing acetic and propionic acids. The optimal initial pH for higher soluble ortho-phosphorus (SOP) removal efficiency should be controlled between 6.4 and 7.2. This pH control strategy will be easier to use in practice of wastewater treatment plant. Zeybeck (2006) presents an experimental application of AHCC to study the coagulation process of wastewater treatment in a dye plant. Also this study includes a series of tests in which an AHCC control was used for pH control. The performance results of the AHCC controller are compared with the results obtained by using a conventional proportional-integral-derivative (PID) algorithm. Although the removal of pollutants from wastewater is similar with AHCC and PID, the results show excellent AHCC performance in the region where conventional PID control fails.

In this paper two scopes of studies related to performance of pH control in a continuous stirred tank reactor (CSTR) by using a PID controller were investigated. Case Study 1 is to learn the effect of different CSTR capacities in controlling pH. In simple words, this study is to verify whether a small or a larger tank should be used in mixing of acid and base in order to achieve a certain value of pH. It will also observe the performance of PID controller at achieving different set points and rejecting disturbance in the system. In Case Study 2, the effect of controller performance at different stirring rates will be studied. The controller performance in reaching the set points and rejecting disturbance is compare at higher and lower agitation rates.

2.0 Experimental Setup

A schematic sketch of the experimental set up is shown in Figure 1. It consists of a 2L continuous stirred tank reactor (CSTR), supply tanks, pumps, pH electrodes (pHE), pH transmitter (pHT), a recorder (pHR) and a controller (pHIC). Agitation is provided in the reactor by means of a mechanical stirrer.

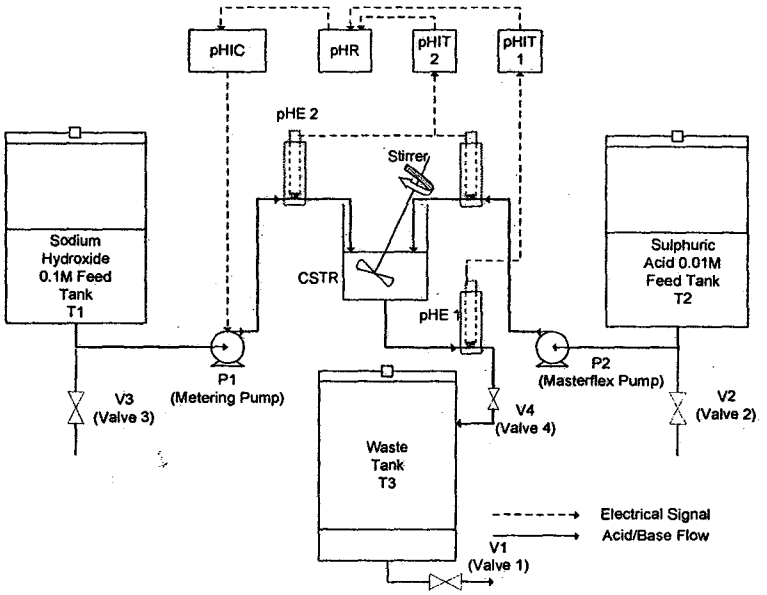


Figure 1: Schematic sketch of the experimental set-up

This experiment was run to control a pH process between H_2SO_4 and NaOH in a continuous stirred tank reactor (CSTR) with PID open loop controller. The influent acid stream acted as disturbance, influent alkaline stream as manipulated variable while the controlled variable was the pH value of the effluent. The concentrations of influents were used at constant 0.01M of H_2SO_4 in Tank 1 and 0.1M of NaOH in Tank 2. First, the CSTR was filled in until 1.0L of H_2SO_4 at constant flow rate. At the controller, an auto mode was chosen and set point was set at pH 7. The stirrer was start at constant 350 rpm and the metering pump was switch on. Automatically, the base flow rates were manipulated until it slowly reached pH 7 and remained steady. After several minutes at steady state condition, the controller mode was changed to manual. Step changed was introduced by increasing the

base flow rate and a new steady state was achieved. This data was plotted to obtain a setting for the PID controller by using s-curve method. This PID setting will be applied through out the experiment. Again, the CSTR was filled in with 1.0L of H_2SO_4 at stirring speed of 350rpm. By using an auto mode controller, pH 7 was set as set point and metering pump was switch on. As base flows in the tank, the tank volume was maintained at 1.0L by manually adjusting the exit valve. Once the set point was achieved, it was remained steady for a while before new set points was set at pH 9 followed by pH 5. This is repeated at different tank capacity of 0.5L and 1.5L. Next, the tank capacity was remained at 1.0L but the stirring rates were changed from 350rpm to 450rpm and 150rpm. All the data recorded in the recorder were then plotted.

3.0 Result and Discussion

This section discussed the results of studying the effect of tank capacities as Case Study 1, and stirring rates as Case Study 2, in controlling pH with PID controller. The controller performances in both studies were evaluated based on its response to set point changes and rejecting disturbance.

