Process Control System Identification

By

MOHAMED ELSADIG MOHAMED

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

PROCESS CONTROL SYSTEM IDENTIFICATION

by

Mohamed Elsadig Mohamed

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Ms. Suhaila Badarol Hisham
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2005

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Mohamed Elsadig Mohamed

ABSTRACT

The main purpose of this project is to develop black-box models. That can introduce another modeling tool in process control system identification field. These models need to address process that is difficult or complex to model using the mathematical or empirical modeling approaches. The core activity for this project is to develop a model for chemical process (single loop pressure control) using black-box identification techniques. This approach overcomes the difficulties that encountered in modeling processes that characterized with low level of a priori knowledge (operators have no priori knowledge about the system). Process control system identification is implemented through conducting an experiment on a gas /air pressure pilot plant where a set of input-output data are collected. Intelligent black-box modeling techniques are implemented for building process model that can be utilized for further process control applications. Feedforward Neural network and fuzzy clustering method are used to obtain a model from the plant data. The author developed and compared the performance of black-box modeling techniques against the performance of empirical model. According to the comparison achieved. It is recommended to use black box modeling techniques for processes where process control engineers may face the lack of knowledge about the system or the plant.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and thanks to my thesis supervisor Ms. Suhaila Badarol Hisham, for introducing me to the field of process control system identification, she is the one who gives me the interest and support to curry on the project phase by phase.

Especial thanks to Mr.Azhar Zainal Abidin, The UTP Pilot Plant Supervisor, for willingness to help giving his long experience in explaining the plant operation, dynamics and issues related with the experiment conducted for this project.

In addition, I would like to thank all UTP, electrical and electronic department lecturers who never hesitate in answering my inquiries, also I would like to thank my family for the words of encouragements that gives me the strength and support.

Finally, I would like to extend my appreciation to Mr. Ali Azizan Maamor, the external examiner from Petronas Carigali SDN BHD, who offered valuable and insightful comments about the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	– Artificial neural network
MV	- Manipulated Variable
PV	- Process Variable
BP	-Back Propagation
FM	- fuzzy model
NN	- Neural network

CHAPTER 1
INTRODUCTION

1.1 Project Background

Process control system identification is achieved through a number of modeling techniques. The conventional modeling techniques which was most common in this trends, seems to be not sufficient enough or it comes to the time where looking for alternatives of old modeling techniques. That makes the process control system identification easier from the implementation aspects.

In another word, depending on the level of a priori knowledge about the system, the identification problem can be approached in different ways. If the identification is based exclusively on the system's measured data such technique is called black-box modeling.

Assuming there is no or only diminutive knowledge about the physics, chemistry and dynamics of the system. The identification techniques handling such lack of knowledge about the system (process) dynamic are called black-box modeling. In contrast to this is white-box modeling. That is used the pure physics and chemistry laws of the systems in order to develop a model. But when a certain level of insight about the system exists and is utilized to improve the empirical modeling the phrase gray-box modeling is used.

Another problem that might exist that is even though knowledge about the process is available but it is difficult to utilize it as it relates to a continuous-time description of the system (i.e., in term off differential equations). In this case transferring knowledge to a discrete–time description is often hard and thus it is frequently lost in discretization process. For this case modeling option will be confined to black-box identification regardless that a certain level of system insight is available. However, a certain level of system's behavior (system dynamics) is always useful tool to facilitate the identification job. Such insight might include: the order of the system, whether the system dynamics are slow or fast, stability properties, operation range, and time delay.

An important stage in control system design is the development of a mathematical model of the system to be controlled. In order to develop a controller, it must be possible to analyze the system to be controlled and this is done using a mathematical model. Another advantage of system identification is evident if the process is changed or modified. System identification allows the real system to be altered without having to calculate the dynamical equations and model the parameters again. The mathematical model in this case is the black-box, it describes the relationship between the input and output signals.

Process identification approach helps in system modeling using different techniques of modeling. One of these techniques is using a black-box system identification technique. Where the main idea of this kind of modeling is to construct process models from identification data (set of input-output).

The purpose of the model is to emulate the relation between system's input-output data. To model a certain process designs an experiment to collect the identification data process data (identification data), assume (select) model structure, and calculate model parameters then evaluating performance of the model or remodeling if necessary.

Process modeling has a significant improvement in the process control system industry. Process control engineers are in need for an accurate model (model with minimum errors). That represents the process dynamics. This model facilitates the implement of different control strategies. Where doing such experiment on the running plant might be impossible due to the operation condition and the product quality.

In brief, the process model can be used for control application, fault detection, prediction, etc.

1.2 Objective

In short, the project aims is to meet the following objectives:-

1) To conduct experiment on UTP Pilot Plant (SIM 305 pilot plant 2: air flow, pressure & temperature pilot plant.), in order to collect data for identification (record of plant input-output).
2) To apply fuzzy logic and neural network in order to obtain "black-box" process models (build model from the process data).

1.3 Scope of the Project

The "Black-box" system identification techniques used in this project are limited to feedforward neural network and fuzzy clustering methods. Plant data set (input-out) is used to build a model that can emulate the plant's dynamics behavior.

Neural network model constructed by using Neural Network toolbox in MATLAB, then the produced model is trained and simulated (model validation).

Fuzzy clustering (building fuzzy model from partition) is implemented using fuzzy modeling and identification toolbox developed by Dr. Robert Babuska. [1]

1.4 Problem Statement

The project addresses modeling practice where difficulties are face in the following aspects:

1) Difficulties to implement process data in order to obtain empirical model.
2) Lack of the knowledge about the system. That enables process control engineers to develop a mathematical model from the physical and chemistry principles (obtain models from the fundamental laws of physics and chemistry).

The black-box model developed in this project can be implemented for developing control strategies, fault detection, etc.

Chapter 2

LITERATURE REVIEW

2.1 System Identification

System identification is the task of inferring a mathematical description or a model of a dynamic system from a series of measurements on the system.

If burden associated with building a model using laws of physics, chemistry economics, etc., is considered overwhelming, system identification techniques are naturally of particular interest.

2.2 Empirical Modeling

Empirical modeling is an alternative modeling method to the mathematical modeling method for process control system. A model developed using this method provides the dynamic relationship between the selected plants's input - output variables.

Empirical model involve design experiments, during which the process is perturbed to generate dynamic response. The success of the model requires close adherence to the principles of experiment design and model fitting. There are two method of model fitting. The first method is termed the process reaction curve which employs simple graphical procedures for model fitting. The second and more general method employs statistical principles for determining the parameters.

2.3 Neural Network

According to Chin-Teng Lin and C.S Gorge Lee:

> Neural Networks are a promising new generation of information processing systems that demonstrate its ability to learn, recall and generalize from training patterns of data. [2]

In another word, neural network can be trained using plant real time data sets [MV PV], the data or the observed plant input-output.

Another general definition for Neural Network, according to Norgaard, Ravan, Poulsen and Hansen:

> Neural network is a system of simple processing elements, neurons that are connected into a network by a set of (synaptic) weights. [3]

Adjustments can be done to determine the function of the network. Such adjustments can be in term of selecting the architecture of the network, the magnitude of weights and the processing element's mode of operation.

2.3.1 Basic Models and Learning Rules of ANNs

Models of ANNs are specified by three basic entities: models of the neurons themselves, models of synaptic interconnections and structures, and the training or learning rules for updating the connecting weights

Processing elements (PE), Information processing of the PE can be viewed as structure consisting of two parts: input and output .associated with the input of the PE is an integration function f which service to combine the information, activation or evidence from an external source or PEs into a net input to PE. The second action of the PE is to output an activation value as a function of its net input through an activation function or transfer function (f).

Connections, an ANN consist of highly interconnected PEs such that each PE output is connected through weights to other PEs or to it self, both delay and lag-free connections are allowed. Hence the structure that organize these PEs and the connection geometry among them should be specified for an ANN. It is also important to point out where the connection originates and terminates in addition to specifying the function of each PE in an ANN. refer to Figure 1below.

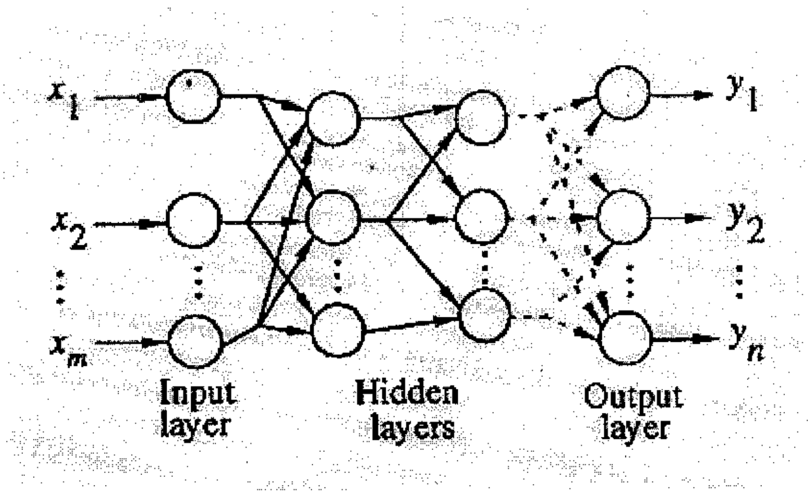


Figure 1 Basic network connection geometries (for feedforward network)

Learning rules, specifying an ANN learning rule is an important design aspect, generally there are two kinds of learning in ANNs: *parameter learning* which is concerned with updating of the connecting weights in an ANN, and *structure learning* which focuses on the change in the network structure, including the number of PEs and their connection types. in general learning rules are classified as supervised , unsupervised and reinforcement learning , attention is paid to supervised learning as learning rule for the neural network model.

In supervised learning, at each instant of time when input is applied to an ANN, the corresponding desired system output response is given as (**d**). The neural network is thus told precisely what it should be emitting as output. More clearly, in the supervised learning mode an ANN is supplied with a sequence of examples, $\left(x^{(1)},d^{(1)}\right),\left(x^{(2)},d^{(2)}\right),\mathrm{K}\ ,\left(x^{(k)},d^{(k)}\right)$ of the desired input-output pairs , when each input $x^{(k)}$ is put into the ANN, the corresponding desired output $d^{(k)}$ is also supplied to the ANN, the difference between the actual output (neural network predicted output) $y^{(k)}$ and the desired output is measured in the error signal generator which then produces the error signal for the ANN to correct its weights in such a way that the actual output will move closer to the desired output. Figure 2 below shows the representation of the supervised learning rule.

