A THESIS
ON

DATA-BASED MODELING: APPLICATION IN PROCESS IDENTIFICATION, MONITORING AND FAULT DETECTION

SUBMITTED BY

NAGA CHAITANYA KAVURI

(608CH301)

FOR THE PARTIAL FULFILLMENT OF

M. TECH (RESEARCH) DEGREE

UNDER THE ESTEEMED GUIDANCE OF

DR. MADHUSREE KUNDU

DEPARTMENT OF CHEMICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

JANUARY 2011

ABSTRACT

Present thesis explores the application of different data based modeling techniques in identification, product quality monitoring and fault detection of a process. Biodegradation of an organic pollutant phenol has been considered for the identification and fault detection purpose. A wine data set has been used for demonstrating the application of data based models in product quality monitoring. A comprehensive discussion was done on theoretical and mathematical background of different data based models, multivariate statistical models and statistical models used in the present thesis.

The identification of phenol biodegradation was done by using Artificial Neural Networks (namely Multi Layer Percetprons) and Auto Regression models with eXogenious inputs (ARX) considering the draw backs and complications associated with the first principle model. Both the models have shown a good efficiency in identifying the dynamics of the phenol biodegradation process. ANN has proved its worth over ARX models when trained with sufficient data with an efficiency of almost 99.99%. A Partial Least Squares (PLS) based model has been developed which can predict the process outcome at any level of the process variables (within the range considered for the development of the model) at steady state. Three continuous process variables namely temperature, pH and RPM were monitored using statistical process monitoring. Both univariate and multivariate statistical process monitoring techniques were used for the fault detection purpose. X-bar charts along with Range charts were used for univariate SPM and Principal Component Analysis (PCA) has been used for multivariate SPM. The advantage of multivariate statistical process monitoring over univariate statistical process monitoring has been demonstrated.

Hierarchical and Non-hierarchical clustering techniques along with PCA were used to find out different classes (qualities) of wine samples in the wine dataset. Once the classes present in the wine dataset were identified, the statistical and ANN based classifiers designed were used for authentication of unknown wine samples. PLS based model has been used for developing the statistical classifier, which has shown an identification efficiency of 98.5%. Two types of neural networks namely Probabilistic Neural Network (PNN) and Adaptive Resonance Theory (ART1) networks were used for the development of ANN based classifiers. ART1 networks unanimously showed their superiority over the other classifiers with 100% efficiency even when trained with a minimum amount of data.

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the thesis entitled "**Data-based Modeling: Application in Process Identification, Monitoring and Fault Detection**" submitted by **Mr. Naga Chaitanya Kavuri**, in partial fulfillment of the requirements for the award of Master of Technology (Research) in Chemical Engineering, at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge, the matter presented in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Prof. Dr. Madhusree Kundu
Department of Chemical Engineering
National Institute Of Technology
Rourkela-769008

Date:

Place: NIT Rourkela

ACKNOWLEDGEMENT

It is impossible to thank one and all in this thesis. A few however stand for me as I go on to complete this project. If words are considerable as symbols of approval and taken as acknowledgement then let the words play a heralding role in expressing my gratitude.

I would like to express my extreme sense of gratitude to **Dr. Madhusree Kundu**, Associate Professor, NIT Rourkela for her guidance throughout the work and her encouragement, positive support and wishes extended to me during the course of investigation.

I would also like to thank **Dr. K. C. Biswal**, H.O.D, Dept of Chemical Engineering, NIT, Rourkela, for his support academically. A special thanks to Prof. **G. K. Roy,** Chemical Engineering Department, NIT, Rourkela, for their valuable advices and moral support.

I am highly indebted to the authorities of NIT, Rourkela for providing me various facilities like library, computers and Internet, which have been very useful.

On a personal front, I would like to thank **Mr. Jagajjanani Rao** from the bottom of my heart without whom I will not be joining this institution and made this far.

I cannot say thanks and walk away from two persons who were there as pillars to my moral stability through my bad times. It’s you *Mom* and *Deepu*.

I express special thanks to all my friends, for being there whenever I needed them. Thank you very much Sonali, Sonu, Gaurav Bhai, Diamond, Seshu, Vamsi and Vikky.

Finally, I am forever indebted to my brother for his understanding and encouragement when it was most required.

I dedicate this thesis to my family and friends.

Naga Chaitanya Kavuri

TABLE OF CONTENTS

Chapter 4- Process Monitoring & Fault detection

Chapter 5-Conclusion

CHAPTER 1- INTRODUCTION

1.1. INTRODUCTION

Present work addresses three different kinds of problems related to Process Identification, Product Quality Monitoring, and detection of abnormal operating condition in a process leading to Process Faults. Process identification helps in developing an efficient monitoring and controller systems for any process. Process identification is about the detection and understanding of dynamics present in a process from its historical data. Different machine learning algorithms can be effectively utilized for these purposes. The detection of fault followed by its diagnosis is extremely important for effective, economic, safe and successful operation of a process. Efforts to manufacture a higher proportion of within specification product and to reduce the variability in the product quality, i.e. to produce more consistent product, has lead to an increase in the use of Statistical Process Control (SPC). SPC refers to a collection of statistical techniques and charting methods that have been found to be useful in ensuring consistent production and, consequently, in obtaining significant advantages. However, most modern industrial processes have available frequent on-line measurements on many process variables and, in some instances, on several properties of raw materials and final product. Furthermore, there are measurements of characteristics related to product quality that are usually measured infrequently off-line. Therefore, industrial quality problems are multivariate, since they involve measurements on a number of characteristics, rather than one single characteristic. As a result, univariate SPC methods and techniques provide little information about the interactions between characteristics and, therefore, it is not appropriate for modern day processes. Most of the limitations of univariate SPC can be addressed through the application of *Multivariate Statistical Process Control* (MSPC), which considers all the characteristics of interest simultaneously and can extract information on the behavior of each characteristic relative to the others. Here there are two applications of MSPC. One is the detection and allotment of end product to one of the predefined categories which is called Statistical Quality Control. Second one is the online process monitoring to make sure the process is under control or not which is referred as fault detection. A biodegradation process of an organic pollutant phenol is used for the identification and fault detection purposes. Phenol, one of the major organic pollutants from paper and pulp, pharmaceutical, iron-steel, coke- petroleum, and paint industry [1-4] is degraded by heterotrophic bacteria *Pseudomonas putida* (ATCC: 11172).

For Statistical Quality Control purpose the determination of quality of wine has been considered. The determination of quality of food stuffs, water and beverages and control of their correspondence to standards is an urgent problem needed to be addressed. Statistical Quality Control (SQC) was designed to sample a large population on an infrequent basis. Classical analytical techniques such as various chromatography, spectrometry, and etc are used for the determination of different characteristics of wine samples. However they are time-consuming, expensive, and laborious which can hardly be done on-site or on-line. For quality control of perishable products, it is necessary to evaluate a group of certain components that reflects the ageing and spoilage of the product. These components can be numerous or unknown and the problem appear to be quite difficult. Besides, it is impractical and very hard to compare the results of instrumental analysis to biological sensing [5]. Chemometric techniques have been used in wine analysis by researchers like Buratti et al., 2007; Parra et al., 2006; Buratti et al., 2004; Riul et al., 2004; Di Natale et al., 2004; Legin et al., 2003; Di Natale et al., 2000[6-12].

Artificial Adoptive Resonance theory network (ART1), Probabilistic neural network (PNN) and PLS based classifiers can be immensely helpful for classification among wine samples.

There is an inherent relation between the objectivity of the proposed project and the modeling; especially the data based modeling. A brief discussion about modeling, hence data based modeling seems to be an integral portion of the prologue.

1.2. MODELING

The essence of process modeling, in general, is to capture the important aspects of the physical reality while discarding irrelevant detail of the process.It may therefore often be possible to devise several types of models of the same physical reality and one can pick and choose among these depending on the desired model accuracy and on their capability of analyzing the process. An efficient and effective process model is required for the following purposes:

- Research & Development
- Planning and Scheduling
- Process Design
- Process Optimization
- Process Simulation
- Process Identification
- Process Control, Monitoring & Safety Measure
- Fault Detection & Diagnosis

Different models with different degrees of sophistication can be built. The degree of complexity to be chosen is balance between accuracy and computational burden. Too sophisticated models are not always computationally affordable. An outline of the procedure for first principle model building can be summarized in the following steps.

- Decision on the level of model complexity
- Writing the model equations
- Judicious model assumptions
- Devising suitable mathematical structure and solution methodology
- Determination of model parameters
- Model verification
- Model validation & refinement
- Model prediction

It is important in this respect to recognize the fact that most mathematical models are not completely based on rigorous mathematical formulation of the physical and chemical processes taking place in the system. Every mathematical model contains a certain degree of empiricism. The degree of empiricism limits the generality of the model and, as our knowledge of the fundamentals of the process increases, the degree of empiricism decreases and the generality of the model increases. The existence of models at any stage, with their appropriate level of empiricism, help greatly in the advancement of the knowledge of the fundamentals, and, therefore, helps to decrease the degree of empiricism and increase the level of rigor in the mathematical models. Models always contain certain simplifying assumptions which are believed; not to affect the predictive nature of the model in any manner that undermines the purpose of it. There are processes; whose physics are poorly known. For example one cannot really determine the kinetics of a biological degradation process as it is highly difficult to recognize the rate determining step because of numerous enzymatic reactions involved in the metabolic pathway. For modeling such process a different approach is followed called 'Data Based Modeling' or 'Black Box Modeling' or 'empirical modeling'. Here the modeling will be done based only on empiricism. In this context the mathematical modeling can be disjointed in to two categories.

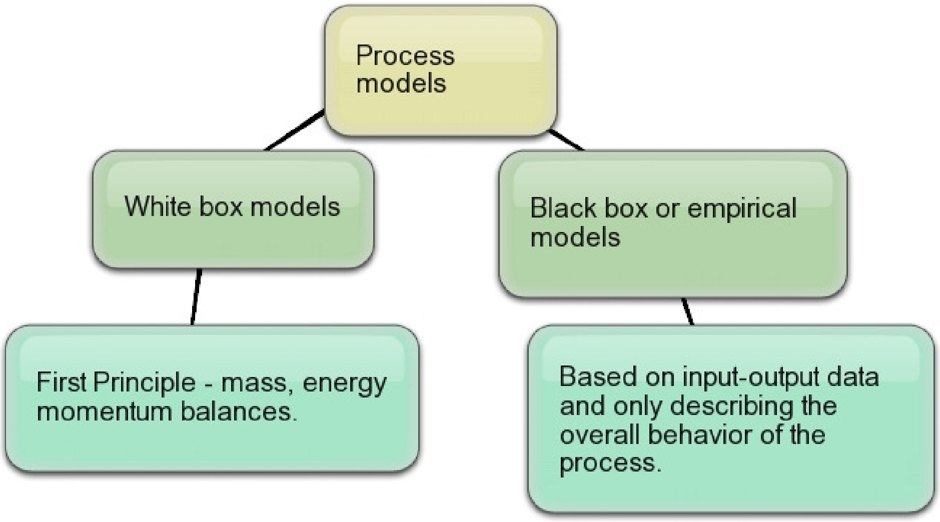
Process models
White box models
Black box or empirical models
First Principle - mass, energy momentum balances.
Based on input-output data and only describing the overall behavior of the process.