3.1. Case Study 1: Reactor Tank Capacities

The set points in pH adjustment processes are usually at the steepest part of the titration curve, near the neutral pH of 7. The process has extremely high gain or sensitivity at this point meaning a small amount of changes in reagent will cause remarkable changes in pH value. To study the effect of tank capacities on the system response, three different pH tracking were used. The servo was varied after 30 minutes at each set point from pH 7 to 9 followed by pH 5. Figure 2 shows the performance of the pH process under different tank capacities for servo control problems with the corresponding base flow rate is given in Figure 3.

When the set point tracking behavior of each tank was compared, the system was more capable of bringing the pH to the set points in the largest tank, 1.5L. By using this tank, PID showed a faster response with minimum oscillations and over/undershoots and settling times less than 10 minutes toward the set point than did in the 1.0L and 0.5L tank as in Figure 2. The errors caused by the set point changes were instantaneously sensed by the controller and immediate corrections in the base flow were taken as can be seen in Figure 3. This was probably because in 1.5 L tank capacity with 350 rpm stirring rates, the mixture of acid and base was well mixed. As base flows in the tank, the pH changes gradually resulting excellent pH control at each set point.

In 1.0L tank with 350rpm stirring rates, PID also showed good control performance but with oscillations and over/undershoots especially at set point pH 5. This was due to severe changes of base flow rates into the reactor had caused the pH suddenly changes until the controller finally reached a suitable flow rates of base. This had taken a longer time especially in obtaining an acidic solution of pH 5. As the controller was manipulating the base flow rates, a high amount of base was used.

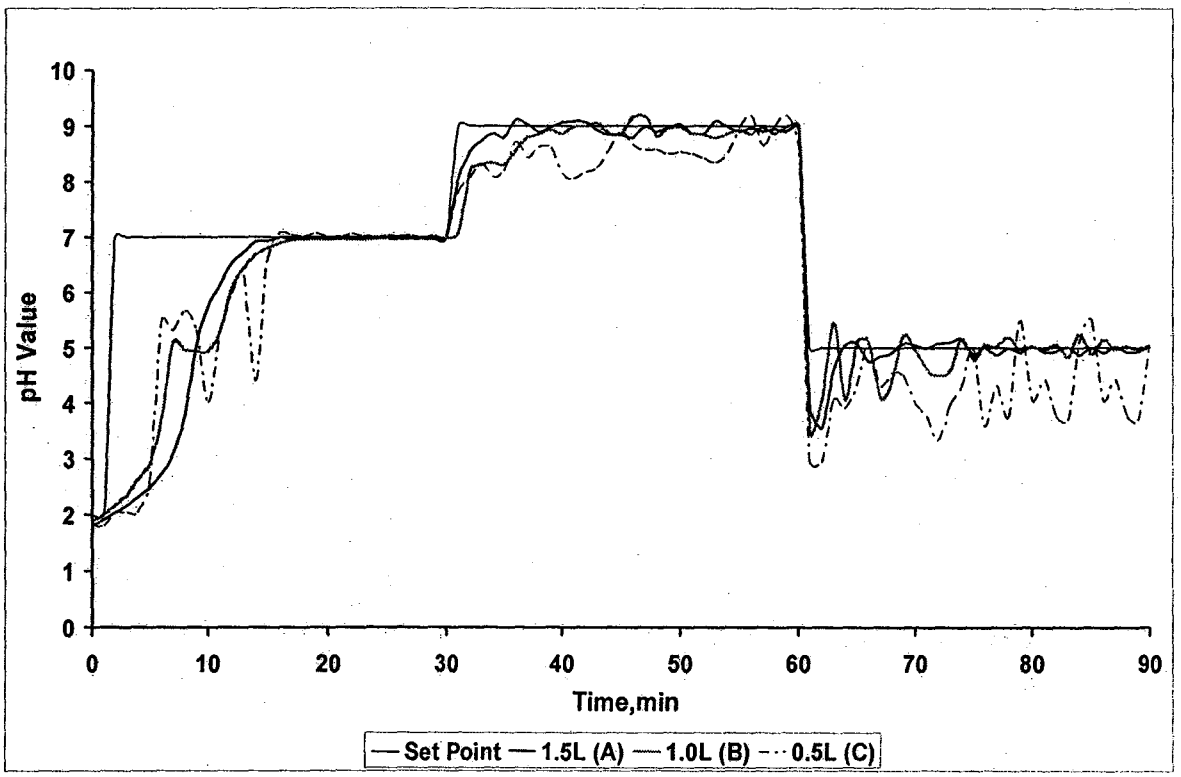


Figure 2: Set point changes at different tank capacities, Time (min) versus pH value