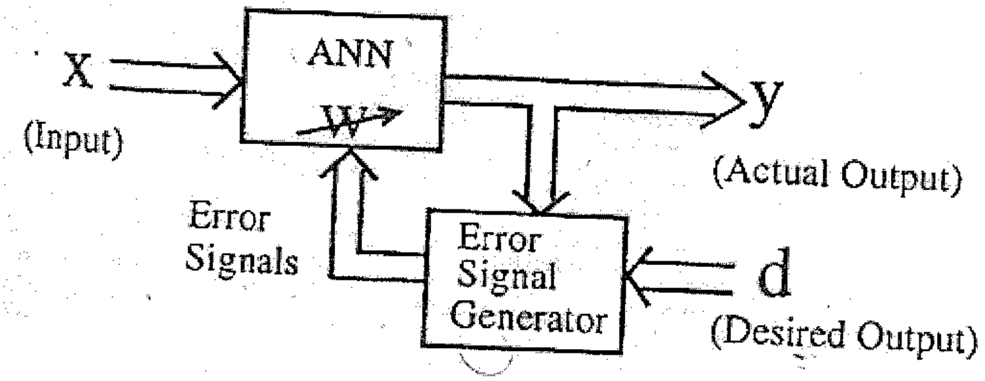


Figure 2 Supervised learning

In supervised learning, it is assumed that the correct "target" output values are known for each input pattern. However, in some situations only less detailed information is available. For example, the ANN may only be told that the current actual output is "too high" or 50% "correct". In extreme case, there is only a single bit of feedback information indicating whether the output is right or wrong. Learning base on this of critic information is called reinforcement learning, and the feedback information is called the reinforcement signal. Reinforcement learning is a form of supervised learning because of the network still receives some feedback from its environment. But the feedback (i.e., the reinforcement signal)is only evaluative (critic) rather than instructive .that is just indicate how good or how bad a particular output is and provide no hints as what the right answer should be . The external reinforcement signal is usually processed by the critic signal generator produce more informative critic signal for the ANN to adjust its weights properly with the hope of getting better critic feedback in the future. Figure 3 below shows the representation of reinforcement learning rule. The reinforcement learning is also called learning with critic as opposed to learning with a teacher, which describes supervised learning rule.

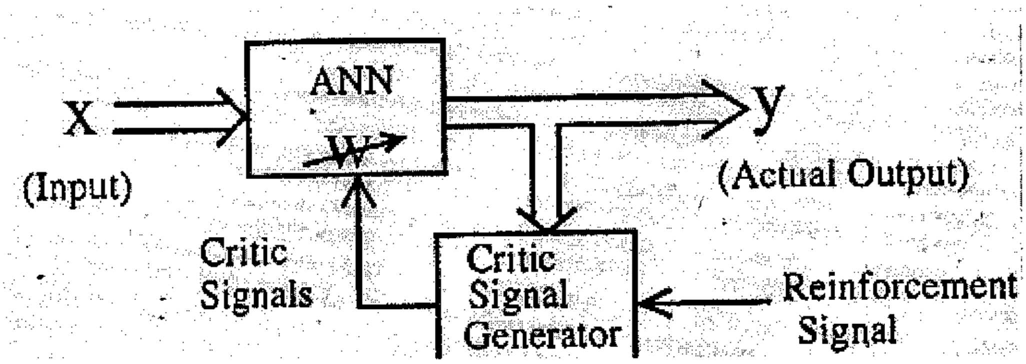


Figure 3 :Reinforcement learning

Learning algorithm (back-propagation), it is applied to multilayer feedforward neural networks consisting of processing elements with continuous differentiable activation functions. Such networks associated with the Back-Propagation learning algorithm are called back-propagation neural networks. Given a training set of input-output pairs $\{(x^{(k)}\ ,d^{(k)})\}$, k=1,2,...,p, the algorithm provide a procedure for changing the weights in the back-propagation network to classify the given input patterns correctly. The basis for this weight update algorithm is simply the gradient-descent method as used with simple perceptron with differential units. For a given input-output pair $\{(X^{(k)}, d^{(k)})\}$, the back-propagation algorithm performs two phases of data flow. First, the input pattern $X^{(k)}$ is propagate from the input layer to the output layer and , as a result of this forward flow of data , it produces an actual output $y^{(k)}$. Then the error signal resulting from the difference between $d^{(k)}$ and $y^{(k)}$ are back-propagate from the output layer to the previous layers in order to update their weights.

2.4 Fuzzy logic

According to Robert Babuska:
The concept of fuzzy-set theory and fuzzy logic can be employed in the modeling of systems in a number of ways. [4]

Examples of fuzzy systems are rule-based fuzzy systems, fuzzy linear regression models.

Fuzzy rule-based systems, are system where the relationship between the variables are represented by a means of fuzzy if-then rule

If *antecedent proposition* **then** *the consequent proposition*

Example :

if *valve is fully open* **then** *pressure is very high*

If *valve is half open* **then** *the pressure is medium*

If *valve is fully close* **then** *the e pressure is low*

Depending on the structure of the consequent proposition three types of models are distinguished:

1. Linguistic fuzzy model, where both antecedent and consequent are fuzzy sets.
2. Fuzzy relational model, which can be regarded as generation of the linguistic model, allowing one particular antecedent proposition to be associated with several different consequent proposition via fuzzy relation.
3. Takagi-Sugeno (TS) model, where the consequent is a crisp function of the antecedent variables rather that the fuzzy proposition.

2.4.1 Fuzzy modeling

System can be represented by mathematical models of many different forms, such as algebraic equations, differential equation, etc. the modeling frame work considered for this project is based on rule-based model fuzzy model, which describe the relationships between the variables by means of if-then rules, such as:

if the heating power is high ***then*** the temperature will increase fast .

2.4.2 Fuzzy identification

According to Robert Babuska:

> The term fuzzy identification usually refers to techniques and algorithm for constructing fuzzy models from data. Two main approaches for the integration of knowledge and data in fuzzy model can be distinguished. [4]

1. The expert knowledge expressed in verbal form is translated into collection of if-then rules. In this way, a certain model structure is created. Parameters in the structure (membership functions, weights of the rules, etc.) can be fine-tuned using input-output data. The particular tuning algorithms exploit the fact that at the computational level. A fuzzy model can be seen as a layered structured (network), similar to artificial neural networks, to which standard learning algorithm can be applied. This approach usually called neuro-fuzzy Modeling.

2. If no prior knowledge about the system under to formulate the fuzzy rules. In such case fuzzy model is constructed using numerical data only. It is expected that the extracted rules and membership functions can provide a posterior interpretation of the system's behavior. An expert can comfort this information with his own knowledge, can modify the rules. Or supply new ones and design additional experiments in order to obtain more informative data.

This project focus on the implementation of the second method (fuzzy *model is constructed using numerical data only (input-output data set)) - automated fuzzy model from data.* In many cases a natural requirement in a model not only accurately predicts the system's output but also provides some insights of the system dynamics.

For this project exercise, fuzzy models is viewed as a class of local Modeling approaches which attempt to solve a complex Modeling problem by decomposing it into a number of similar sub-problems.

Since it cannot be expected that sufficient prior knowledge is available concerning this decomposition method for automated generation of the decomposition, primary from system /plant data set are developed. A suitable class of fuzzy clustering algorithms is used for this purpose. Moreover, additional techniques are proposed for the reduction and simplification of the initial model acquired from data.

2.4.3 Fuzzy clustering

According to Robert Babuska:

> Fuzzy clustering is an effective approach to the identification of complex nonlinear systems, where the available data is partitioned or clustered into subsets and approximate each subset by simple model. [4]

In another term, fuzzy clustering can be used as a tool to obtain a partitioning of data where the transitions between the subsets are gradual rather than abrupt.

The objective of cluster analysis is the classification of the objects according to similarities among them, and organizing of data into groups.

This method can be applied for the approximation of non-linear systems, and facilitate the task of building and analyzing models of complex systems based on numerical data.

Chapter 3
METHODOLOGY

Modeling and simulation of process control systems using a black-box system identification technique is achieved by gathering real time data from a pressure control loop. That is chosen to be the ground of this exercise on university Technology Petronas pilot plant under the tag number (SIM 305 pilot plant 2: air flow, pressure & temperature pilot plant.).Refer to appendix A for the plant layout.

The overall project methodology is achieved through black-box identification techniques (building model from the process data): two different approaches are implemented .the performance of both models are compared to the performance of the empirical model developed using the same experiment data.

3.1 Neural Network based System Identification

The following procedures [see figure 4 below]were followed using real time data collected from the plant to build neural network model that emulate the behavior of the plant. But before going into the steps of building or constructing the model , the collected data are plotted and studied to give some understanding of the system behavior where it facilitate the steps of building the model. Such process insight might include:

1. Order of the system.
2. Dynamics of system whether slow or fast.
3. Operating range.
4. Time delay.
5. Nonlinearities (hard/smooth).

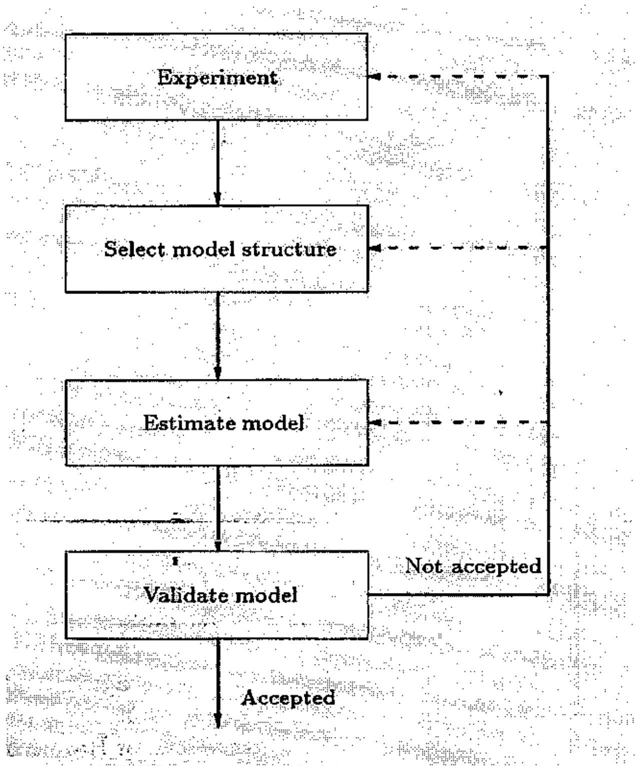


Figure 4 Overview of the basic system identification procedures using neural network.