Figure 1.1: Classification of Process Models

1.2.1. DATA BASED MODELING

Data based modeling is one of the very recently added crafts in process identification, monitoring and control. To derive models based on first principles for complex processes become difficult because of poor knowledge in terms of process kinetics, order, and parameters. The black box models are data dependent and model parameters are determined by experimental results /wet labs, hence these models are called data based models or experimental models. Unlike the white box models derived from first principles, the black box/data based models or empirical models do not describe the mechanistic phenomena of the process; they are based on input-output data and only describing the overall behavior of the process [13]. The data based models are especially appropriate for problems that are data rich, but hypothesis and/or information poor. In all the cases the availability of sufficient number of quality data points are required to propose a good model. Quality data is defined by noise free data; free of outliers is ensured by data mining and pre conditioning.

The phases in the Data based modeling are:

- System analysis
- Data collection
- Data conditioning
- Key variable analysis
- Model structure design
- Model identification
- Model evaluation

Types of Data Based Models:

Data based models can be divided in to two major categories namely:

- ***Unsupervised models:*** These are the models which try to extract the different features present in the data without any prior knowledge of the patterns present in the data. Examples are Principal Component Analysis (PCA), Hierarchical Clustering Techniques (Dendrograms), non-hierarchical Clustering Techniques (*K*-means).

- ***Supervised models:*** These are the models which try to learn the patterns in the data under the guidance of a supervisor who trains these models with inputs along with their corresponding outputs. Examples include Artificial Neural Networks (ANN), Partial Least Squares (PLS) and Auto Regression Models etc.

In this era of data explosion, rational as well as potential conclusions can be drawn from the data by the help of data based modeling like Partial least squares analysis (PLS), Neural networks, Fuzzy, and Neuro Fuzzy. Principal component analysis (PCA), Independent component analysis (ICA), Canonical analysis, PLS, clustering analysis which are being used for data based modeling are all chemo-metric techniques. In this regard, we owe a profound debt to multivariate statistics. Efficient data mining, hence, efficient data based modeling will enable the future era to exploit the huge database available; in newer dimensions and perspective; embraced with never expected possibilities.

In the present work, MATLAB 7.6 & STATISTICA 9.0 were used to implement all the machine learning algorithms.

1.3. MOTIVATION

In this era of data explosion, rational as well as potential inferences can be drawn from the data with the help of data based modeling like Partial least squares analysis (PLS), Neural networks, Fuzzy, and Neuro Fuzzy, Principal component analysis (PCA), Independent component analysis (ICA), Canonical analysis, PLS, and different clustering techniques. In this regard, we owe a profound debt to multivariate statistics. Considering correlated and non-linear nature, non-stationary and multi-scale behavior of the chemical and biochemical processes, data driven/ chemo-metric techniques seems to be the logical choice for process identification and monitoring. Efficient data mining, hence, efficient data based modeling may enable the future era to exploit the huge database available; in newer dimensions and perspective; embraced with never expected possibilities.

1.4. OBJECTIVES

The main objectives of the present project are as follows:

1. **Process Identification:** Organic pollutant phenol was degraded by using bacteria named *Pseudomonas putida* (ATCC: 11172). In a batch reactor, four parameters namely temperature, pH, RPM and phenol dosage were varied systematically using experimental design (Taguchi method L'16) techniques to produce a set of useful data. By using this dataset; the phenol degradation as a rate process was identified (modeled) using supervised techniques ANN and ARX. In an effort to develop an alternative rate model without the help of fundamental kinetic data of the very process, present work was taken up to develop data based rate models. PLS technique was used to develop an empirical model relating phenol degradation process with the variables like temperature, pH, RPM and Phenol loading at steady state.
2. **Process Quality monitoring:** Development of a feature based classifier could circumvent the problem of monitoring food quality without relating instrumental analysis to biological sensing like ageing and spoilage of the product and it is one of the significant steps of on-line product quality

monitoring. A wine data set containing 178 samples and their corresponding 13 features was taken as a case study. The unsupervised technique like PCA and *K*-means clustering were used to reduce the dimensionality and classify the samples into three groups. Followed by the development of supervised classifiers using various machine learning algorithms like Artificial Adoptive Resonance theory network (ARTI), Probabilistic neural network (PNN) and Partial least squares (PLS).

3. **Process Fault detection:** For the phenol degradation process, three experimental runs were used to produce time series data which were being used for the univariate and multivariate statistical monitoring of phenol degradation process. Different SPC charts and PCA were used monitoring the process of phenol degradation; hence identification of process faults, if any.

1.5. ORGANIZATION OF THESIS

Chapter 1 presents the abridged introduction of the thesis with its overview on the perspective of the present state of art, and the objectivity of the thesis with its organization. **Chapter 2**presents a detail discussion on modeling; especially data based modeling with a mention to PCA, PLS, clustering, ARX and different types of neural networks, which were used in the subsequent chapters. This chapter also presents a concise discussion on SPCA with a special mention to different types of control charts used for process monitoring purpose, hence fault detection. **Chapter 3** presents the identification of phenol degradation process. In the present work, the organic pollutant phenol was degraded by bacteria named *Pseudomonas Putida* (ATCC: 11172). Four parameters namely temperature, pH, RPM and phenol dosage were varied systematically using experimental design (Tugachi method) techniques to produce useful data.

Chapter 4 is about process monitoring to ensure product/process quality. A wine data set was taken as a case study for process quality. This chapter is also about stringent maintenance of normal operating condition of the phenol degradation process by detecting faults. In an ending note, **Chapter 5** concludes the thesis with future recommendations.

REFERENCES:

1. Aksu, S., & Yener, J., 1998. Investigation of biosorption of phenol and monochlorinated phenols on the dried activated sludge. Process Biochem., 33, 649–655.
2. Patterson, J.N., 1997. Waste Water Treatment Technology. Ann Arbor Science, New York.
3. Bulbul, G., & Aksu, Z., 1997. Investigation of wastewater treatment containing phenol using free and Ca-alginated gel immobilized *Pseudomonas putida* in a batch stirred reactor. Turkish J. Eng. Environ. Sci., 21, 175–181.
4. Sung, R.H., Soydoa, V., & Hiroaki, O., 2000. Biodegradation by mixed microorganism of granular activated carbon loaded with a mixture of phenols. Biotechnol. Letters, 22, 1093–1096.
5. Legin, A., Rudnitskaya, A., Vlasov, Y., Natale, C. D., Davide, F., & D'Amico, A. 1997. Tasting of beverages using an electronic tongue. Sensors and Actuators, B 44, 291-296.
6. Burrati, S., Ballabio, D., Benedetti, S., & Cosio, M.S. 2007. Prediction of Italian red wine sensorial descriptors from electronic nose, electronic tongue and spectrophotometric measurements by means of Genetic Algorithm regression models. Food Chemistry, 100, 211-218.
7. Parra, V., Arrieta, A.A., Fernández-Escudero, J.B., Ro¬dríguez-Méndez, M.L., & De Saja, J.A. 2006. Electronic tongue based on chemically modified electrodes and voltammetry for the detection of adulterations in wines. Sensors and Actuators, B 118, 448-453.

8. Buratti, S., Benedetti, S., Scampicchio, M., & Pangerod E.C. 2004. Characterization and classification of Italian Barbera wines by using an electronic nose and an amperometric electronic tongue. Analytica Chimicha Acta, 525, 133-139.
9. Riul, A., de Sousa, H.C., Malmegrim, R.R., dos Santos, D.S., Carvalho, A.C.P.L.F., Fonseca, F.J., Oliveira, O.N., & Mattoso, L.H.C. 2004. Wine classification by taste sensors made from ultra-thin films and using neural networks. Sensors and Actuators, B 98, 77-82.
10. Di Natale, C., Paolesse, R., Burgio, M., Martinelli, E., Pennazza, G., & D'Amico, A. (2004). Application of metalloporphyrins - based gas and liquid sensor arrays to the analysis of red wine. Analytica Chimica Acta, 513, 49-56.
11. Legin, A., Rudnitskaya, A., Lvova, L., Vlasov, Y., Di Natale, C., & D'Amico, A. 2003. Evaluation of Italian wine by the electronic tongue: recognition, quantitative analysis and correlation with human sensory perception. Analytica Chimicha Acta, 484, 33-44.
12. Di Natale, C., Paolesse, R., Macagnano, A., Mantini, A., D'Amico, A., Ubigli, M., Legin, A., Lvova, L., Rudnitskaya, A., & Vlasov, Y. 2000. Application of a combined artificial olfaction and taste system to the quantification of relevant compounds in red wine. Sensors and Actuators, B 69, 342-347.
13. Roffel, B. & Betlem, B. 2006. Process dynamics and control: modeling for control and prediction. John Wiley & Sons, West Sussex, England.

CHAPTER 2- RELATED WORK AND COMPUTATIONAL TECHNIQUES

2.1. RELATED WORK

Over the last decade, the use of data based modeling techniques have gain a huge momentum in process identification, process fault diagnosis, and process control. The mature data collection technology has catalyzed the very activity. Chemical and biochemical processes are inherently non-linear, correlated in nature, shows non-stationarities and multi-scale behavior. Knowledge gathered from the process might have been a natural and logical choice for monitoring and controlling such a process. In this regard, some of the research efforts made by the previous researchers deserve mentioning.

Chen et. al. (2002) have integrated two data driven techniques, neural network (NN) and principal component analysis (PCA) to develop a method called NNPCA for process monitoring[1]. In this method NN was used to summarize the operating process information into a nonlinear dynamic mathematical model and PCA was employed to generate simple monitoring charts based on the multivariable residuals derived from the difference between the process measurements and the neural network predictions. Examples from the recent monitoring practice in the industry and the large-scale system in the Tennessee Eastman process problem were presented.

Zhao et. al. (2007) have introduced a new STMPCA (soft-transition multiple PCA) modeling method to avoid misclassification problems associated with simple stage-based sub-PCA while monitoring batch processes [2]. The method was based on the idea that process transition could be detected by analyzing changes in the loading matrices, which revealed evolvement of the underlying process behaviors. They proposed that by setting a series of multiple PCA models with time-varying covariance structures, which reflected the diversity of transitional characteristics and could preferably solve the stage-transition monitoring problem in multistage batch processes. The superiority of the proposed method was illustrated by applying it to both the real three-tank system and the simulation of benchmark fed-batch penicillin fermentation process with more reliable monitoring charts. Both results of real experiment and simulation clearly demonstrated the effectiveness and feasibility of the proposed method.