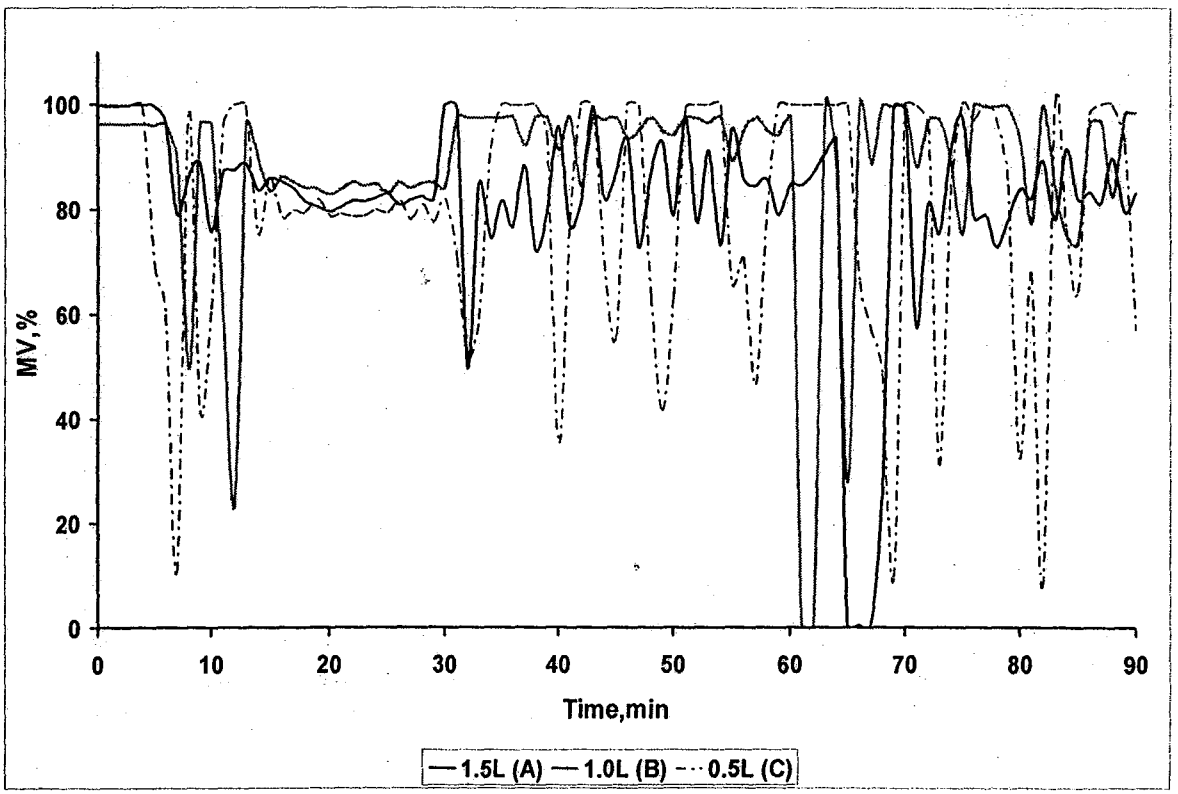


Figure 3: Changes of base flow rate, MV % versus time (min)

Meanwhile, in tank of 0.5L, PID showed very slow response with extremely poor performance. This was most probably because as the tank volume is smaller, minor changes of base flow rates will caused major changes of pH in the mixture. The controller could not obtain the suitable amount of base needed in this small volume of mixture as the pH was fluctuating. A high amount of base had been used in this process without success.

In order to assess the robustness of the controller at different capacities of tanks, its ability to maintain the pH value of the effluent stream at the neutral value of pH 7 in the presence of disturbances was examined. Instead of keeping the acid flow rate constant, it was changed from 1.5 ml/s to 3.4 ml/s for 15 seconds. It was disturbed respectively at time 30 minutes which was after it reached steady state at pH 7. This characteristic is important in applications, such as waste-water treatment, where disturbances should not cause the pH value of the effluent stream to deviate too much from the set point.

Figure 4 displayed the drop of pH caused by disturbance while Figure 5 showed the controller performance in rejecting disturbance. Tank A showed a small change of pH value because the disturbance was introduced only for a short time of 15 seconds, thus it took only 4 minutes to return back to pH 7. The pH value of Tank B drop to pH 5 with 7 minutes needed to recover back to the process pH while Tank C took 10 minutes to trace the set point from pH 3.5. All three different tank capacity managed to reject the disturbance but at different range of time.