3.1.1 Plant Experiment

The experiment is performed on the pilot plant (305 pilot plant 2: air flow, pressure & temperature pilot plant.) - Refer to appendix B for the control loop under this exercise. An experiment is performed to know the dynamics of the plant operation explore and choose a suitable input perturbation that is strong enough to excite the system.

Open loop experiment, first a control loop is identified to carry the experiment on, then the plant or the loop is operated in manual mode (controller is switched to manual-mode (PIC221)). This enable the operator to change the manipulated variable [MV] and observe the change on the process variable [PV], this facilitate how to choose a suitable perturbation to excite the plant. This is done by varying the input of the plant signal [CV].

After exploring the behavior of the plant a step input is chosen as 15% valve open, although it is much better to collect data across the entire operating range (i.e., keep in creasing the step change to cover the entire range of the operation), but the philosophy used here to choose just a perturbation MV=15% (valve opening 15 % is observed to produce dynamic response higher than five times the signal-to-noise-ratio, which is estimated to be (0.6 X5) equivalent to 3 psi)for observing the behavior of the plant with giving the plant enough time until it reaches the steady state. Here some advantage is considered from designing experiment for empirical modeling. The data set of corresponding inputs and outputs takes the form of

$$Z^N = \{[u(t), y(t)], T = 1, \mathrm{K}, N\}$$

This data is used for inferring a model of the plant.

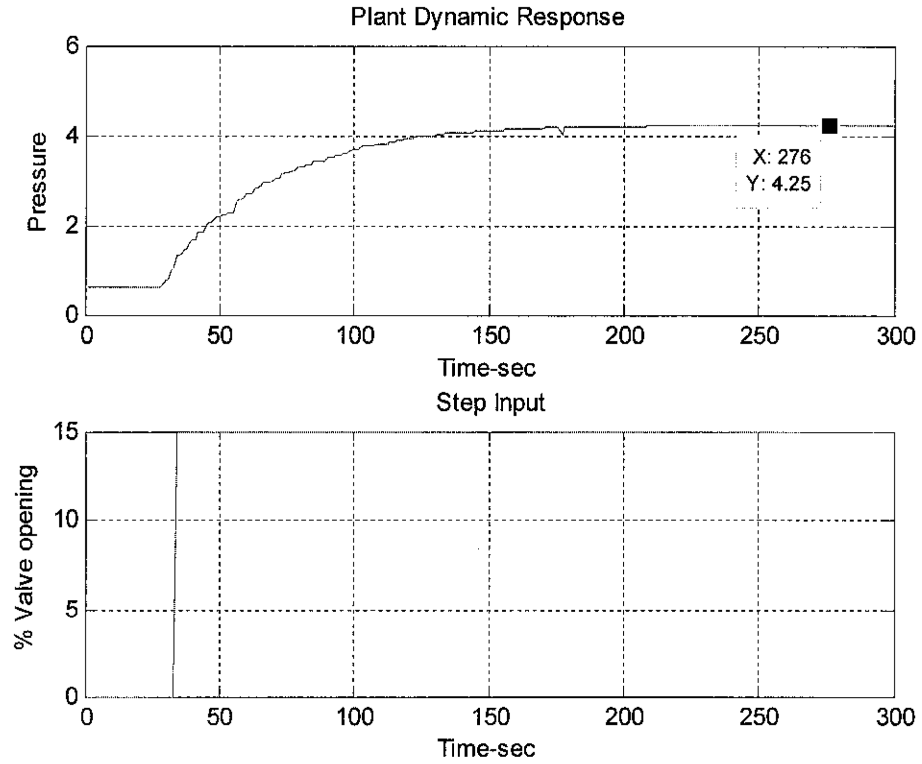


Figure 5 Identification Data

3.1.2 Model Structure Selection

The goal is to select a model structure. That is considered appropriate for describing the system (plant).A family of model structures is explored, including multilayer neural network, radial bias function network. Generally model structure plays an important role as an instrument or tool of mapping or emulating the plant dynamics. This might be decomposed in how to form the data vectors or regressors from past inputs and outputs and how to choose the nonlinear mapping from the regressor space to the output space. As practice in this project, multilayer Feedforward neural network is chosen as model structure, where parametric tuning of the model is achieved through the choosing of the hidden neuron and the number of layers.

3.1.3 Model Estimation

Model estimation is the process of picking a model from model structure family. This in neural network community is called training or learning. The model is trained using input-output data. Learning algorithm used is back-propagation or error back propagation.

3.1.4 Validation

After the model is being trained /estimated, it goes through evaluation stage to investigate whether it meets the desired requirements or not. As a measure of model quality model errors are evaluated to check the ability of the model in mapping or emulating the plant dynamic.

3.2 Fuzzy Identification

This section presents the individual steps of the identification techniques in order to obtain fuzzy model from data. Data set of 349 samples taken or recorded every second. Refer to figure 6 for the overview of fuzzy modeling procedures.

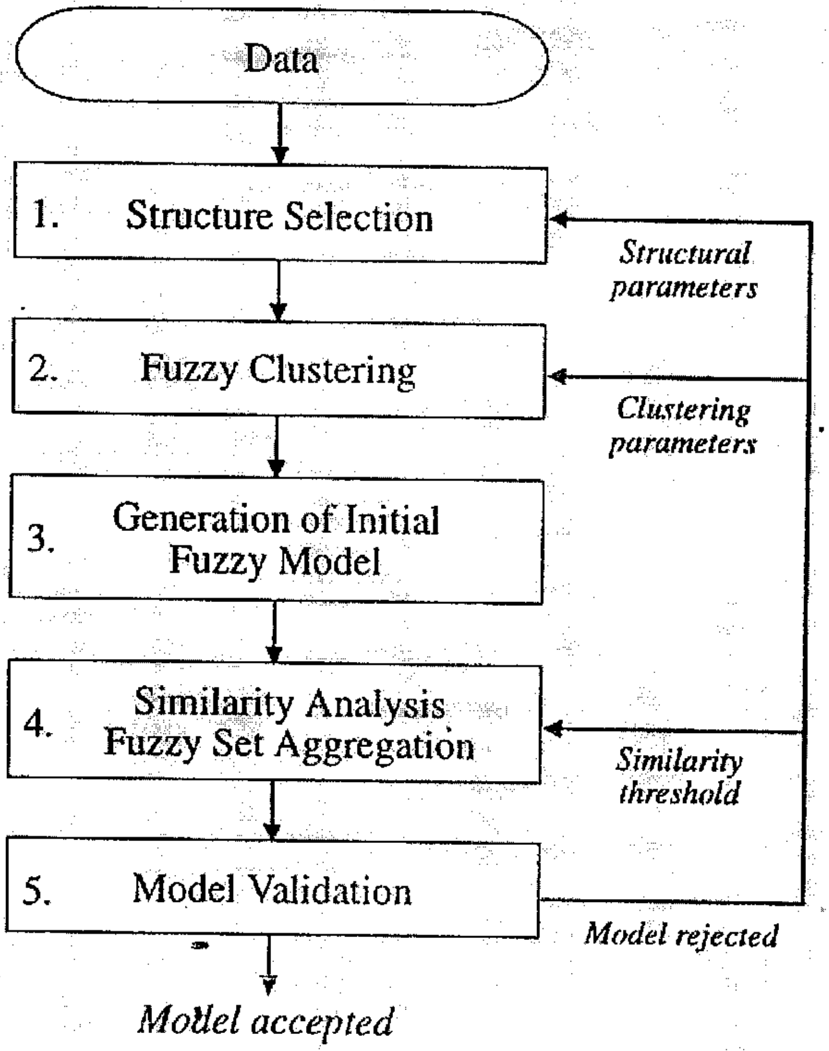


Figure 6 Overview of the identification approach based on fuzzy clustering

3.2.1 Design of Identification Experiment and Data Collection

The same input signal used for neural network identification is applied here, although for system identification it is preferable to choose an input signal that excite the system in the entire range of the considered variable [MV,PV] or the input out put range. Assumption is made by assuming the plant operation range MV 0 – 15 % valve open.

3.2.2 Structure Selection

Model structure selection helps in translating the identification of the dynamic system into a regression problem that can be solved by static manners. The order of the system is chosen as first order due to the prior knowledge available from the identification data, but in cases where the data can not provide good visualization of the system dynamics. The user can try different order of models and compare the models, then choose a model with best performance(less error).

The choice of the input and out put variables, input-output selection was not a problem in the identification of the loop in concern, because the other variables in the loop are kept constant. Referring to appendix .A , the plant input is chosen to be control valve(manipulated variable [MV]) , process variable [PV], is chosen to be the control, other variables are kept constant, this include hand valves (HV-200, HV-211, HV221)plus the control valve PY-243 is set manually to 40% pen (controller in manual mode (PIC-243) and the motor operated valve MV244 is maintained open through the experiment. the same set up implemented here is also applied loop identification for neural network model.