Gaetano et. al. (2009) have designed a novel supervised neural network-based algorithm to reliably distinguish the electrocardiographic (ECG) records between normal and ischemic beats of the same patient [3]. The basic idea was to consider an ECG digital recording of two consecutive R-wave segments (RRR interval) as a noisy sample and an underlying function was approximated by a fixed number of Radial Basis Functions (RBF). The linear expansion coefficients of the RRR interval represent the input signal of a feed-forward neural network which classified a single beat as normal or ischemic. The developed system used several patient records taken from the European ST-T database. The obtained results showed that the proposed algorithm offered a good combination of sensitivity and specificity, making the design of a practical automatic ischemia detector feasible.

Meleiro et. al (2009) have employed a constructive learning algorithm to design a near-optimal one- hidden layer neural network structure that approximated the dynamic behavior of a bioprocess [4].The method determined not only a proper number of hidden neurons but also the particular shape of the activation function for each node. Here, the projection pursuit technique was applied in association with the optimization of the solvability condition, giving rise to a more efficient and accurate computational learning algorithm. Each activation function of a hidden neuron is defined according to the peculiarities of each approximation problem guiding to parsimonious neural network architectures. The proposed constructive learning algorithm was successfully applied to identify a MIMO bioprocess, providing a multivariable model that was able to describe the complex process dynamics, even in long-range horizon predictions. The identified model was considered as part of a model-based predictive control strategy, producing high-quality performance in closed-loop experiments.

Sadrzadeha et.al. (2009) have used a simple two layered feed forward MLP neural network to predict separation percent (SP) of lead ions from wastewater using electro-dialysis (ED) [5]. The aim was to predict SP of Pb^{2+} as a function of concentration, temperature, flow rate and voltage. Once optimum numbers of hidden layers and nodes in each layer were determined, the selected structure (4:6:2:1) was used for prediction of SP of lead ions as well as current efficiency (CE) of ED cell for different inputs in the domain of training data. They have claimed that ANN successfully tracked the non-linear behavior of SP and CE versus temperature, voltage, concentration and flow rate with standard deviation not more than 1%.

Jyh-Cheng Jeng (2010) presented the use of both recursive PCA (RPCA) and moving window based PCA (MWPCA) for online updation of the PCA model and its corresponding control limits for monitoring statistics [6]. He derived an efficient algorithm based on rank one matrix update of the covariance matrix, which was tailored for RPCA and MWPCA computations. He demonstrated the complete monitoring system through simulation examples and the results had shown the effectiveness of the proposed method.

Marchitana et. al. (2010) have attempted to compare two popular non-parametric modeling and optimization techniques, response surface methodology (RSM) and artificial neural network (ANN) for reactive extraction of tartaric acid from aqueous solution using Amberlite LA-2 (amine) [7]. The extraction efficiency was modeled and optimized as a function of three input variables, i.e. tartaric acid concentration in aqueous phase CAT (g/L), pH of aqueous solution and amine concentration in organic phase CA/O (% v/v). Both methodologies were compared for their modeling and optimization abilities. According to analysis of variance (ANOVA) the coefficient of multiple determination of 0.841 was obtained for RSM and 0.974 for ANN. The optimal conditions offered by RSM and genetic algorithm (GA) led to an experimental extraction efficiency of 83.06%. On the other hand, optimal conditions offered by the ANN model coupled with GA led to an experimental reactive extraction efficiency of 96.08%.

Bin-Shams et. al. (2011) have used a CUSUM based statistical monitoring scheme to monitor a particular set of Tennessee Eastman Process (TEP), which were early monitored by using contribution plots [8]. Contribution plots were found to be inadequate when similar variable responses were associated with different faults. Abnormal situations from the process historical database were then used in combination with the proposed CUSUM based PCA model to unambiguously characterize the different fault signatures. The use of a family of PCA models trained with CUSUM transformations of all the available measurements collected during individual or simultaneous occurrence of the faults were found effective in correctly diagnosing these faults.

Pendashteh et. al. (2011) have employed a feed-forward neural network trained by batch back propagation algorithm to model a membrane sequencing batch reactor (MSBR) treating hypersaline oily wastewater [9]. The MSBR operated at different total dissolved solids (TDSs) (mg/L), various organic loading rates (OLRs) (kg COD/(m^3 day)) and cyclic time (h). They have used a set of 193 operational data from the wastewater treatment with the MSBR to train the network. The training, validating and testing procedures for the effluent COD, total organic carbon (TOC) and oil and grease (O&G) concentrations were successful and a good correlation was observed between the measured and predicted values. The results of this study showed that ANN-GA could easily be applied to evaluate the performance of a membrane bioreactor even though it involved the highly complex physical and biochemical mechanisms associated with the membrane and the microorganisms.

In view of this, present work was taken up to identify the Phenol degradation process and diagnosing its faults with a view to monitor it. ANN and PLS based classifiers were designed as an integral part of wine quality monitoring.

2.2. COMPUTATIONAL TECHNIQUES

2.2.1. CLUSTERING

Clustering techniques were adopted for the classification of wine samples in the present project. Before going for the classification or development of classifier, one needs to identify different classes present in the data in an accurate manner using the historical data thus ensuring the efficiency of the classifier developed. Clustering techniques comes handy for the purpose. Clustering technique is more primitive in that; no a-priori assumptions are made regarding the group structures. Grouping can be made on the basis of similarities or distances (dissimilarities). Hierarchical clustering techniques processed either by a series of successive mergers or a series of successive divisions. Agglomerative hierarchical methods start with individual objects ensuring as much number of clusters as objects initially. The most similar objects (or less Inter-cluster distance) are grouped first, and these initial groups are merged according to their similarities. Eventually as the similarities decreases (distance increases), all subgroups are fused into a single cluster. The divisive hierarchical method works in the opposite direction. An initial single group of objects is divided into two subgroups such that they are far from each other. This subdivision continues until there are as many subgroups as objects; that is until each object forms a group. The results of both the methods may be displayed in the form of a two-dimensional diagram called dendrogram. Inter-cluster distances are expressed by single linkage, complete linkage and average linkage. In the present work the agglomerative hierarchical method was applied to group the different wine samples not the variables. Non-hierarchical, unsupervised method, *K*-means clustering was also applied in this work. The number of clusters *K* can be pre-specified or can be determined iteratively as a part of the clustering procedure. The K-means clustering proceeds in three steps, which are as follows:

1. Partitioning of the items in to *K* initial clusters.
2. Assigning an item to the cluster whose centroid is nearest (distance is usually a Euclidian). Recalculation of the centroid for the cluster receiving the new item and for the cluster losing the item.
3. Repeating the step-2 until no more reassignment takes place or stable cluster tags are available for all the items.

The basic algorithm of K-means is as follows:

- For a given assignment *C*, computation of the cluster mean m_k:

$$m_k = \frac{\sum\limits_{i:C(i)=k} x_i}{N_k}, k = 1,..., K. \tag{2.1}$$

- For a current set of cluster means, assigning each observation as:

$$C(i) = \arg\min_{1 \le k \le K} \left\| x_i - m_k \right\|^2, i = 1,..., N$$

(2.2)

- Iteration of the above two steps until convergence.

The K-means clustering has a specific advantage of not requiring the distance matrix as required in hierarchical clustering, hence ensures a faster computation than the latter. The K-means algorithm has been applied to many engineering problems [10-14].

2.2.2. Principal Component Analysis

The two major advantages of principal component analysis are pattern recognition and dimensionality reduction. Both these features of PCA were explored in wine classification and the pattern recognition capacity was explored in monitoring of phenol degradation process. PCA is a multivariate statistical technique initially used to analyze data from process plants. PCA offers a new set of uncorrelated variables that are a linear combination of the original variables. The new variables capture the maximum variance in the original data set in ascending order. The new variables are called the ‘principal components’ and they are estimated from the eigenvectors of the covariance or correlation matrix of the original variables. PCA was originally developed by Pearson, 1901. The PCA model can be used to detect outliers in data, data reconciliation, and deviations from normal operation condition that indicate excessive variation from normal target or unusual patterns of variation. Operation under various known upsets can also be modeled if sufficient historical data are available to develop automated diagnosis of source causes of abnormal process behavior [15]. Depending on the field of application, it is also named the discrete Karhunen–Loève transform (KLT), the Hotelling transform or proper orthogonal decomposition (POD).

The applicability of PCA is based on certain assumptions, which are as follows,

- Assumption on Linearity

- Assumption on the statistical importance of mean and covariance
- Assumption that large variances have important dynamics

A principal component analysis is concerned with explaining the variance-covariance structure of a set of variables through a few linear combinations of these variables. Its general objectives are first data reduction and second interpretation.

Generally, this is a mathematical transform used to find correlations and explain variance in a data set. The goal is to map the raw data vector *E* onto vectors *S*, where, the vector x can be represented as a linear combination of a set of *m* ortho-normal vectors u_i

$$x = \sum_{i=1}^{m} z_i u_i \tag{2.3}$$

where the coefficients z_i can be found from the equation $z_i = {u_i}^T x$

This corresponds to a rotation of the coordinate system from the original x to a new set of coordinates given by z. To reduce the dimensions of the data set, only a subset (*k* <*m*) of the basic feature vectors are preserved. The remaining coefficients are replaced by constants b_i and each vector *x* is then approximated as

$$\check{x} = \sum_{i=1}^{m} z_i u_i + \sum_{i=1}^{d} b_i u_i \tag{2.4}$$

The basic vectors u_i are called principal components which are equal to the eigenvectors of the covariance matrix of the data set. The coefficients $b_i$and the principal components should be chosen such that the best approximation of the original vector on an average is obtained. However, the reduction of dimensionality from m to k causes an approximation error. The sum of squares of the errors over the whole data set is minimized if we select the vectors $u_i$that correspond to the largest Eigen values of the covariance matrix. As a result of the PCA transformation, the original data set is represented in fewer dimensions (typically 2-3) and the measurements can be plotted in the same coordinate system. This plot shows the relation between different observations or experiments. Grouping of data points in this plot suggest some common properties and those can be used for classification.