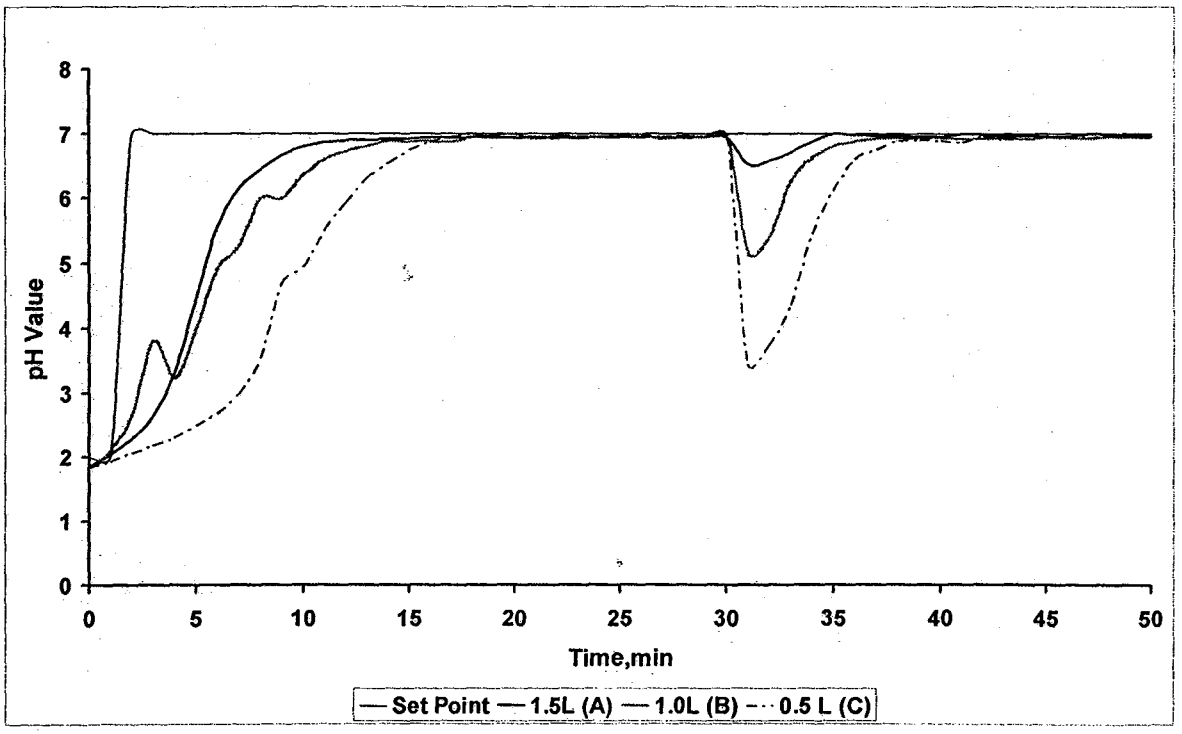


Figure 4: Disturbance effect at different tank capacities, Time (min) versus pH value

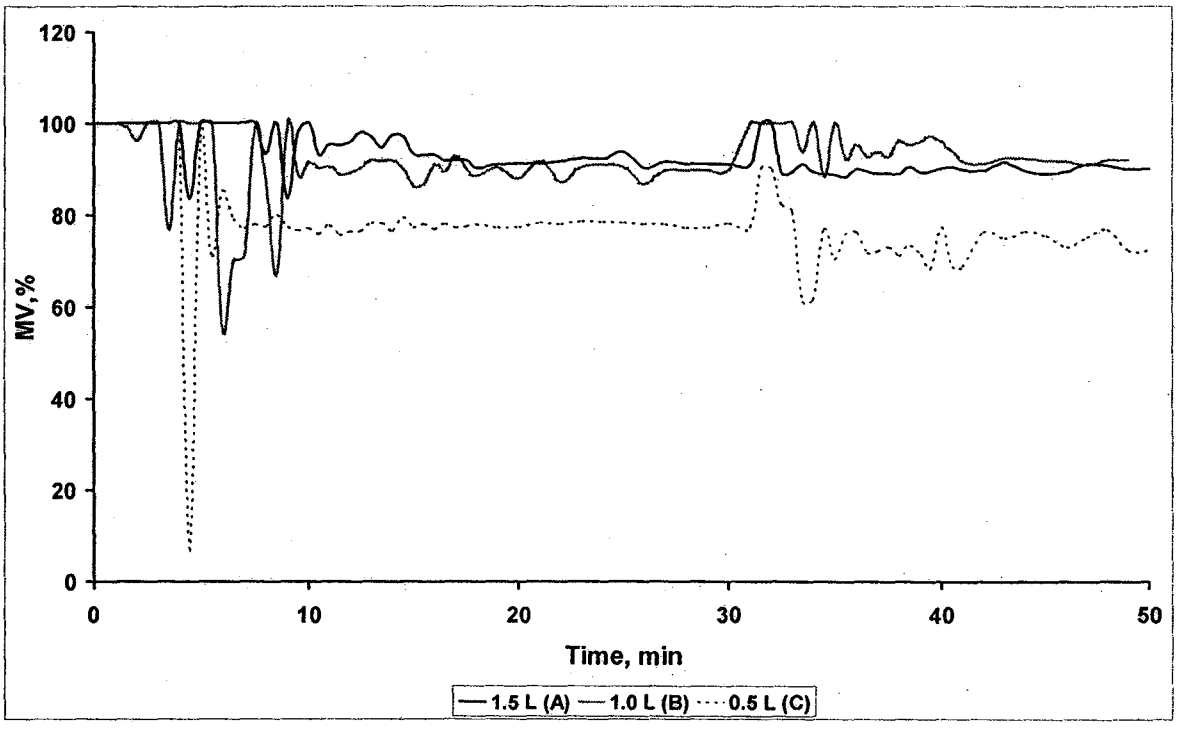


Figure 5: Changes of base flow rate, MV % versus time (min)

3.2. Case Study 2: Stirring Rates

The set points were changed similar to the first experiments which were pH 5, 7 and 9 to evaluate the controller response at different stirring rates. The servo was varied after 30 minutes at each set point from pH 7 to 9 followed by pH 5. Figure 6 shows the control of pH at different stirring rates with the corresponding base flow rate is given in Figure 7.