3.2.3 Clustering of the data

Structure selection led to a nonlinear static regression problem, which is approximated by a collection of linear models. By using MATLAB fuzzy model identification toolbox the identification data were clustered into submodels represented by linear approximation. The following function is used to cluster the data in to 6 clusters. Refer to appendix E for the out put argument of this function. The data matrix was constructed from the identification data

$$Z = \begin{bmatrix} opressure(1) & valveopen(1) \\ opressure(2) & valveopen(2) \\ \mathrm{M} & \mathrm{M} \\ opresssure(N-1) & valveopen(N-1) \end{bmatrix}$$

```
[F,C,P,V,D,J,M] = gkfast([opressure vopen],6,2,1e-3,1,[ones(1,500)])
```

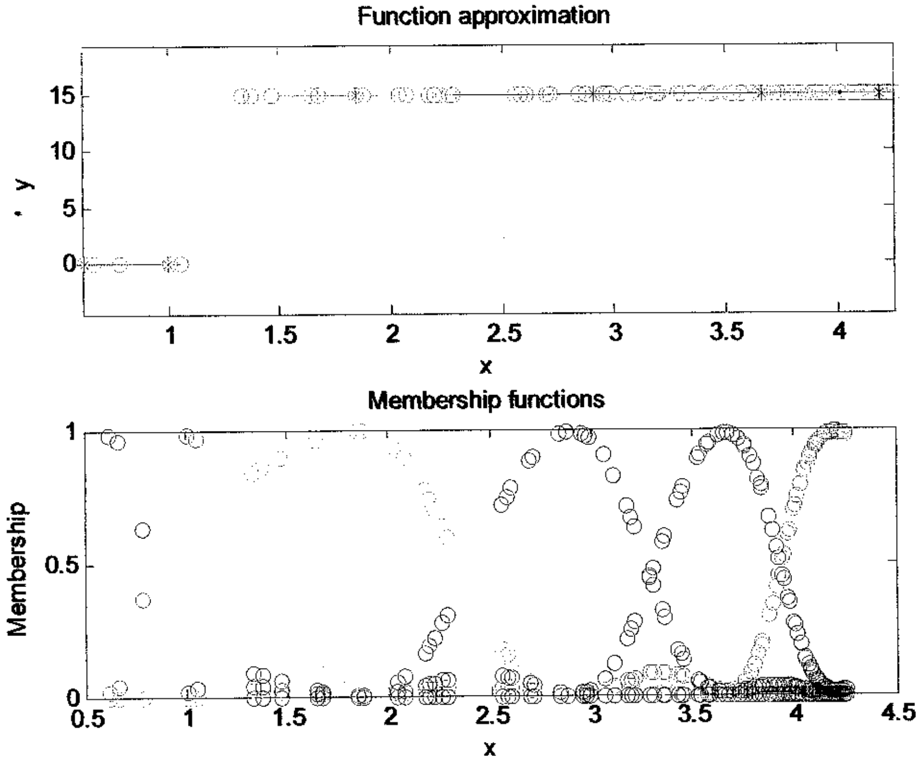


Figure 7 The upper graph shows the data and the local linear models given by the eigenvectors of the cluster covariance matrices. Cluster center are denoted by the '+'marks. the bottom graph depicts membership functions for x obtained as an approximation envelope of the projection of fuzzy partition matrix onto x

To determine the number of clusters, the validity measures were applied. Where the upper limit number of clusters was set to c=6, assuming that the nonlinearity of the process can be sufficiently approximated by 6 local linear models.

3.2.4 Generation of an Initial Model

After investigating the cluster in the identification data, a rough estimate about the suitable data cluster is made. The identification data is implemented in MATLAB, fuzzy model identification tool box to create model suitable for emulating the plant behavior.

3.2.5 Model validation

The fuzzy model produced (structure) is simulated to evaluate it is ability to predict the plant output. Model performance is highlighted in the result section. Where the visualization of the fuzzy model predicted output is plotted on the same graph with the plant observed output.

3.3 Empirical model

The plant data collected in the experiment section is used in order to obtain the empirical model. The main purpose for obtaining empirical model in this project is to compare the performance of black-box model.

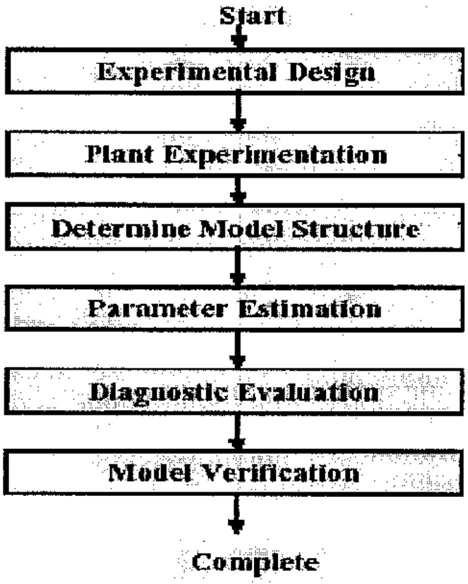


Figure 8 :Overview of the empirical model procedures

By referring to figure 8, this shows the overview of empirical model procedures. The first two steps are almost the same as in the section of plant experiment. However, model structure is assumed to be first order with dead time. This due to the plant's dynamic response.

3.3.1 Parameter Estimation

The process reaction curve or (plant dynamic response) obtained in the experiment section is used to approximate the process parameters. That includes:

1. Dead time (time delay), θ
2. Process gain,k_p
3. Time constant, τ

The graph in figure 9 demonstrates the process parameters calculation using the plant dynamic response. An input perturbation is chosen as 15 % valve open. That value chosen due to constrain of signal to noise ratio. Which is approximated as 5 times the plant optimum operation (i.e., for zero input or valve fully close the plant observed output is 0.6 psi. this value times 5 is equivalent to 3.00 psi. Hence any input can be chosen as long as it produces a response higher than (5X0.6=3 psi). The measures and calculated parameters are as below:

$\delta = 15\%\ valve\ opening(manipulated\ \mathrm{var}iable(MV))$

$\Delta = (\ 4.37 - 0.6146) = 3.7554 psi\ output\ \mathrm{var}iable(PV)$

$$k_p = \frac{\Delta}{\delta} = \frac{(3.7554 psi)}{(15\% open)} = 0.25$$

$\tau = 1.5(t_{63\%} - t_{28\%}) = 1.5(0.7 - 0.2) = 0.75\min = 45\sec$

$\theta = t_{63\%} - \tau = 0.7 - 0.75 = 0.05\min = 3\sec$

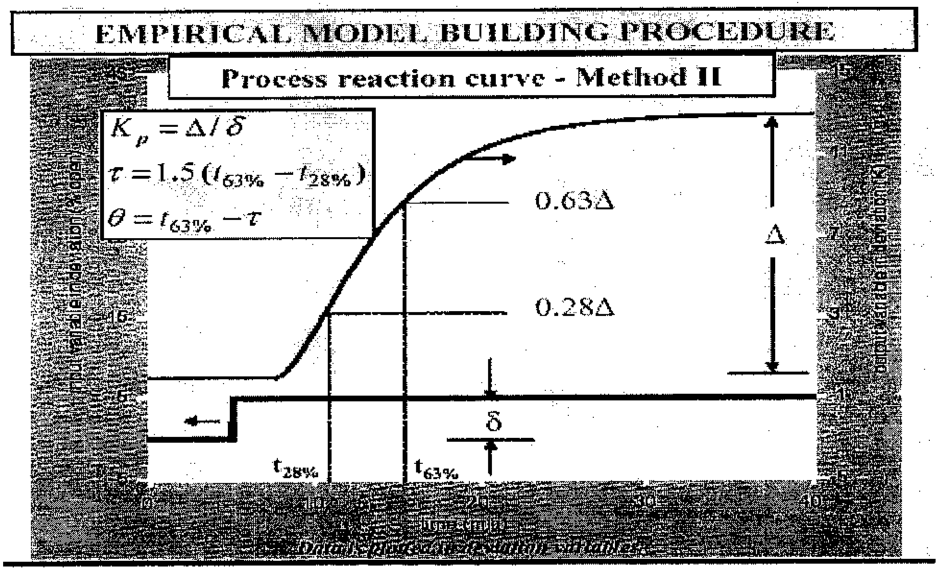


Figure 9 Processes reaction curve (PRC)

The graphical calculations determine the parameters for a first-order-with –dead time model. The process reaction curve is restricted to this model. The form of the model is as follows, with X(s) denoting the input and Y(s) denoting the output

$$\frac{X(s)}{Y(s)}=\frac{K_p e^{-\theta s}}{\tau s+1}$$

3.3.2 Empirical Model Simulation

The model parameters obtained above are used to develop a transfer function model. This transfer function is simulated in SIMULINK, refer to figure 10 for the simulink layout.

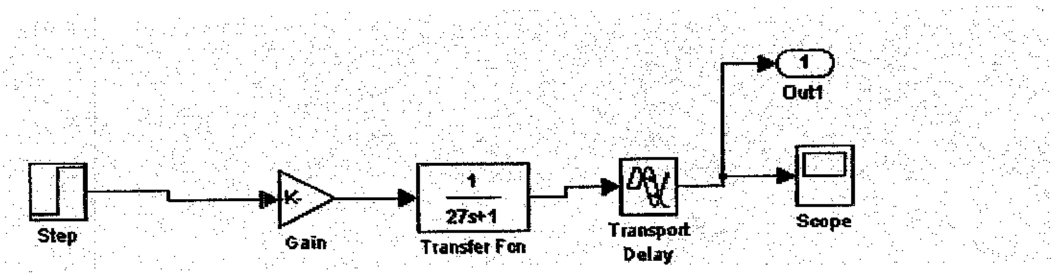


Figure 10 SIMULINK layout of the empirical model

The model is simulated with a perturbation same as the plant input 15% valve open, then the model is validated by comparing the plant output and the model output.

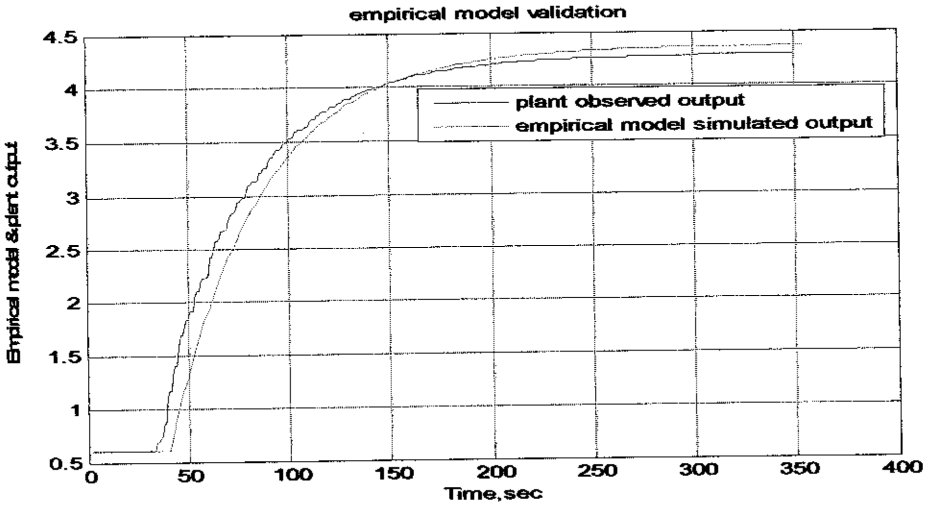


Figure 11 Empirical model output for an input of 15 % valve opening

CHAPTER 4
RESULT

The result obtained from neural network and fuzzy model is represented in term of models. These models are simulated to verify how close the models can emulate the plant dynamic behavior. Model quality is judged by evaluating and calculating the model residuals.

According to Lennart Ljung:

> Residuals are the "leftovers" from the modeling process-the part of data that the model could not produce. [5]

The performances of both models, developed in the previous stage are analyzed and compared to the empirical model performance in the following manner.