We had a number of wine samples with different features/properties. Part of the available measurements can be used as a training set to define the classes, while the rest can be kept out for validation purposes. Assuming *n* measurements are used for training and *p* for validation, the training data is organized in a single matrix of the following form:

$$\mathbf{X}=\begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \tag{2.5}$$

where, each row in $\boldsymbol{X}$ represents one measurement and the number of columns *m* is equal to the length of the measurement sequence or features. Following the step described above, the covariance matrix$\boldsymbol{C} = cov(\boldsymbol{X})$and its Eigen values λ were calculated. Its eigenvectors $u_i$form an orthonormal basis $= [u_1, u_2, u_3, \dots u_m]$; that is $\boldsymbol{U}^T\boldsymbol{U} = 1$. The original data set can be represented in the new basis using the relation: $\boldsymbol{Z} = \boldsymbol{U}^T\boldsymbol{X}$

After this transformation, a new data matrix of reduced dimension can be constructed with the help of Eigen values of the matrix $\boldsymbol{C}$. This is done by selecting the highest λ values since they correspond to the principal components with highest significance. The number of PCs to be included should be high enough to ensure good separation between the classes. Principal components with low contribution (low values of λ) should be neglected. Let the first *k* PCs as new features be selected neglecting the remaining (*m-k*) principal components. In this way, a new data matrix *D* of dimension *n* × *k* was obtained.

$$\boldsymbol{D}=\begin{bmatrix} z_{11} & \cdots & z_{1k} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nk} \end{bmatrix} \tag{2.6}$$

With the matrix D is defined, the next step is directed towards classification of substances. The matrix ***U*** is used during the validation and also plays a key role in the online implementation of the classification algorithm. The PCA score data sets are grouped into number of classes following the rule of nearest neighborhood clustering algorithm. The above reduced data matrix is utilized for construction of class prototypes. Let $(C)_{i=1\ldots l}$ denote *l* pattern classes in n number of measurements, represented by the single prototype vector$(y)_{i=1\ldots l}$. The maximum value of *l* can reach up to n.

The mean or class centroids of $(y)_{i=1\ldots l}$ vectors have *m* numbers of latent features, each of which represents unique feature in reduced dimension space. The distance between an incoming pattern x and the prototype vectors are$D_j = \|x - y_i\|,\ 1 \leq i \leq l.$ The minimum distance will classify *x* at C_j for which D_j is minimum:

$$D_j = min\|x - y_i\|,\ for\ 1 \leq i \leq l. \tag{2.7}$$

For online system, it may be inferred that the incoming pattern represented by unknown type has similarity with the one of the l^{th} class of known types.

2.2.3. Partial Least Squares Model

The data of Phenol degradation process was used for the identification of this process with the help of Partial Least Squares (PLS) technique. Partial least squares is one of the important multivariable statistical techniques to reduce the dimensionality of the plant data, to find the latent variables from the plant data by capturing the largest variance in the data and achieves the maximum correlation between the predictor (X) variables and response (Y) variables. First proposed by Wold (1966) [16]. PLS has been successfully applied in diverse fields including process modeling, identification of process dynamics & fault detection, process monitoring and it deals with noisy and highly correlated data, quite often, only with a limited number of observations available. A tutorial description along with some examples on the PLS model was provided by Geladi and Kowalski (1986) [17]. When dealing with nonlinear systems, the underlying nonlinear relationship between predictor variables (X) and response variables (Y) can be approximated by quadratic PLS (QPLS) or splines. Sometimes it may not function well when the non-linearities cannot be described by quadratic relationship. Qin and McAvoy (1992)[18] suggested a new approach to replace the inner model by neural network model followed by the focused R & D activities taken up by several other researchers like Holcomb & Morari (1992); Malthouse et al.(1997); Zhao et al.(2006); Lee et al. (2006)[19-22]. The mathematical formulation of static PLS is as follows:

Input - output data were generated by exciting the processes with pseudo random binary signals (PRBS). X and Y matrices are scaled in the following way before they are processed by PLS algorithm.

$$X = XS_X{}^{-1} \text{ and } Y = YS_Y{}^{-1} \tag{2.8}$$

Where $S_X = \begin{bmatrix} s_{x1} & 0 \\ 0 & s_{x2} \end{bmatrix}$ and $S_Y = \begin{bmatrix} s_{y1} & 0 \\ 0 & s_{y2} \end{bmatrix}$

The S_X and S_Y are scaled matrices. The idea of PLS is to develop a model by relating the scores of X and Ydata. PLS model consists of outer relations (X&Ydata being related to their scores individually) and inner relations that links X data scores to Ydata scores. The outer relationship for the input matrix and output matrix with predictor variables can be written as

$$X = t_1 p_1^T + t_2 p_2^T + ... + t_n p_n^T + E = TP^T + E \tag{2.9}$$

$$Y = u_1 q_1^T + u_2 q_2^T + ... + u_n q_n^T + F = UQ^T + F \tag{2.10}$$

where, $T$and U represents the matrices of scores of X and Y while $P$and Q represent the loading matrices for X and Y. If all the components of $X$and Y are described, the errors E &Fbecome zero. The inner model that relates X to$Y$ is the relation between the scores T&U.

$$U = TB \tag{2.11}$$

Where B is the regression matrix. The response Y can now be expressed as:

$$Y = TBQ^T + F \tag{2.12}$$

To determine the dominant direction of projection of $X$and Y data, the maximization of covariance within X and Y is used as a criterion.The first set of loading vectors p_1 and q_1 represent the dominant direction obtained by maximization of covariance within $X$and Y. Projection of X data on p_1 and Y data on q_1 resulted in the first set of score vectors $t_1$and u_1 , hence the establishment of outer relation. The matrices X and Y can now be related through their respective scores, which is called the inner model, representing a linear regression between $t_1$and u_1: $\hat{u}_1 = t_1 b_1$. The calculation of first two dimensions is shown in Fig. 2.1.

The residuals are calculated at this stage is given by the following equations.

$$E_1 = X - t_1 p_1^{'} \tag{2.13}$$

$$F_1 = Y - u_1 q_1^{'} = Y - t_1 b_1 q_1^{'} \tag{2.14}$$

The procedure for determining the scores and loading vectors is continued by using the newly computed residuals till they are small enough or the number of PLS dimensions required are exceeded. In practice, the number of PLS dimensions is calculated by percentage of variance explained and cross validation. The irrelevant directions originating from noise and redundancy are left as E and F. The multivariate regression problems decomposed into several univariate regression problems with the application of PLS.

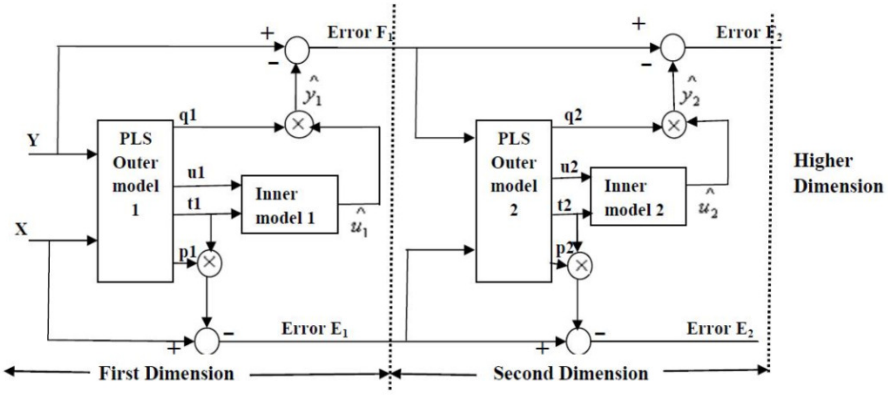

Figure 2.1: Standard linear PLS algorithm.

2.2.4. Statistical Process Monitoring (SPM) Charts

The goal of *statistical process monitoring* (SPM) is to detect the existence, magnitude, and time of occurrence of changes that cause a process to deviate from its desired operation. The methodology for detecting changes is based on statistical techniques that deal with the collection, classification, analysis, and interpretation of data. Traditional statistical process control (SPC) has focused on monitoring quality variables at the end of a batch and if the quality variables are outside the range of their specifications, making adjustments (hence control the process) in subsequent batches. An improvement of this approach is to monitor quality variables during the progress of the batch and make adjustments if they deviate from their expected ranges. Monitoring quality variables usually delays the detection of abnormal process operation because the appearance of the defect in the quality variable takes time. Information about quality variations is encoded in process variables. The measurement of process variables is often highly automated and more frequent, enabling speedy refinement of measurement information and inferencing about product quality. Monitoring of process variables is useful not only for assessing the status of the process, but also in controlling the product quality. When the process monitoring indicates abnormal process operation, diagnosis operations are initiated to determine the source cause of this abnormal behaviour. In this framework, each quality variable is treated as a single independent variable.The abnormal operating conditions in the Phenol degradation process were detected using traditional univariate statistical control charts like $\bar{x}$ Charts, R charts, Moving Range charts and CUSUM charts.The theoretical postulations of control limits charts are as follows:

Traditional statistical monitoring techniques for quality control of batch products relied on the use of univariate SPC tools on product quality variables. Before going into any further details of the control charts, one should have a brief idea about statistical hypothesis testing. A *statistical hypothesis* is an assumption or a guess about the population expressed as a statement about the parameters of the probability distributions of the populations. Procedures that enable decision making whether to accept or reject a hypothesis are called *tests of hypotheses.* For example, if the equality of the mean of a variable (μ) to a value 'a'is to be tested, the hypotheses are:

Null hypothesis: $H_0 : \mu = a$

Alternate hypothesis: $H_1 : \mu \neq a$

Two kinds of errors may occur while testing the hypothesis:

Type I error (α): $P\{reject\ H_0 | H_0\ is\ true\}$

Type II error (β): $P\{fail\ to\ reject\ H_0 | H_0\ is\ false\}$

First $\alpha$is selected to compute the confidence limit for testing the hypothesis then a test procedure is designed to obtain a small value for β, if possible. β is a function of sample size and is reduced as sample size increases.

Three parameters affect the control limit selection:

- The estimate of average level of the variable
- The variable spread expressed as range or standard deviation,
- A constant based on the probability of Type I error; α.

The "3 σ " *(* σ denoting the standard deviation of the variable) control limits are the most popular control limits. The constant 3 yields a Type I error probability of 0.00135 on each side (α = 0.0027). The control limits expressed as a function of population standard deviation σ are:

$$UCL = Target + 3\sigma \qquad LCL = Target - 3\sigma \tag{2.15}$$

2.2.4.1. Range Charts

Development of $\bar{\boldsymbol{x}}$ starts with the R charts. Since the control limits of the $\bar{\boldsymbol{x}}$chart depends on process variability, its limits are not meaningful before R is in-control.Range is the difference between the maximum and minimum observations in a sample.

$$R_i = x_{\max i} - x_{\min i} \qquad \overline{R} = \frac{1}{m}\sum_{i=1}^{m} R_i \tag{2.16}$$

The random variable *R/* σ is called the *relative range.* The parameters of its distribution depend on sample size *n*, with the mean being d_2. An estimate of σ (the estimates are denoted by $\hat{\sigma}$ ') can be computed from the range data by using

$$\hat{\sigma} = \frac{\overline{R}}{d_2} \tag{2.17}$$

Where d_2 is called Hartley's constant

The standard deviation of $\overline{R}$ is estimated by using the standard deviation of *R/* σ *,* d_3,:

$$\hat{\sigma}_R = d_3\sigma = d_3\frac{\overline{R}}{d_2} \tag{2.18}$$

The control limits of the R chart are:

$$UCL, LCL = \overline{R} \pm 3d_3 \frac{\overline{R}}{d_2} \tag{2.19}$$

Defining

$$D_3 = 1 - 3\frac{d_3}{d_2} \text{ and } D_4 = 1 + 3\frac{d_3}{d_2} \tag{2.20}$$

The control limits become

$$UCl = \overline{R}D_3 \text{ and } LCL = \overline{R}D_4 \tag{2.21}$$

2.2.4.2. X-bar Charts

One or more observations may be made at each sampling instant. The collection of all observations at a specific sampling time is called a sample.

$$\overline{X_i} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \overline{\overline{X_{ij}}} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}$$

Where m is the number of samples and n is number of observations in a sample (sample size).