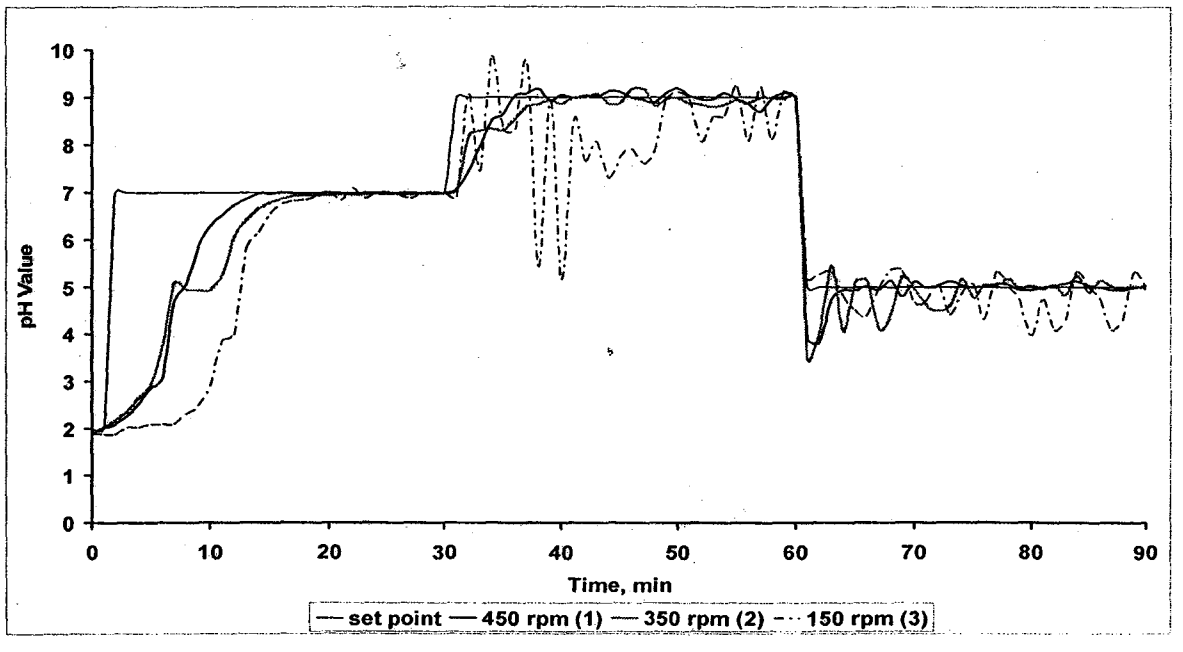


Figure 6: Set point changes at different stirring rates, Time (min) versus pH value