4.1 Neural network model analysis and evaluation

The neural network model produces plant predicted output in figure 12 below. It is observed that the neural network model tries to learn the dynamic of the system which include the dead time or the time delay. The model has the ability to learn or estimate the plant dead time.

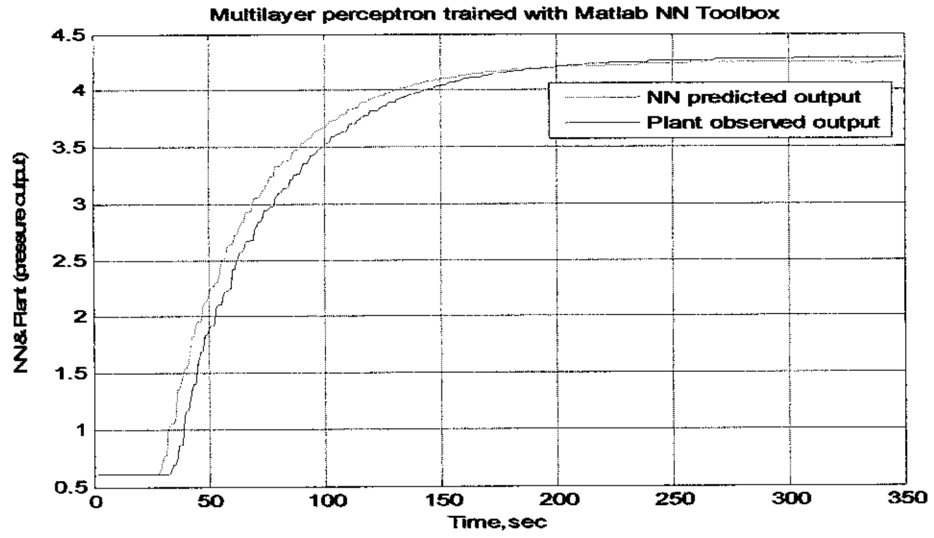


Figure 12 :Neural Network Model Predicted Output

In term of linearity in the transient response, nonlinearity problem appears. The model output has more deviation from the plant observed output. This is mainly due to the way how the experiment is designed to collect the data from the plant. In the steady state period the model produce a very good predicted output. The model succeeds in mapping the plant dynamic behavior as a linear relationship relating the plant input to the output. Plotting the model output and the plant output on the same graph gives a good visualization measure for evaluating how the model can fit the plant data or how good the model can emulate the plant output.

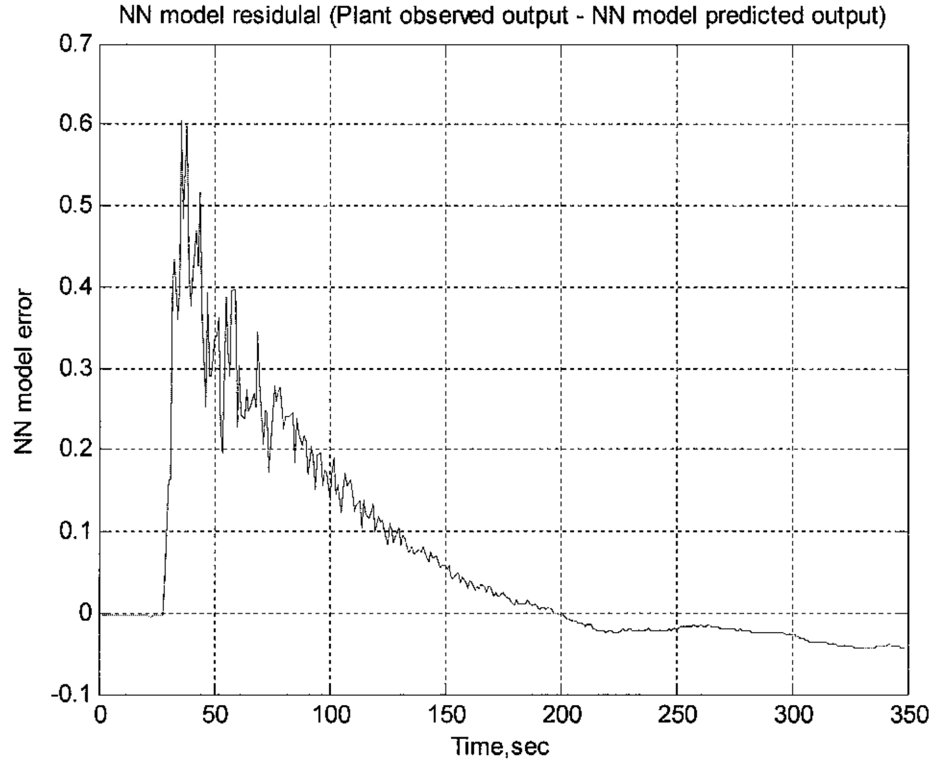


Figure 13 : NN Model Residuals

Model residuals or model errors are plotted as in figure 13 above, to give a measure of the model quality, investigating the model ability to reduce or minimize the error, the dynamic of the error (residuals [the observed output- the predicted output]). This dynamic shows that the model error is declining. In another word error at the transient period is declining to achieve a very minimum or almost zero model error at the steady state period.

4.2 Fuzzy Model Analysis and Evaluation

As a common practice in process control identification viewing the visual plant observed output and model predicted output. This gives some insight about the ability of the model to fit the plant data. Hence it provides the process control engineers with a criteria or a tool for judging the quality of different modeling techniques. Generally speaking it represents how the model can be used for further control application. Another aspect of modeling and system identification is that it gives diagnosis information about the system performance. This means instead of using the model for control application, it can be used for diagnostic purposes.

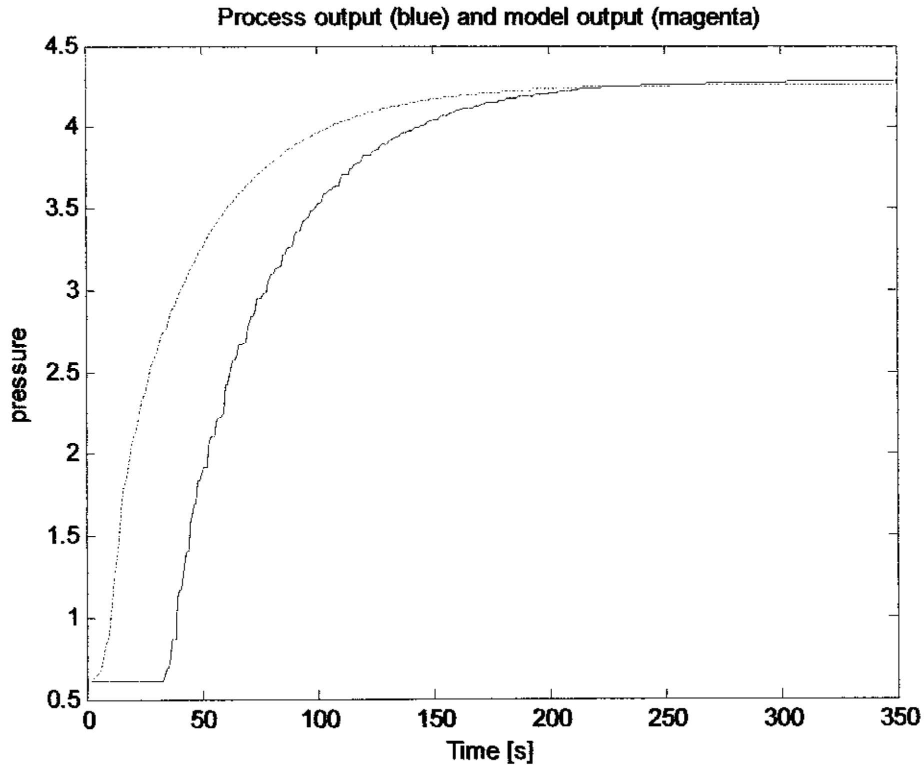


Figure 14 :Fuzzy Model Predicted Output

the plot of predicted model output and the observed output give information on how the model can emulate the plant dynamics.

4.2.1 Fuzzy Model Validation

In process control system identification, validation is achieved by comparing the predicted model output and the plant observed output (computing the residuals).More over visual comparison is also obtained to evaluate the ability of the model fit the plant data.

The author faces challenges in using the simulink model in fuzzy logic toolbox. The model simulated only in form of Matlab m-file attached in appendix D. The simulation result is plotted in figure 14 above.

As model quality measure the model residuals are plotted in figure 15 below. The dynamic of the model error is declining to zero, but in the transient response it shows a very high error. That is means the model is not able to produce the transient response of the plant, especially the dead time. The plant dynamics characterized by a dead time but the model doesn't have the ability to emulate this response.

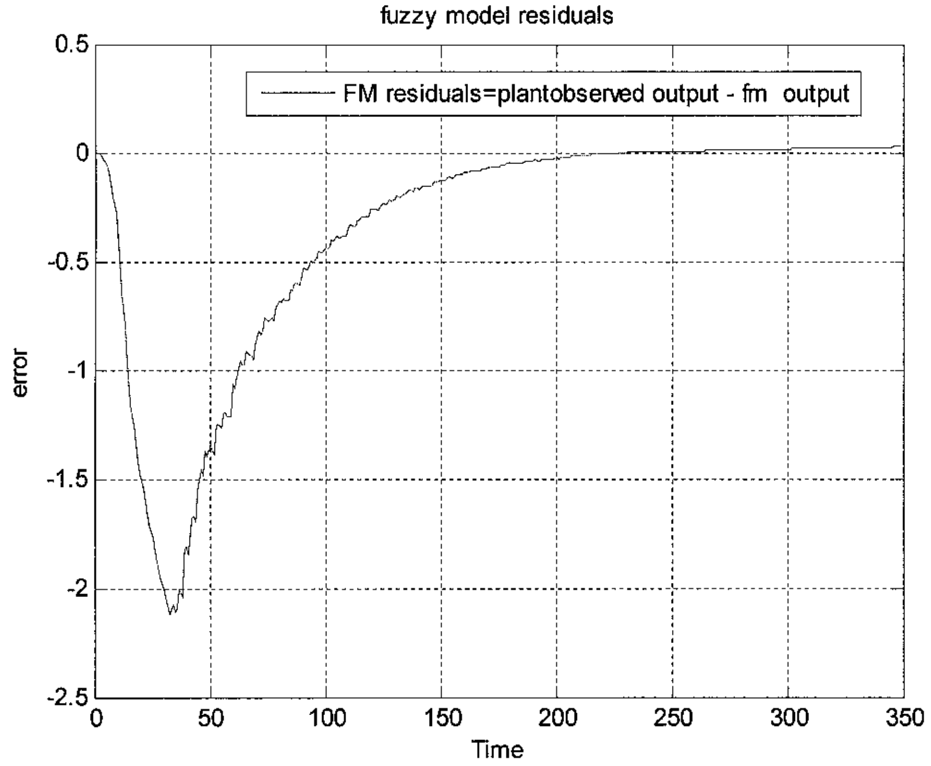


Figure 15 : fuzzy model residuals

4.3 Empirical Model Validation

The dynamic response of the empirical model plotted in figure 16 below. That shows the great ability of the empirical model to emulate the dynamics of the plant in relating the input to the output.