The estimator for the mean process level (centerline) is $\overline{\overline{X}}$.

Since the estimate of the standard deviation of the mean process levelσ is $\frac{\overline{R}}{d_n}$,

$$UCL / LCL = \overline{\overline{\text{X}}} \pm (A_2 \times \overline{R}) \tag{2.22}$$

$$\text{Where } A_2 = \frac{3}{d_2\sqrt{n}} \tag{2.23}$$

Where n is the number of readings, d_2 is Hartley's constant

A typical X-bar and R-chart was shown in Fig. 2.2.

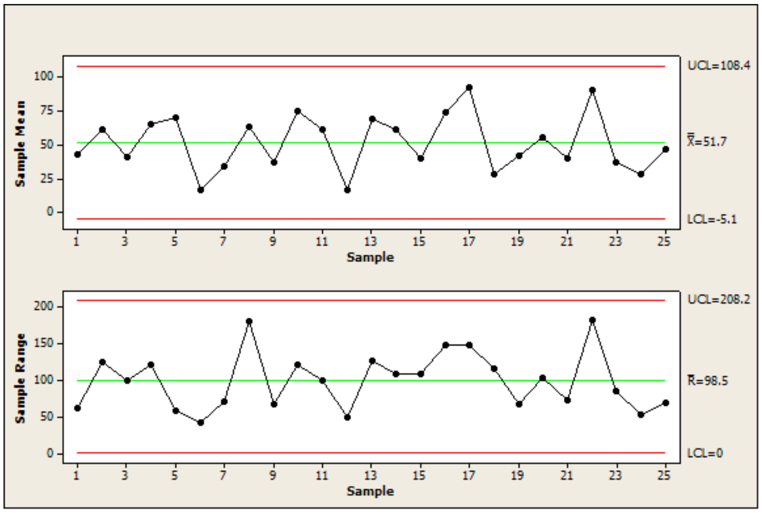

Figure 2.2: A typical X-bar and R- Chart.

2.2.4.3. CUSUM CHARTS

The cumulative sum (CUSUM) chart incorporates all the information in a data sequence to highlight changes in the process average level. The values to be plotted on the chart are computed by subtracting the overall mean μ_0 from the data and then accumulating the differences.For a sample size n ≥ 1, denote the average of the j^{th} sample x_j. The quantity

$$S_i = \sum_{j=1}^{i} (\bar{x}_j - \mu_0) \tag{2.24}$$

is plotted against sample number i. CUSUM charts are very effective in detecting *small process shifts*, since they combine information from several samples. CUSUM charts are very effective with samples of size 1. The CUSUM values can be computed recursively.

$$S_i = (x_i - \mu_0) + S_{i-1} \tag{2.25}$$

If the process is in-control at the target value μ_0, the CUSUM S_i should meander randomly in the vicinity of 0. If the process mean is shifted, an *upward or downward trend* will develop in the plot. Visual inspection of changes of slope indicates the sample number (and consequently the time) of the process shift. Even when the mean is on target, the CUSUM S_i may wander far from the zero line and give the appearance of a signal of change in the mean. Control limits in the form of a V-mask were employed when CUSUM charts were first proposed in order to decide that a statistically significant change in slope has occurred and the trend of the CUSUM plots different than that of a random walk. CUSUM plots generated by a computer became more popular in recent years and the V-mask has been replaced by upper and lower confidence limits of one-sided CUSUM charts. *One-Sided* CUSUM charts are developed by plotting

$$S_i = \sum_{j=1}^{i} [\bar{x}_j - (\mu_0 - K)] \tag{2.26}$$

Where, *K* is the *reference value* to detect an increase in the mean level. If S_i becomes negative for $\mu_1 > \mu_0$, it is reset to zero. When S_i exceeds the decision interval *H,* a statistically significant increase in the mean level is declared. Values for *K* and *H* can be computed from the relations:

$$K = \frac{\Delta}{2} \qquad H = \frac{d\Delta}{2} \tag{2.27}$$

Given the probabilities of type 1 (α) and type 2 (β) errors, the size of the shift in the mean to be detected (Δ), and the standard deviation of the average value of the variable $\bar{x}(\sigma_{\bar{x}})$, the parameters in above equation are as follows:

$$d = \frac{2}{\delta^2} \ln\left(\frac{1-\alpha}{\beta}\right) \qquad \text{Where } \delta = \frac{\Delta}{\sigma_{\bar{x}}} \tag{2.28}$$

2.2.4.4. Moving Range Charts

In a moving-range chart, the range of two consecutive sample groups of size a are computed and plotted. For $a \geq 2$

$$MR_t = \left|\max(i) - \min(i)\right|, \tag{2.29}$$

Where *i* is the subgroup containing samples from '*t*-*a*+1' to '*t*'.

The computation procedure is as follows:

- Selecting the moving range size *a*. Often $a = 2$

- Obtaining the estimates of $\overline{MR}$ and $\sigma = MR / d_2$ by using the moving-ranges MR_t of length a. For a total of m samples:

$$\overline{MR} = \frac{1}{m - a + 1} \sum_{t=1}^{m-a+1} MR_t \tag{2.30}$$

- Computing the control limits with the center line at $\overline{MR}$:

$$LCL = D_3 \times \overline{MR} \;\; UCL = D_4 \times \overline{MR} \tag{2.31}$$

2.2.5. Artificial Neural Networks

Artificial Neural Networks (ANNs) are widely applied nowadays for classification, identification, control, diagnostics, recognition, etc. An Artificial Neural Network (ANN) is an information processing paradigm that is stimulated by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. Basically, a neural network (NN) is composed of a set of nodes (Fig. 2.3). Each node is connected to the others via a set of links. Information is transmitted from the input to the output cells depending on the strength of the links. Usually, neural networks operate in two phases. The first phase is a learning phase where each of the nodes and links adjust their strength in order to match with the desired output. A learning algorithm is in charge of this process. When the learning phase is complete, the NN is ready to recognize the incoming information and to work as a pattern recognition system.

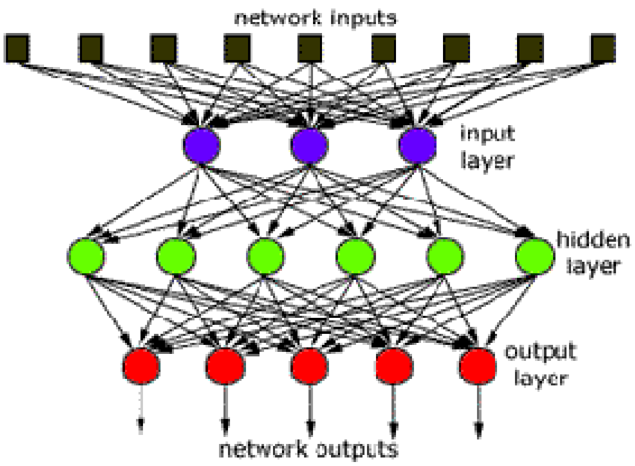

Figure 2.3: General Neural Network Architecture

2.2.5.1. Neural Networks as Classifiers

Neural networks, either supervised or unsupervised have emerged as an important tool for classification. The recent vast research activities in neural classification have established that neural networks are a promising alternative to various conventional classification methods. The advantage of neural networks lies in the following theoretical aspects. First, neural networks are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Second, they are universal functional approximators in that neural networks can approximate any function with arbitrary accuracy [23-25]. Since any classification procedure seeks a functional relationship between the group membership and the attributes of the object, accurate identification of this underlying function is doubtlessly important. Third, neural networks are nonlinear models, which makes them flexible in modeling real world complex relationships. Finally, neural networks are able to estimate the posterior probabilities, which provide the basis for establishing classification rule and performing statistical analysis [26]. In the present work, two network architectures were used for the classification purpose namely Probabilistic Neural Network (PNN) and Adaptive Resonance Theory Network (ART1).

2.2.5.1.1. PROBABILISTIC NEURAL NETWORKS (PNN)

The *probabilistic neural network* (PNN) was introduced by Donald Specht (1990) [27] and it can be used for classification problems. The multilayer feed forward network can be used to approximate non-linear functions where the network structure is sufficiently large. Any continuous function can be approximated by carefully choosing the parameters in the network. For the determination of those highly non-linear parameters learning should be based on non-traditional optimization techniques. A viable alternative is the radial basis function neural network. A RBF network uses radial basis functions as activation functions. Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer as shown in Fig. 2.4. In the basic form all inputs are connected to each hidden neuron. The inputs vector $X = [x_1, x_2, x_3, \ldots x_n]$ is applied to the neurons in the hidden layer. Each hidden layer neurons computes the following exponential functions, called Gaussian response function as given by:

$$h_i = \exp - \left[\frac{D_i^2}{2\sigma^2} \right] \text{, where } X \text{=an input} \tag{2.32}$$

$$D_i^2 = (x - u_i)^T (x - u_i) = \textit{squared distance between the input vector and training vector.} \tag{2.33}$$

u_i=weight vector of hidden layer neuron i. The weights of each hidden layer neuron are assigned the values of an input training vector. The output neuron produces the linear weighted summation of these as given by:

$$y = \sum h_i w_i \text{ , where } w_i = \textit{a weight in the output layer} \tag{2.34}$$

Sometimes the outputs are optionally normalized according to the following formula that divides the output of each neuron in the output layer by the sum of all hidden layer outputs. i.e. $out_i = \sum_i h_i w_i \Big/ \sum_i h_i$

Thus the output has a significant response to the input only over a range of values of spread parameter, called the receptive field of the neuron, the size of which is determined by the value of σ as shown in Fig. 2.4.

PNN is having its theoretical foundation on Bayesian classifier theory. PNN being basically a Radial basis neural network (RBNN) containing an extra competitive layer in addition to the radial basis layer. When an input is presented, the first layer computes distances from the input vector to the training input vectors and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a competitive transfer function on the output of the second layer picks the maximum of these probabilities, and produces a '1' for that class and a '0' for the other classes. The basic architecture of PNN is presented in Fig. 2.5.