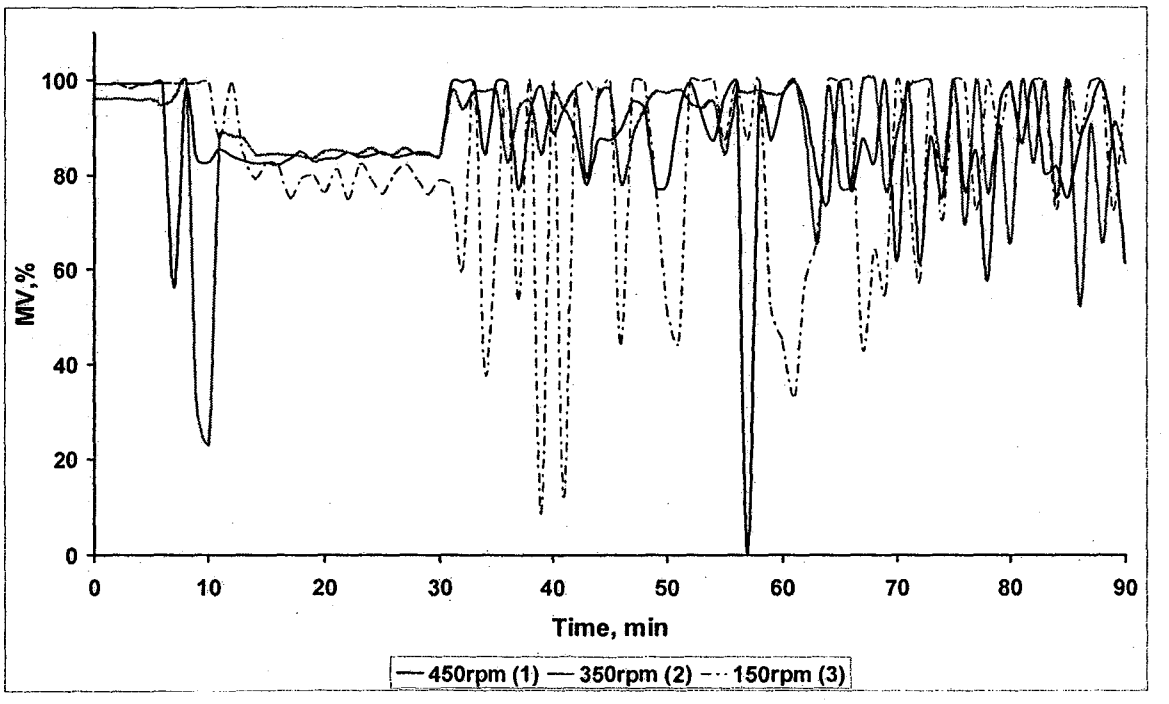


Figure 7: Changes of base flow rate, MV % versus time (min)

Overall, at 450rpm in 1.0L tank, the controller was able to control the pH value at different set points with minimum oscillations and over/undershoots. This resulted from a complete mixing in which acid and base molecules were dissociated completely and the pH changes are instantaneous. This vigorous mixing ensures uniform composition through out the reaction tank thus an accurate measurement of pH was obtained and consequently an easier and accurate control of pH value. Referring to Figure 7, it can be seen that the controller was able to manipulate and maintained the base flow rates efficiently at each set point.

At rate 350rpm, the set point tracking response was not as good as at 450rpm especially at pH 5. In obtaining pH 5, the flow rates of base was decreased, thus stirring rate plays a crucial role in mixing this less amount of base in the mixture in a short time. This is probably the reason at 350rpm, the mixing was not fully complete and thus more time is needed to perfectly mix the acid-base mixture. Therefore, a longer time needed by the controller to reach the set points.

The mixing process with rate 150rpm was inadequate therefore it exhibited a poor controller performance. Inadequate mixing resulted inaccurate reading of pH measurement. Since this controller responded based on error of measurement and set point, it will proceed with its action. Therefore, as shown in Figure 7, several actions taken were inappropriate such as low amount of base were feed in, in order to obtain an alkaline solution of pH 9.

To study the controller ability to maintain the pH value of the effluent stream at the neutral value of pH 7, disturbance was introduce by increasing the acid flow rate for 20 seconds. The result was showed in Figure 8 and the controller response as in Figure 9.

From the below Figure 8, at Rate 3, 150rpm, the pH changes faster followed by Rate 2 and then Rate 1. However, the rejection of disturbance at the three different rates was almost the same, after 10 minutes, the disturbance was rejected and back to its initial set point. Therefore, even though Rate 3

showed faster changes of pH, the time it took to reach the original pH is the same as Rate 1. Thus, Rate 1 can still be considered as the best stirring rates in rejecting disturbance.