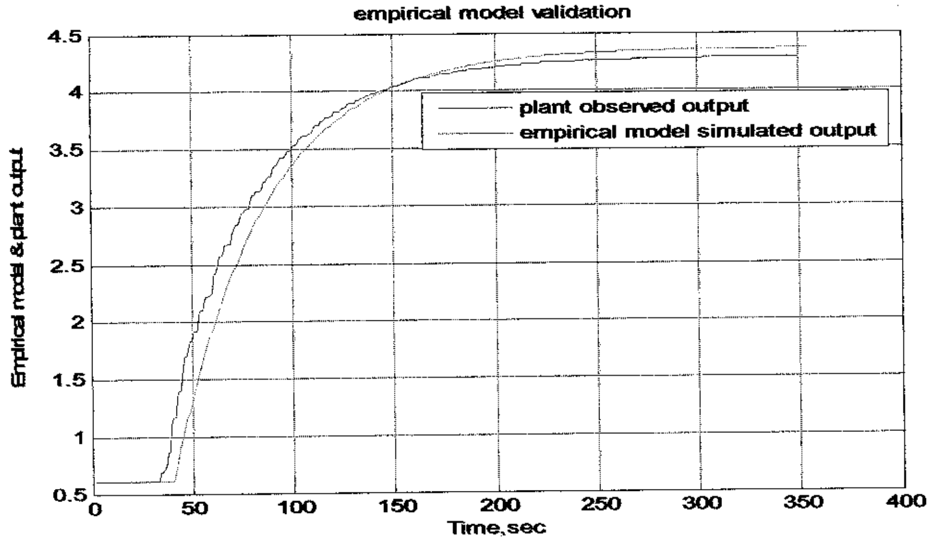


Figure 16 :Empirical model validation

Another validation is to plot the residuals. The residual plot shows the dynamic of the error or the ability of the mode to emulate the plant output.

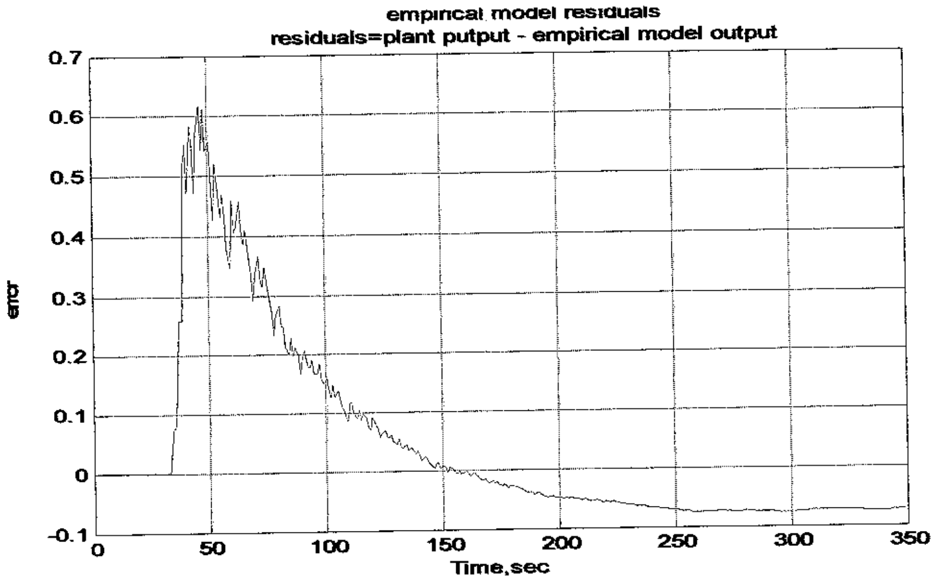


Figure 17 Empirical model residuals

4.4 Comparison between the black-box modeling techniques and empirical model performance

Modeling performance is evaluated by comparing different performance criteria. That is includes:

1) Mean squared error (MSE)
2) Mean absolute error (MAE)
3) Sum of the squared error (SSE)
4) Integral absolute error(IAE)

The tabulated data below show the model performance in term of comparing the error criteria. In general, the performance of the neural network model is the best in a achieving minimum error. It is followed by empirical model performance. However, the performance of the fuzzy model far behind the neural network model and empirical model. It is seen as a promising identification method but the main drawback of this method is the identification data set used in this project for fuzzy model.

In short, this performance measure showed that the black-box modeling techniques (neural network) performance is better than the empirical model performance.

Model	Performance			
	MSE	MAE	SSE	IAE
Neural Network	0.0212	0.0882	7.3926	30.7796
Fuzzy	0.4331	0.3567	151.1523	124.4957
Empirical	0.0287	0.1071	10.0265	37.3872

Table 2 Comparison of the performance of neural network, fuzzy and empirical models.

CHAPTER 5
DISCUSSION

System identification techniques applied in this project requires the process control system engineers to appreciate the difference between empirical modeling techniques where the collected data represent informative data. That contains the process dynamic in a clear picture. Using the same data in order to obtain a black-box model might not work especially for data with monotonic signal or only exited by single input. Although system identification or building process black-box models from data represents a very powerful method, where there is no any priori knowledge about the dynamic of the system. The ability of black-box modeling techniques to be used in creating models that can emulate the plant dynamic or create a proper mapping of the plant input output combination.

System identification is still faced by some limitation: first limitation on data quality, it is obvious that limitation of the use of system identification techniques is linked to the availability of good data and good model structures, without a reasonable data record not much can be achieved.

> According to Lennart Ljung, Bad signal-to-noise can, in theory, be compensated for, by longer data records. [5]

In the case of this project on pilot plant where there no restriction on manipulating the input signal (% of control valve opening). These features of pilot plant facilitate the learning issues for the process control engineer where strategies can be developed based on the plant operation.

The author recommends that the experiment duration can be longer, taking in consideration the signal conditioning. In the case of empirical modeling practice, the process control system engineer can choose the input signal that produce an output 5 times higher than the signal to noise ration to grantee an informative identification data .

Obtaining a fuzzy model from data requires performing fuzzy clustering that divide the identification data to submodels. Every submodel is represented by one rule in the fuzzy combination, but unfortunately here because the identification data contain on step input, or the input signal does not excite the whole range of the operation. There is only one submodel found in the identification data. The idea behind fuzzy clustering method is to cluster or group all the patterns in the identification data that have an excitation signal of different levels and different frequency. For example if the identification data contain excitation as valve open 0-15% ,15-30% ,30-43% ,40-55% and 60- 70% and each input has an interval length of 5 minutes, 10 minutes, 15 minutes ,5 minutes and 20 minutes respectively, apply fuzzy clustering for such data can create four or five sub models similar to the one in figure 7.

More constructive work can be done in the future. By using the same project flow or procedure but taking advantage of neural network toolbox and fuzzy model identification toolbox. By doing some effort on the design of the experiment to collect data this project can be more useful for the future work.

The both model performance (neural network and fuzzy model) were promising. However fuzzy model performance contains higher value of error. But still there is way where the model performance can be improved.

Empirical model can be a good guide to evaluate the feasibility of this project. It is easy to implement and easy to evaluate if the empirical modeling consideration is fulfilled.

CHAPTER 6
CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Two method were investigated to create a black box model ,as a system identification concern there are two areas need to be concentrated on to build a good model from data .

First the model structure which represent the dynamic of the system to be modeled, in both techniques (i.e., fuzzy identification and neural network identification). By selecting the model structure where the inputs out put variables are chosen. In this project exercise the selection of the model input was not a problem. But in running plant engineers might face problems that set as boundaries in conducting an informative experiment. Such constrains might include plant safety, product quality, etc.

Close loop experiment some time is suitable for such application but is still faced with restriction on varying the setpoint due to the reasons mentioned above like the product quality and safety consideration.

The project highlights the application of the black-box modeling (neural network and fuzzy identification). Upon the experimental practice in this project it is found that these techniques give more accurate performance in term of evaluation the model errors. This is a good feature of the black box modeling , because the main concern of process control system practice is to develop a model that fulfill certain performance criteria.

The methods used here for model performance (evaluating different characteristic of the error (MAE, MSE, SSE, and IAE). this gives a good insight of the model dynamic. Another concern is to reduce these errors.

6.2 Recommendations

The project can be revised and implemented in a more proper way by putting much effort on how to construct an informative identification data .setting the experiment to construct and record the identification data for the modeling purpose.

Differentiate between designing an experiment for empirical modeling and black-box modeling, because misunderstanding of the experiment designing objective. For these different modeling techniques might lead to failure of black-box system identification techniques.

In designing an experiment for process control system identification (black-box) or creating model from data, the following issues must be addressed carefully:

6.2.1 Operation Range

This is simple to be chandelled, but very important criteria to collect an informative identification data.

To achieve this requirement an input signal that is used to excite the system for the sake of collecting identification data must be design to cover the whole operation range (the idea is to vary the input signal and observe the impact on the output(s))

Finally, the purpose of the experiment must be stated clearly to collect a set of data that describe how the system behaves over its entire range of operation.

6.2.2 Developing SIMULINK block

The black-box models developed in this project can be implemented in most productive way if simulink blocks are available. Because these model are represented in structure form (i.e., not a transfer function form like the empirical model). That is different from the empirical model which can be represented in transfer function models.

Simulink blocks can help in implementing many control strategies in SIMULINK. Hence it helps the process control system engineers to investigate more about the process in their concern. Another advantage of developing process model in simulink blocks it helps to study more about the process. This is considered as learning facilities for activities that it cannot be performed on the running plant due to operation , quality , and safety constrains that cannot entertain any mistakes or changes in the plant dynamic operation aspects such like introducing a setpoint change.