The basis of PNN is Bayes theorem, which states:

$$P(y^i | x) = P(x | y^i) P(y^i) / P(x) \tag{2.35}$$

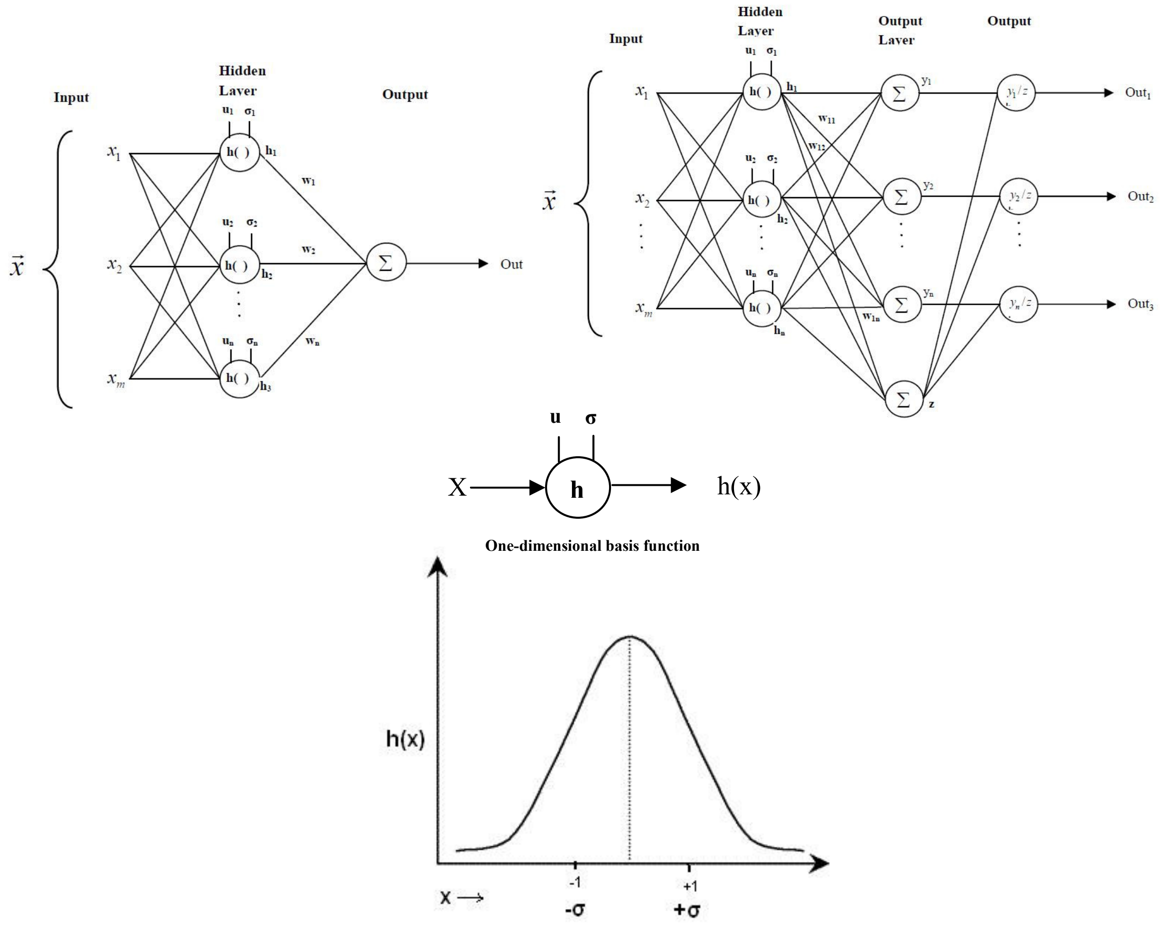

Figure 2.4: Radial Basis network architecture

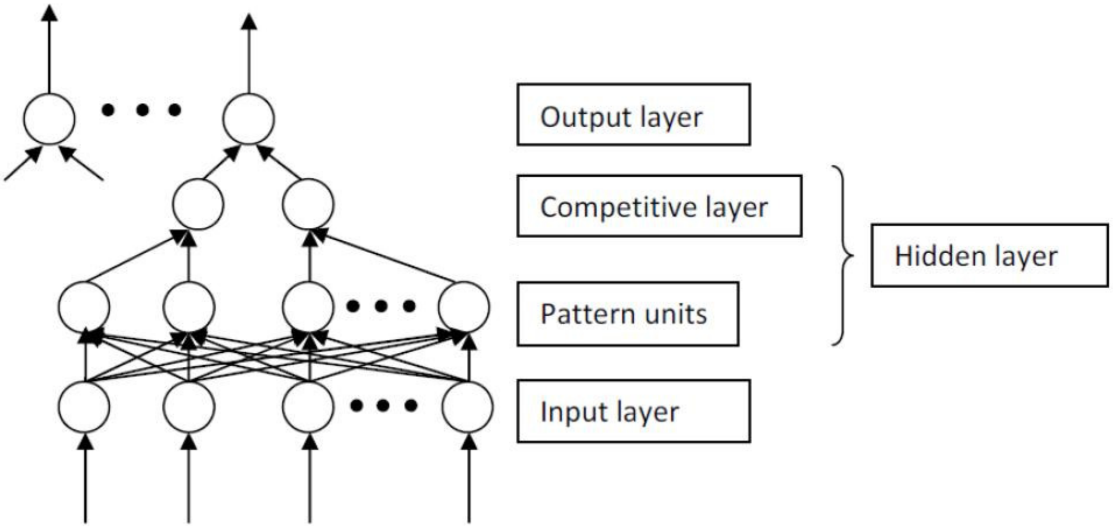

Figure2.5: Basic architecture of Probabilistic neural networks (PNN).

Let y denote the membership variable that takes a value of y_i if the object belongs to group *i*. $P(y^i)$ is the prior probability of class i and $P(x|y^i)$=Probability density function. $P(y^i|x)$ is the posterior probability of group i . $P(x)$ is the probability density function:

$$P(x) = \sum_{j=1}^{N} P(x|y^i)\ P(y^i) \tag{2.36}$$

In PNN, the probability distribution function is approximated by Parzon windows, typically using the exponential function.

By the previous definitions, $P(x|y^i) = (\frac{K}{N^i})\sum_{j=1}^{N^i} \exp(-\frac{D_j^{i2}}{2\sigma^2}) = (\frac{K}{N^i})\sum_{j=1}^{N^i} h_j^i$ (2.37)

Where $K = \frac{1}{2\pi^{d/2}\sigma^d}$ = the scaling factor to produce a multidimensional unit Gaussian, $D_j^{i2} = (x-u_i)^T(x-u_i)$ = the squared Euclidian distance between the current input x and training vector j in class i .

Applying Bayer's theorem to calculate the conditional probability of x for each class, then summing these over all classes yields,

$$P(x) = \sum_{i=1}^{N_i} P(x|y^i)P(y^i) = \sum_{i=1}^{N_i} (\frac{K}{N^i})\sum_{j=1}^{N_i} \exp[(-\frac{D_i^2}{2\sigma^2})(\frac{N^i}{N})] = \frac{K}{N}\sum_{i=1}^{N_i}\sum_{j=1}^{N_i} h_j^i \tag{2.38}$$

where, N_i is the number of classes. Thus the inner summation adds all hidden layer neuron outputs associated with class i; the outer summation counts these over all classes. The double summation may be eliminated by simply counting all hidden neuron outputs.

If we have an object with a particular feature vector x and a decision is to be made about its group membership, the probability of classification error is

$$P({}^{error}/_{x}) = \sum_{i \neq j} \left(P(y^i | x) \right)$$

$$= 1 - P(y^i | x), if\ y^i\ is\ already\ decided \qquad (2.39)$$

Hence, if the purpose is to minimize the probability of total classification error, then the following is the widely used Bayesian classification rule,

$$\text{Decide } y^i \text{ for x if } P(y^i | x) = \max_{i=1 \ldots N} P(y^i | x) \qquad (2.40)$$

2.2.5.1.2. ART1 NETWORK

The stability-plasticity dilemma remained unresolved for many conventional artificial neural networks. The ability of a net to learn new patterns equally well at any stage of learning without washing away the previously learnt patterns is called its plasticity. A stable net does not return any pattern to a previous cluster. Some nets achieve stability gradually adjusting their learning rates provided the same training set is presented many times before them. Those conventional nets cannot learn a pattern while presented first time before them. A real network is constantly exposed to changing patterns; it may never see the same training vector twice. Under such a circumstance the back propagation networks can learn nothing with continuously modifying their weights without a respite of getting a stationary setting. Artificial adoptive resonance theory networks (ART1) are designed to be both plastic and stable. The first version of ART network was "ART1", which was developed by Carpenter and Grossberg (1988). The ART1 network is a vector classifier. It accepts an input vector and classifies it as one of the categories depending upon which of the stored pattern it resembles within a specified tolerance otherwise a new category is created by storing that pattern as an input vector. No stored pattern is modified if it does not match the current input vector within a specified tolerance; hence the stability- plasticity dilemma is solved. ART1 is designed for classifying binary vectors. The classification process through ART involves three steps; recognition, comparison and the search phase. During learning one input vector is presented to the network. The degree of similarity is controlled by vigilance parameter ρ (0-1).

2.2.5.1.3. Basic Architecture

The ART1 network consists of three major components accompanying groups of neurons (Fig.2.6&2.7).