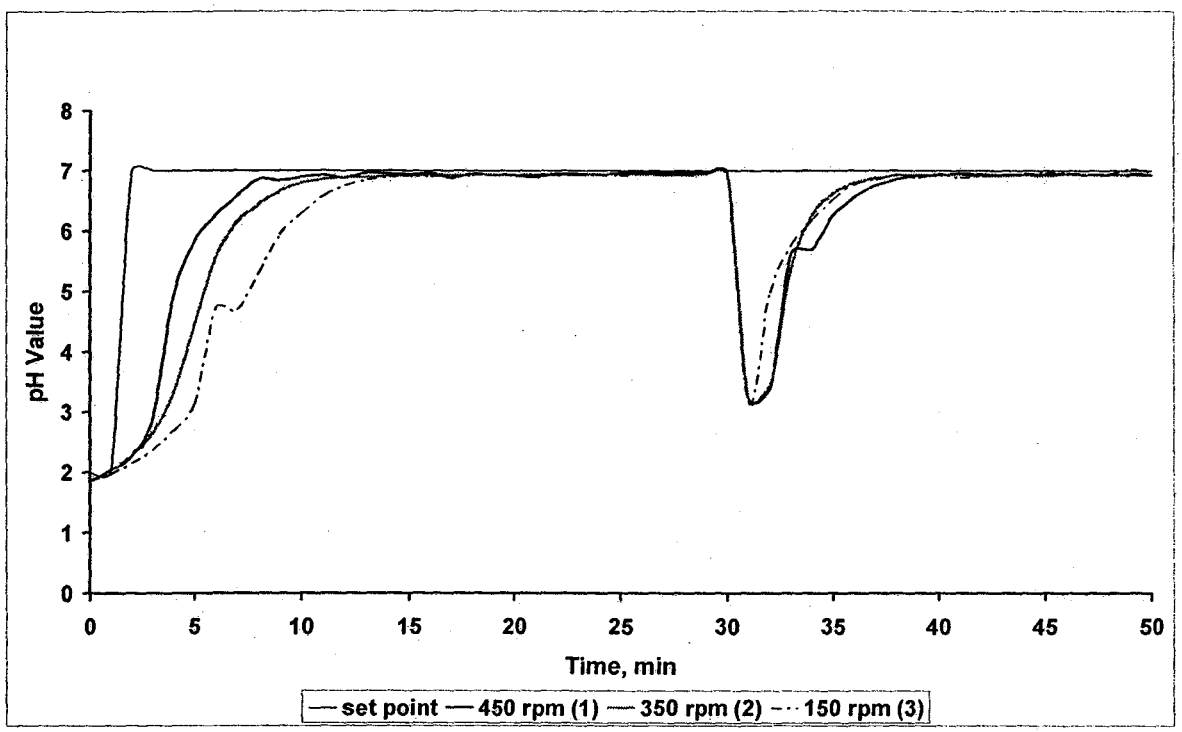


Figure 8: Disturbance effect at different stirring rates, Time (min) versus pH value

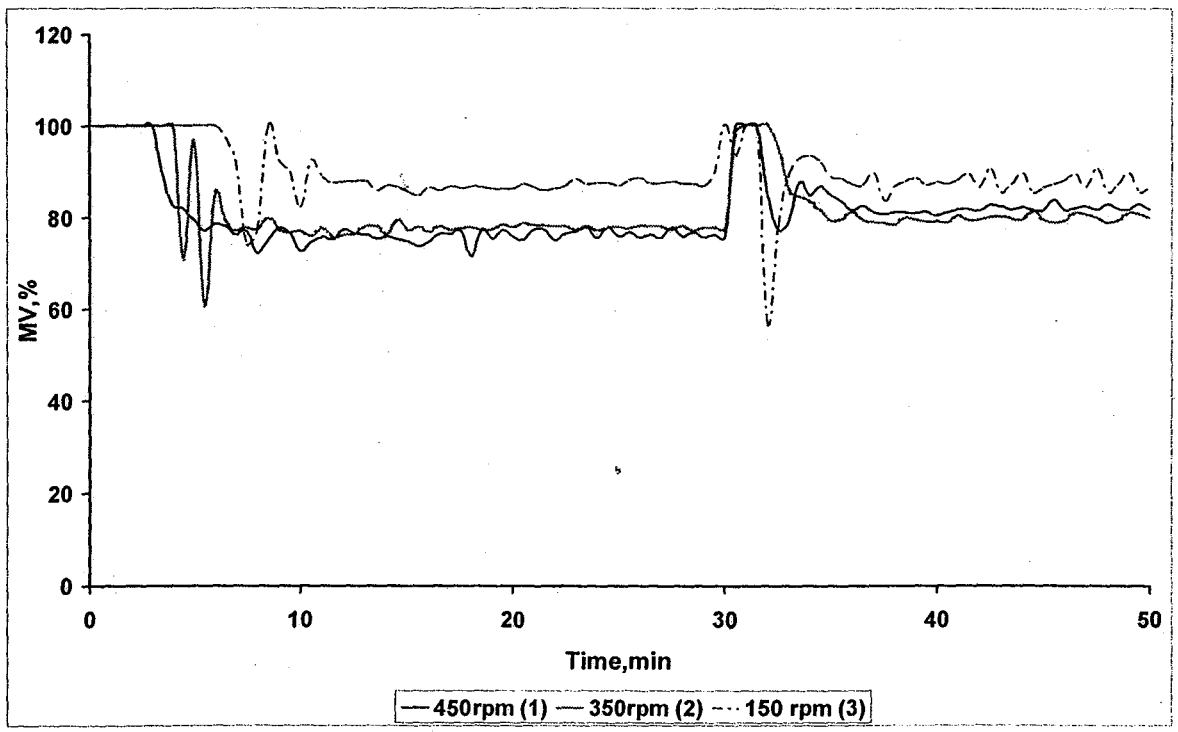


Figure 9: Changes of base flow rate, MV % versus time (min)

4.0 Conclusion

4.1. Case Study 1: Reactor Tank Capacities

High capacity is favorable for effective control since it levels out abrupt changes and gives time for mixing. It is concluded that Tank A, 1.5L with stirring rates of 350rpm, is the optimum tank capacity in controlling pH of effluent with a PID controller. It gives excellent set point tracking from neutral (pH 7) to alkaline (pH 9) and acidic mixture (pH 5) with minimum oscillations. Moreover, it reduces the effect of disturbance forming a good disturbance rejection in a short time. It also minimizes the amount of reagent required in the process.