REFERENCES

[1] http://icewww.et.tudelft.nl/~babuska

[2] Chin-Teng Lin and C.S. George Lee.1995, Neural Fuzzy Systems, Prentice Hall PTR

[3] M.Norgaard.,O.Ravn,N.K.Poulsen and L.K.Hansen,2000,Neural Networks for Modelling and Control of Dynamic Systems, Springer.

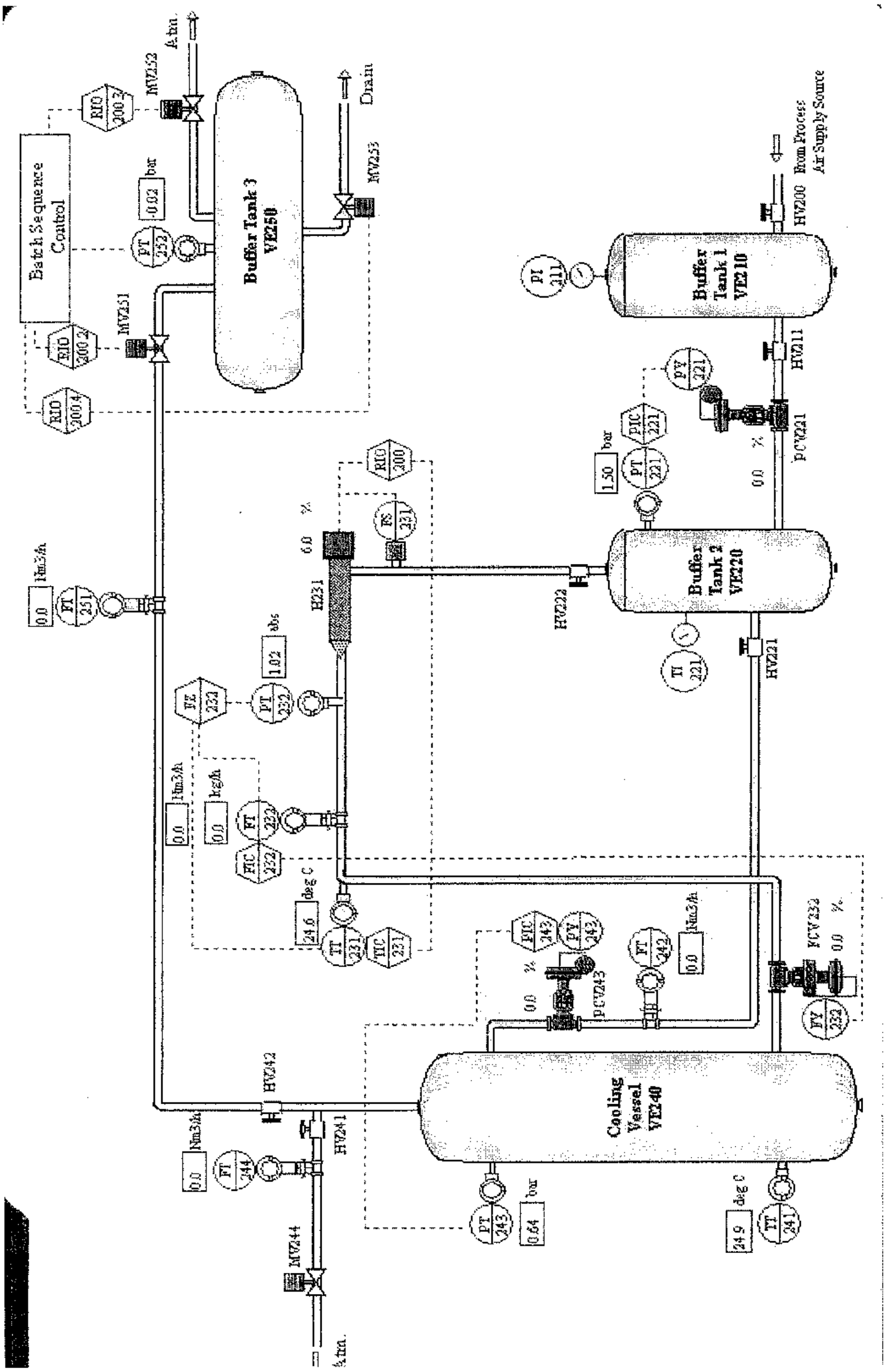
[4] Robert Babuska .1998, Fuzzy Modeling for Control, Kluwer Academic Publisher

[5] Lennart Ljung, 1999, System Identification, Theory for the user, Prentice Hall PTR

APPENDICES

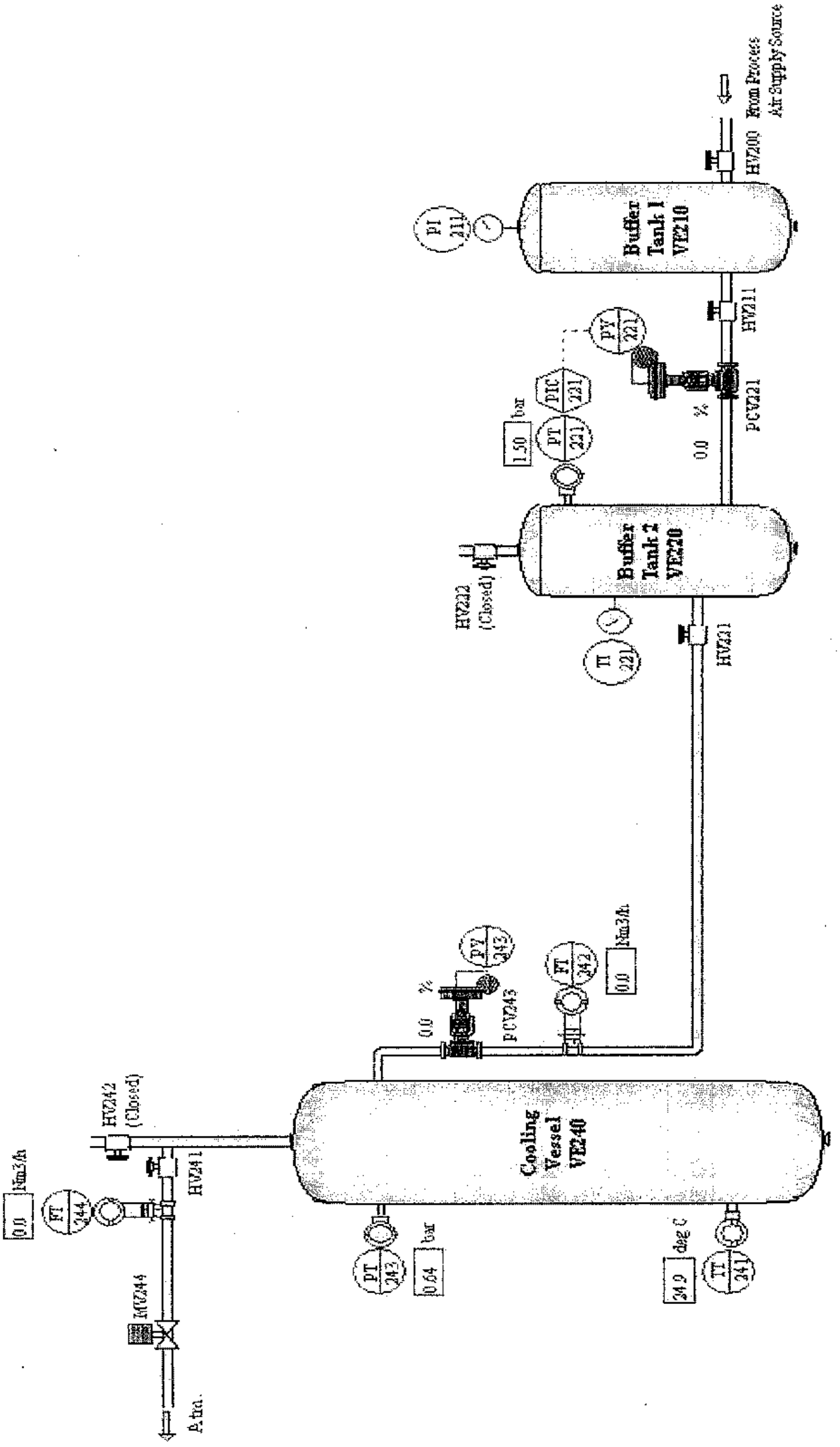
APPENDIX A

OVERVIEW OF SIM 305 PILOT PLANT 2: AIR FLOW, PRESSURE & TEMPERATURE PILOT PLAN



APPENDIX B

CONTROL LOOP



APPENDIX C

MATLAB M-FILE TO GENERATE NEURAL NETWORK PROCESS MODEL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%M-file to create NEURAL NETWORK model for simple air pressure process
%feed forward neural network(MLP),implememting back-prpoagation learning
%algorthims as supervised learning rule.
%%%%%%%%%%%%%%%%%%%%%%%%%
load DATA3;
%  premnmx  - Normalize data for maximum of 1 and minimum of -1.
[PN,minp,maxp,TN,mint,maxt]=premnmx(TrainIP,TrainTgt);
%------- newff - Create a feed-forward backpropagation----------
%-----------------------network.---------------------------

% ----------Transfer functions.------------
%(purelin   - Linear transfer function.)%%%%%%%%%%%%%%
%(tansig    - Hyperbolic tangent sigmoid transfer function)
%(purelin   - Linear transfer function.)
 net=newff(minmax(PN), [1
349],{'tansig','purelin','tansig','tansig','tansig'},'trainlm','learngdm','mse'
);
 net.trainParam.epochs=10;
 net.trainParam.show=1;
 net.trainParam.lr=.001;
 net.trainParam.goal=0;
 %%%%%%%%%%%%%%%%%%%%%%%%%
 % Using networks.
 %%%%%%%%%%%%%%%%%%%%%%%%%%
 %  (1) train    - Train a neural network.
 [net,tr,Y,E]=train(net,PN,TN);
 % tramnmx  - Transform data with precalculated minimum and maximum.
 PN_Test = tramnmx(TestIP,minp,maxp);
 %  (2) sim      - Simulate a neural network.
TN_Test = sim(net,PN_Test); %TN_Test is the neural networl predicted output
 %  postmnmx - Unnormalize data which has been normalized by PREMNMX.
 [quireyInputs predictOutputs]=postmnmx(PN_Test,minp,maxp,TN_Test,mint,maxt);
  plot(TestTgt,'-*');hold on
plot(predictOutputs,'r:');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% neural network performance
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------Performance functions.-------------------
%   MAE:      - Mean absolute error performance function.
  MAEA=mae(E)% E (error)is the difference between  neural network
       %predicted output and the plant output (target output)
%   mse       - Mean squared error performance function.
%MSE: mean squred error
MSEA=mse(E)
%   SSE:      - Sum squared error performance function.
 SSEA=sse( E)
```

```
TRAINLM, Epoch 0/10, MSE 1.03995/0, Gradient 305.853/1e-010
TRAINLM, Epoch 1/10, MSE 1.27785e-006/0, Gradient 1.8587/1e-010
TRAINLM, Epoch 2/10, MSE 3.67202e-015/0, Gradient 0.000219261/1e-010
TRAINLM, Epoch 3/10, MSE 2.32261e-026/0, Gradient 4.71818e-012/1e-010
TRAINLM, Minimum gradient reached, performance goal was not met.