- Input processing field-F1 layer
- Cluster units –F2 layer
- Reset mechanism

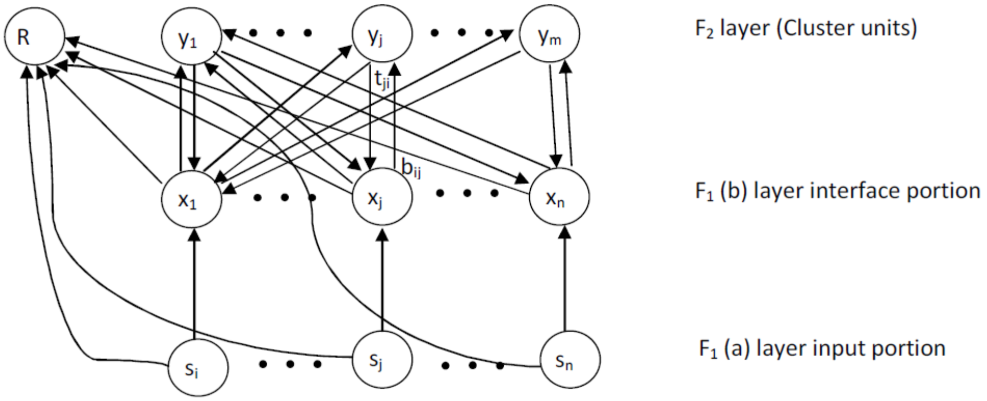
R
y_1
y_j
y_m
F_2 layer (Cluster units)
t_{ji}
b_{ij}
x_1
x_j
x_n
F_1 (b) layer interface portion
s_i
s_j
s_n
F_1 (a) layer input portion

Figure 2.6: Architecture of ART1

Input processing layer is divided into two layers

- Input portions – F1(a): Represents the given input vector
- Interface portion –F1(b): Exchanges the input portion signal with the F2 layer

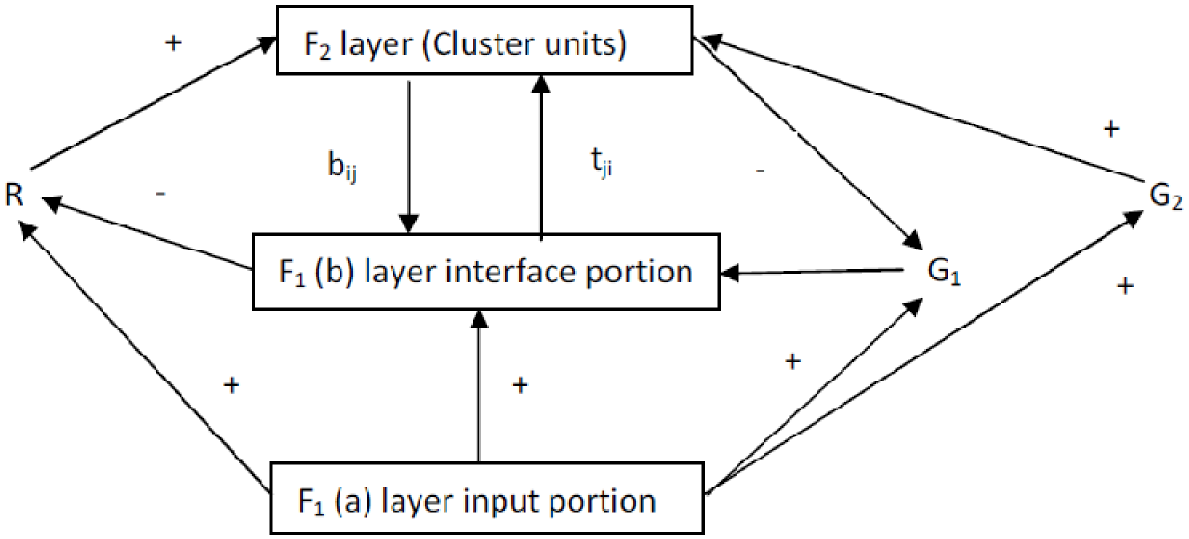

Figure 2.7: Structure of supplemental units

Cluster units –F2 layer

This is a competitive layer. The cluster unit with largest net input is selected to learn the input pattern. The activation of all other F2 units is set to zero. F1(b) is connected to F2 layer through bottom-up weights b_{ij} and F2 layer is connected to F1(b) layer by top down weight t_{ji}.

Reset Mechanism

Depending on the similarity between the top down weight and the input vector, the cluster unit is allowed to learn a pattern or not. This is done at the reset unit, based on the signals it receives from the input and interface portion of the F1 later. If the cluster unit is not allowed to learn, it becomes inhibited and a new cluster unit is selected for learning. It dictates the three possible states for F2 layer neurons; they are namely active, inactive and inhibited. The difference between the inactive and inhibited is that for both the cases activation state of F2 unit is zero. In its inactive state, the F2 neurons are available in next competition during the presentation of current input vector which is not possible when the F2 layer is inhibited.

2.2.5.1.4. ALGORITHM

The binary input vector is presented to F1 (a) layer and is then passed on to F1 (b) layer. The F1 (b) layer sends signal to F2 layer over weighted interconnection path (Bottom-up weights). Each F2 unit calculates the net input. The node with the largest input is the winner and its activation state is 1. All the other nodes in F2 layer are considered to have activation state of 0 but not inhibited and the reset is true. The winning node of F2 layer alone is eligible to learn the input pattern. The signal is sent from F2 layer to F1 (b) through weighted interconnections which are top down weights. The activation vector X of the F1 (b) layer are considered to be active if they receive non-zero weights both from F1 (a) and F2 layer.

The norm of the vector $||X||$ renders the number of components in which the top-down weight vector for the winning unit (t_{ji}) and the input vector S are both 1. Depending upon the ratio of norm of x to norm of $S\left(\frac{||X||}{||S||}\right)$, either the weights of the winning cluster units are adjusted or the reset mechanism is rescheduled. The whole process is repeated until either a match is found or all neurons in the F2 layer are inhibited. The training algorithm for ART1 is as follows,

Step1. Initialization of parameters and weights

L>1 &$< \rho \leq 1$

$0 < b_{ij}(0) < \frac{L}{L-1+n}$. Where n is the number of components in the input vector

$t_{ji}(0) = 1$

For each training input

Step2. Activation states of all F2 neurons are set to zero and all F1 (a) neurons are assigned to the input vector S

Step3. Computation of norm of S ($\|S\|) = \sum s_i$

Step4. Sending signals from F1 (a) to F1 (b) layer, $x_i = s_i$

<u>For each F2 node that is not already inhibited under the current reset schedule</u>

Step5. Calculation of net input to that particular F2 node

$$y_j = \sum_i (b_{ij} x_i)$$

Step6. Finding highest y_j among all y_j's.

Step7. Re-computation of x of F1 (b) layer. $x_i = s_i t_{ji}$

Step8. Computation of the norm of vector x=$\|X\| = \sum x_i$

Step9. Test for reset, if $\left(\frac{\|X\|}{\|S\|}\right) < \rho$, the jth node is inhibited. Continue from Step 5.

If $\left(\frac{\|X\|}{\|S\|}\right) > \rho$

Step10. Updation of weights for node j, $b_{ij_{(new)}}$=$\frac{Lx_i}{L-1+\|x\|}$&$t_{ji_{(new)}}$=x_i

Test for stopping criterion:

- No change in top-down or bottom up weights.
- No reset
- Maximum number of epochs exceeded

2.2.5.1.5. Data Treatment & Processing for ART1

Processing the data according to the demand of ART1 network architecture plays an important role in their successful implementation, which is as follows,

- Normalization of data matrix: All the elements in the scaled matrix are lying between 0-1. The linear scaling function for zero to one transforms a variable x_k into $x^*{}_k$ in the following way:

$$x^*{}_{k,j} = \frac{x_{k,j} - \min(x_k)_{for\ all\ j's}}{\max(x_k)_{for\ all\ j's} - \min(x_k)_{for\ all\ j's}} \tag{2.41}$$

Where, k and j are column and row of the data matrix respectively.

- Conversion of scaled data matrix into binary matrix: The elements of the above scaled matrix which are below 0.5 are given an attribute of 0 & elements including 0.5 and above are considered as 1.0 in the binary data matrix.

2.2.5.2. Neural Networks as Functional Approximator

As mentioned in the section 2.5.1, neural networks serve as universal functional approximator. A simple Multi Layer Perceptron model is good enough to understand and adapt the functional relationship between inputs and output data. A detailed architecture and working tactic of MLP is given below.

2.2.5.2.1. MULTI LAYER PERCEPTRON (MLP)

The multilayer perception neural network is built up of simple components. In the beginning, a single input neuron will described which will then be extended to multiple inputs. Next, these neurons will be stacked together to produce layers [28]. Finally, the layers are cascaded together to form the network.

Single-input neuron: A single-input neuron is shown in Fig. 2.8. The scalar input p is multiplied by the scalar weight W to form W_p, one of the terms that is sent to the summer. The other input, 1, is multiplied by a bias b and then passed to the summer. The summer output n often referred to as the net input, goes into a transfer function f which produces the scalar neuron output *a* (sometimes "activation function" is used rather than transfer function and offset rather than bias).

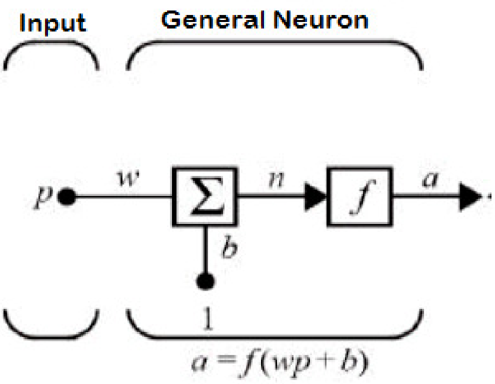

Figure 2.8: Single input neuron

From Fig. 2.8, both *w* and *b* are both adjustable scalar parameters of the neuron. Typically the transfer function is chosen by the designer and then the parameters *w* and *b* will be adjusted by some learning rule so that the neuron input/output relationship meet some specific goal. The transfer function in Fig. 2.8 may be a linear or nonlinear function of *n*. A particular transfer function is chosen to satisfy some specification of the problem that the neuron is attempting to solve. One of the most commonly used functions is the log-sigmoid transfer function, which is shown in Fig. 2.9[28].

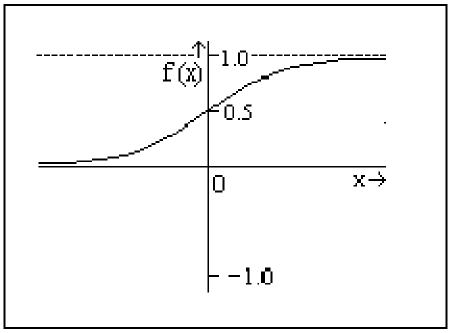

Figure 2.9: Log-sigmoid transfer function

This transfer function takes the input (which may have any value between plus and minus infinity) and squashes the output into the range 0 to 1 according to the expression:

$$a = \frac{1}{1-e^{-1}} \tag{2.42}$$

The log-sigmoid transfer function is commonly used in multi-layer networks that are trained using the back propagation algorithm.

Multiple-input neuron: Typically, a neuron has more than one input. A neuron with R inputs is shown in Fig. 2.10. The individual inputs p_1, p_2,……, p_g are each weighted by corresponding elements $W_{1,1}W_{1,2}$,.... $W_{1,R}$of the weight matrix W.

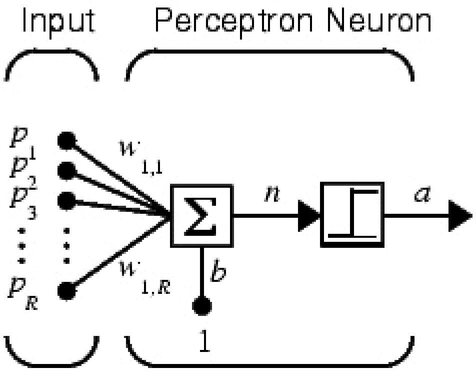

Figure 2.10: Multiple input neuron.