4.2. Case Study 2: Stirring Rates

Agitation serves the purpose of equalizing the hydrogen or hydroxide concentration profile within the reaction vessel as the influent is dispersed in the tank. The optimum stirring rate in controlling pH is Rate 1 at 450rpm. The controller performance in tracking set point and rejecting disturbance was good due to complete mixing of acid and base. It blends all reactants efficiently in minimum time and ensures all the tanks contents are well blended thus obtaining an accurate measurement of pH which leads to an easier controlling of pH. A faster control of pH will minimizes the amount of reagent needed.

Acknowledgement

This research is supported by University Science Malaysia through short term grant 6035182

List of References

Abdul Aziz Ishak, Mohamed Azlan Hussain2 & Elamin Elkanzi (2001). Modelling and control strategies of wastewater neutralization process. *Proceeding to Brunei International Conference on Engineering & Technology,221-231*

Åkesson B.M., Toivonen' H.T. , Waller J.B. and Nyström R.H. (2005). Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process. *Computers & Chemical Engineering,* **29**(2), 323-335

Altınten A. (2007).Generalized predictive control applied to a pH neutralization process. *Computers & Chemical Engineering,* **31**(10), 1199-1204.

Go´mez J.C., Jutan A. and Baeyens E. (2004). Wiener model identification and predictive control of pH neutralization process. *IEE Proc.-Control Theory Appl.*, **151**(3)

Lee J. and Choi J.Y. (2000). In-line mixer for feedforward control and adaptive feedback control of pH processes. *Chemical Engineering Science*, **55**(7), 1337-1345

Naohiro Kishida, Ju-Hyun Kim, Meixue Chen, Hiroshi Sasaki and Ryuichi (2003). Effectiveness of oxidation-reduction potential and pH as monitoring and control parameters for nitrogen removal in

swine wastewater treatment by sequencing batch reactors. *Journal of Biosciense and Bioengineering*, **96**(3), 285-290

Regunath S. and Kadirkamanathan V. (2001). Design of a pH control system using fuzzy non-uniform grid scheduling and evolutionary programming. *Applied Soft Computing*, **1**(1), 91-104

Skogestad S. and Faanes A. (2004). pH-neutralization: integrated process and control design. *Computers & Chemical Engineering*, **28** (8), 1475-1487

Syafiie S.,Tadeo F. and Martinez E. (2007). Model-free learning control of neutralization processes using reinforcement learning. *Engineering Applications of Artificial Intelligence*, **20**(6), 767-782

Yan Liu, Yinguang Chen, and Qi Zhou (2007). Effect of initial pH control on enhanced biological phosphorus removal from wastewater containing acetic and propionic acids. *Chemosphere*,**66**(1),123-129

Z. Zeybek, S. Yüce Cetinkaya, F. Alioglu and M. Alpbaz (2007). Application of adaptive heuristic criticism control (AHCC) to dye wastewater. *Journal of Environmental Management*, **84**(4), 461-472.

APPENDIX C

DR. ZAINAL AHMAD

304.PJKIMIA.6035182

JUMLAH GERAN :- 18,479.00

NO PROJEK :-

PANEL :- J/PENDEK

PENAJA :- JANGKA PENDEK

UNIT KUMPULAN WANG AMANAH
UNIVERSITI SAINS MALAYSIA
KAMPUS KEJURUTERAAN
SERI AMPANGAN

PENYATA KUMPULAN WANG

TEMPOH BERAKHIR 30/04/2008

Tempoh Projek:01/04/2006 - 31/03/200

NONLINEAR PROCESS MODELLING OF Ph NEUTRALIZATION PROCESS IN CSTR USING SELECTIVE

Vot	Peruntukan (a)	Perbelanjaan sehingga 31/12/2007 (b)	Tanggungan semasa 2008 (c)	Perbelanjaan Semasa 2008 (d)	Jumlah Perbelanjaan 2008 (c + d)	Jumlah Perbelanjaan Terkumpul (b+c+d)	Baki Peruntukan Semasa 2008 (a-(b+c+d)
11000: GAJI KAKITANGAN AWAM	5,864.00	4,071.35	0.00	0.00	0.00	4,071.35	1,792.65
21000: PERBELANJAAN PERJALANAN DAN SARAHI	2,000.00	1,741.50	0.00	1,280.00	1,280.00	3,021.50	(1,021.50)
23000: PERHUBUNGAN DAN UTILITI	200.00	130.00	0.00	0.00	0.00	130.00	70.00
27000: BEKALAN DAN ALAT PAKAI HABIS	7,115.00	4,432.10	0.00	1,550.00	1,550.00	5,982.10	1,132.90
28000: PENYELENGGARAAN & PEMBAIKAN KECIL	800.00	0.00	0.00	2,330.00	2,330.00	2,330.00	(1,530.00)
29000: PERKHIDMATAN IKTISAS & HOSPITALITI	2,500.00	2,935.00	0.00	0.00	0.00	2,935.00	(435.00)
	18,479.00	13,309.95	0.00	5,160.00	5,160.00	18,469.95	9.05
Jumlah Besar	18,479.00	13,309.95	0.00	5,160.00	5,160.00	18,469.95	9.05