MAE =

  1.2786e-013


MSE =

  2.3226e-026


SSE =

  8.1059e-024
```

Note: Errors calculated here with reference to the normalized error signal

But in the report errors are calculated due to the original data

APPENDIX D

MATLAB M-FILE FOR IMPLEMENTING FUZZY MODEL USING FUZZY MODEL IDENTIFICATION TOOLBOX

```
% generate a fuzzy model for air pressure process

clear FM

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define constants
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
FM.c = 4;           % number of clusters
FM.m = 1.2;              % fuzziness parameter
FM.seed = 0;           % seed
FM.ante = 2;            % antecedent:   1 - product-space MFS
                            %               2 - projected MFS
FM.cons = 2;           % consequent estimation

FM.Ny = 1;         % denominator order
FM.Nu = 1;         % numerator orders
FM.Nd = 1;         % transport delays
                   % (set to 1 for y(k+1) = f(u(k),....))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% identification data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load MVSET;
Dat.U = vopen(:);
Dat.Y = opressure(:);
Dat.Ts = 1;           % sample time [s]
Dat.InputName = 'valve opening';
Dat.OutputName = 'pressure';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% validation data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load evaldat;
ue = vopen(:);
ye = opressure(:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% make fuzzy model by means of fuzzy clustering
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('hit any key to create  fuzzy model by means of fuzzy clustering ');pause
[FM,Part] = fmclust(Dat,FM);

[gain,T] = fm2kt(FM);
 a = antename(FM);
 k=1;
 ni = length(a{k});
 disp('hit any key to update the mpdel');pause
 FM = FMupdate(FM);
 %FIS = fm2fis(FM,ye)
% txt = fuzz2txt(rls,ue,ye,300,test_vo_2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% simulate the fuzzy model for validation data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1); clf
[ym,VAF,dof,yl,ylm] = fmsim(ue,ye,FM); VAF
title('Process output (blue) and model output (magenta)');
ylabel('pressure');
disp('Hit any key to see the local models ...'); pause

figure(2); clf
subplot(211); plot([ylm{1}]);
title('Individual local models');
xlabel('Time'); ylabel('pressure');
subplot(212); plot(dof{1})
title('Degrees of fulfillment');
xlabel('Time'); ylabel('Membership grade');
disp('Hit any key to see the membership functions ...'); pause

figure(3); clf
plotmfs(FM);
```

APPENDIX E

```
function [f,v,P,V,D,J,M] = gkfast(x,f0,m,e,s,rho)
% Gustastafson-Kessel clustering algorithm (with fuzzy covariance matrix).
%
% [F,C,P,V,D,J,M] = GKFAST(X,F0,m,e,s,rho)
%   Input:
%    X ... M by N data matrix,
%         M is the number of data points and N data dimension
%    F0 ... either an initial fuzzy partition matrix, or the
%         number of clusters. In the latter case, a default
%         partition matrix is generated.
%    m ... optional parameter m > 1, determines the fuzziness
%         of clustering, for m close to 1 clusters become crisp,
%         default value is 2
%    e ... optional termination tolerance, the algorithm stops
%         when max(max(|F(k-1) - F(k)|)) <= e, default tolerance 1e-3
%    s ... optional parameter for plotting intermediate results
%         (only for 2D data), default 0 - i.e. no plot, set
%         to 1 to show the clustering process on-line,
%      set to 2 to speed-up the plots for ordered data
%    rho .. 1xM vector of expected cluster volumes (default a unit vector)
%
%   Output:
%    F ... fuzzy partition matrix
%    C ... cluster means matrix
%    P ... cluster covariance matrices concatenated in one matrix
%         P = [P1;P2;...Pk], where k is number of clusters
%    V ... eigenvectors of the covariance matrices, corresponding
%         to the smallest eigenvalues
%    D ... eigenvalues of the covariance matrices
%    J ... history of the clustering criterion J
%     M ... matrices inducing the distance norm, calculated as
%          M = det(P).^(1/N)*inv(P), M = [M1;M2;...Mk]

% (c) Robert Babuska, 1994-96

if (nargin < 3), m = 2;    elseif isempty(m),   m = 2; end;
if (nargin < 4), e = 1e-2; elseif isempty(e),   e = 1e-2; end;
if (nargin < 5), s = 0;    elseif isempty(s),   s = 0; end;

[mx,nx] = size(x);
[mf0,nf0] = size(f0);
x1 = ones(mx,1);
inx = 1/nx;

% Initialize fuzzy partition matrix

if max(mf0,nf0) == 1,      % only # of cluster given
  c = f0;
  mm = mean(x);
  aa = max(abs(x - ones(mx,1)*mm));
  v = 2*(ones(c,1)*aa).*(rand(c,nx)-0.5) + ones(c,1)*mm;
elseif nf0 == nx,      % centers given
  c = mf0;
  v = f0;
end;

if mf0 ~= mx,
% Calculate f0
  for j = 1 : c,                    % for all clusters
    xv = x - x1*v(j,:);
    d(:,j) = sum((xv.^2)')';
  end;
  d = (d+1e-100).^(-1/(m-1));
  f0 = (d ./ (sum(d')'*ones(1,c)));
```

```
else
 c = size(f0,2);
 fm = f0.^m; sumf = sum(fm);
 v = (fm'*x)./(sumf'*ones(1,nx));
end


f = zeros(mx,c);            % partition matrix
iter = 0;             % iteration counter


if (nargin < 6), rho = ones(1,c); elseif isempty(rho), rho = ones(1,c); end;


% initialize graphics
if (s ~= 0)  &  (nx < 3),
  subplot(211);
  lines = [v(:,1)*x(:,1)'+v(:,2)*ones(1,mx)]';
  mask = find(f0 < 0.2);            % find membership degrees < 0.2
  lines(mask) = NaN*ones(size(mask));     % mask with NaN's for plots
  H1 = plot(x(:,1),x(:,2),'go',v(:,1),v(:,2),'r*',x(:,1),lines,'EraseMode','xor');
  title('Function approximation');
  xlabel('x'); ylabel('y');
  minx = min(x(:,1)); maxx = max(x(:,1));
  miny = min(x(:,2)); maxy = max(x(:,2));
  ma = 0.3*max(abs(x(:,2)));
  axis([minx maxx miny-ma maxy+ma]);
  subplot(212);
  if s == 1,
   H2 = plot(x(:,1),f0,'o','EraseMode','xor');
  else
   H2 = plot(x(:,1),f0,'EraseMode','xor');
  end;
  title('Membership functions');
  xlabel('x'); ylabel('Membership');
  set(gcf,'UserData',[H1;H2]);
end;
```

```
% Iterate

while  max(max(abs(f0-f))) > e

  iter = iter + 1;

  f = f0;

  fm = f.^m; sumf = sum(fm);

% Calculate centers

  v = (fm'*x)./(sumf'*ones(1,nx));

  for j = 1 : c,                    % for all clusters

    xv = x - x1*v(j,:);

% Calculate covariance matrix

    p = ones(nx,1)*fm(:,j)'.*xv'*xv/sumf(j);

%    p = ones(nx,1)*fm(:,j)'.*xv'*xv;

    if rcond(p)<1e-15;

        [ev,ei]=eig(p);

        ei(find(ei<max(diag(ei))*1e-15))=max(diag(ei))*1e-15;

        ei=diag(diag(ei));

        p=ev*ei*inv(ev);

              end

% Calculate distances

    M = (det(p)/rho(j))^inx*inv(p);

    d(:,j) = sum((xv*M.*xv)')';

% Calculate eigen vectors and cluster prototypes (lines)

    if s ~= 0 & nx < 3,

      [ev,ed] = eig(p); ed = diag(ed);

      ev = ev(:,ed == min(ed));

      lines(:,j) = -x(:,1)*ev(1)/ev(2) + v(j,:)*ev/ev(2);

      mask = find(f0 < 0.2);              % find membership degrees < 0.2

      lines(mask) = NaN*ones(size(mask));     % mask with NaN's for plots

    end;

  end;

  J(iter) = sum(sum(f0.*d));

% Update f0

  d = (d+1e-10).^(-1/(m-1));
```

```
 f0 = (d ./ (sum(d')'*ones(1,c)));



% Plot intermediate results

 if (s ~= 0), fprintf('Iteration count = %d, J = %f\n\n',iter,max(max(f-f0))),

  if (nx < 3),

   H = get(gcf,'UserData');

   set(H(2),'xdata',v(:,1),'ydata',v(:,2));

   for i = 1 : c,

    set(H(2+i),'ydata',lines(:,i));

    set(H(2+c+i),'ydata',f0(:,i));

   end;

   drawnow;

  end;

 end;

end


fm = f0.^m; sumf = sum(fm);


P = zeros(nx,nx,c);          % covariance matrix

M = P;                       % norm-inducing matrix

V = zeros(c,nx);             % eigenvectors

D = V;                       % eigenvalues


% calculate P,V,D,M

for j = 1 : c,               % for all clusters

  xv = x - ones(mx,1)*v(j,:);

% Calculate covariance matrix

  p = ones(nx,1)*fm(:,j)'.*xv'*xv/sumf(j);

  if rcond(p)<1e-15;

    [ev,ei]=eig(p);

    ei(find(ei<max(diag(ei))*1e-15))=max(diag(ei))*1e-15;

    ei=diag(diag(ei));

    p=ev*ei*inv(ev);
```

```
            end

% Calculate eigen values and eigen vectors

   [ev,ed] = eig(p); ed = diag(ed)';

   ev = ev(:,ed == min(ed));

% Put cluster info in one matrix

   P(:,:,j) = p;

   M(:,:,j) = (det(p)/rho(j)).^(1/nx)*inv(p);

   V(j,:) = ev(:,end)';

   D(j,:) = ed;

end;
```