The neuron has a bias b, which is summed with the weight inputs to form the net input n:

$$n = W_{11}p_1 + W_{12}p_2 + \cdots + W_{1R}p_R + b \tag{2.43}$$

This expression can be written in matrix form as:

$$n = Wp + b \tag{2.44}$$

Where, the matrix W for the single neuron case has only one row. Now the neuron output can be written as:

$$a = f(Wp + b) \tag{2.45}$$

A particular convention in assigning the indices of the elements of the weight matrix has been adopted [28].

The first index indicates the particular neuron destination for the weight. The second index indicates the source of the signal fed to the neuron. Thus, the indices in $W_{1,2}$ say that this weight represents the connection to the first (and only) neuron from the second source [28]. A multiple-input neuron using abbreviated notation is shown in Fig. 2.11.

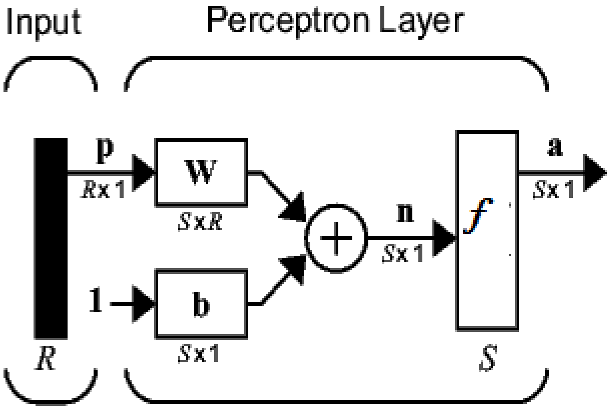

Figure 2.11: Neuron with R inputs, abbreviated notation.

As shown in Fig. 2.11, the input vector p is represented by the solid vertical bar at left. The dimensions of p are displayed below the variable as $R \times 1$, indicating that the input is a single vector of R elements. These inputs go to the weight matrix W, which has R columns but only one row in this single neuron case.

A constant 1 enters the neuron as an input and is multiplied by a scalar bias b. The net input to the transfer function f is n, which is the sum of the bias b and the product W_p. The neuron's output is a scalar in this case. If there exit more than one neuron, the network output would be a vector.

2.2.5.2.1.1 Multilayer Perceptron Network Architectures

Commonly one neuron, even with many inputs, may not be sufficient. We might need five or ten, operating parallel, in what we will call a “layer". This concept of a layer is discussed below.

A layer of neurons: A single-layer network of *S* is shown in Fig. 2.12. Note that each of the *R* inputs is connected to each of the neurons and that the weight matrix now has *s* rows. The layer includes the weight matrix, the summers, the bias vector *b*, the transfer function boxes and the output vector *a*. Each element of the input vector *p* is connected to each neuron through the weight matrix *W*. Each neuron has a bias b_i, a summer, a transfer function *f* and an output a_i. Taken together, the outputs form the output vector *a*. It is common for the number of inputs to a layer to be different from the number of neurons (i.e. $R \neq S$). The input vector elements enter the network through the weight matrix *W*:

$$W = \begin{bmatrix} W_{1,1} & \cdots & W_{1,R} \\ \vdots & \ddots & \vdots \\ W_{S,1} & \cdots & W_{S,R} \end{bmatrix} \tag{2.46}$$

The row indices of the elements of matrix *W* indicate the destination neuron associated with that weight, while the column indices indicate the source of the input for that weight. Thus, the indices in $W_{3,2}$ say that this weight represents the connection to the third neuron from the second source. The S-neuron, R-input, one-layer network also can be drawn in abbreviated notation as shown in Fig. 2.13. Here again, the symbols below the variables tell that for this layer, *p* is a vector of length *R*, *W* is an *S*×*R* matrix and *a* and *b* are vectors of length *S*. As defined previously, the layer includes the weight matrix, the summation and multiplication operations, the bias vector *b*, the transfer function boxes and the output vector.

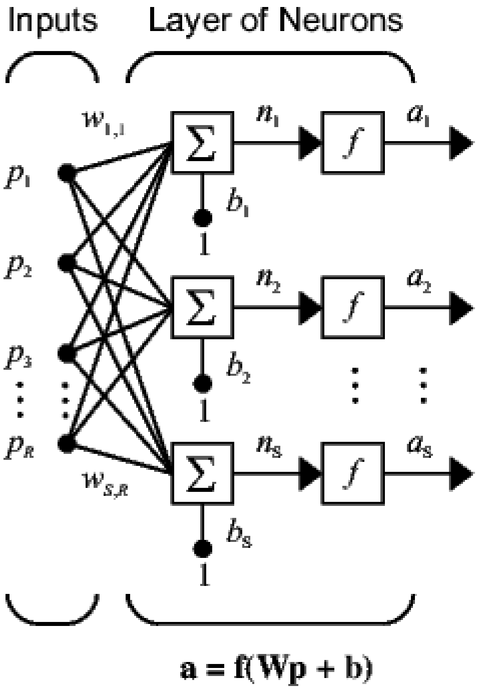

Figure 2.12: Layer of *S* neurons.

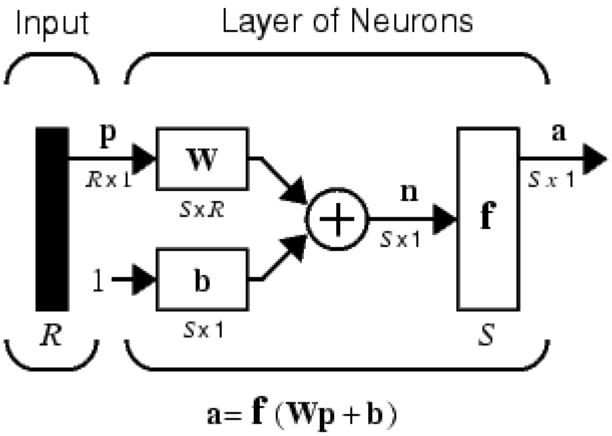

Figure 2.13: Layer of *S* neurons, abbreviated notation.

Multiple layers of neurons: Now consider a network with several layers which has been implemented in this project for the purpose of process identification. In this network each layer has its own weight matrix *W*, its own bias vector *b*, a net input vector *n* and an output vector *a*. Some additional notation should be introduced to distinguish between these layers. Superscripts are used to identify these layers. The number of the layer as a superscript is appended to the names for each of these variables. Thus, the weight matrix for the second layer is written as W^2. This notation is used in the three-layer network shown in Fig. 2.14.

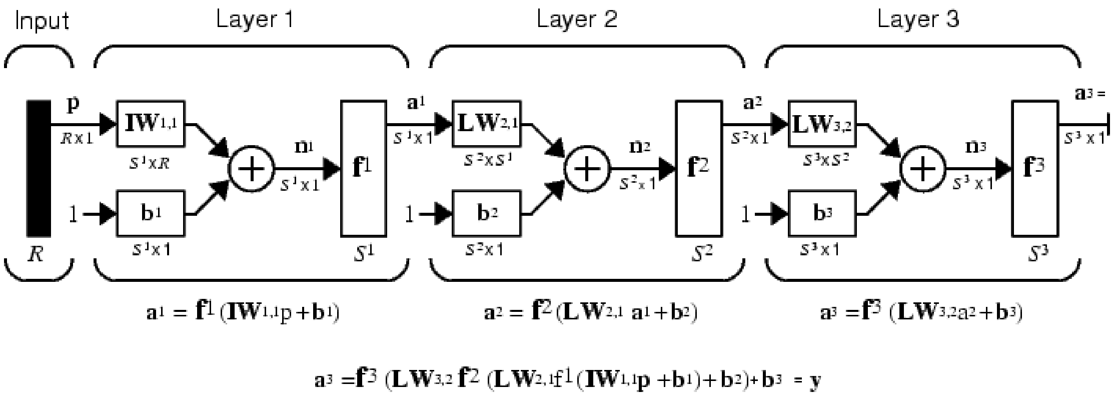

Figure 2.14: Three layer network.

2.2.5.2.1.2 Structure and Operation of Multilayer Perceptron Neural Network (MLP)

MLP neural networks consist of units arranged in layers [29]. Each layer is composed of nodes and in the fully connected network considered here each node connects to every node in subsequent layers. Each MLP is composed of a minimum of three layers consisting of an input layer, one or more hidden layers and an output layer. The input layer distributes the inputs to subsequent layers. Input nodes have liner activation functions and no thresholds. Each hidden unit node and each output node have thresholds associated with them in addition to the weights. The hidden unit nodes have nonlinear activation functions and the outputs have linear activation functions. Hence, each signal feeding into anode in a subsequent layer has the original input multiplied by a weight with a threshold added and then is passed through an activation function that may be linear or nonlinear (hidden units). A typical three layer network is shown in Fig. 2.15. Only three layer MLPs will be considered in this work since these networks have been shown to approximate any continuous function [23, 25, 30]. For the actual three layers MLP, all of the inputs are also connected directly to all of the outputs [29].

The training data consist of a set N_V training patterns (x_p, t_p) where P represents the pattern number. In Fig. 2.15, X_P corresponds to the N-dimensional input vector of the P^{th} training pattern and Y_P corresponds to the M-dimensional output vector from the trained network for the P^{th} pattern. For ease of notation and analysis, threshold on hidden units and output units are handled by assigning the value of one to an augmented vector component denoted by $X_p\ (N+1)$. The output and input units have linear activations. The input to the J^{th} hidden unit, $net\ _P(j)$ is expressed [29] as:

$$net_p(j) = \sum_{k=1}^{N+1} W_{hi}(j,k) X_p(k) \qquad 1 \le j \le N_h \tag{2.47}$$

With the output activation for the P^{th} training pattern, $O_p(j)$, being expressed by:

$$O_p(j) = f(net_p(j)) \tag{2.48}$$

The nonlinear activation is typically chosen to be the sigmoid function

$$f(net_p(j)) = \frac{1}{1 - e^{-net_p(j)}} \tag{2.49}$$

In (2.47) and (2.48), the N input units are represented by the index K and $W_{hi}\ (J,\ K)$ denotes the weight connecting the K^{th} input unit to the J^{th} hidden unit.

The overall performance of the MLP is measured by the mean square error (MSE) expressed by:

$$E = \frac{1}{N} \sum_{p=1}^{Nv} E_p = \frac{1}{N} \sum_{p=1}^{Nv} \sum_{i=1}^{M} [t_p(i) - y_p(i)]^2 \tag{2.50}$$

E_p corresponds to the error for the P^{th} pattern and t_p is the desired output for the P^{th} pattern. This also allows the calculation of the napping error for the i^{th} output unit to be expressed by:

$$E_i = \frac{1}{N_v} \sum_{i=1}^{M} [t_p(i) - y_p(i)]^2 \tag{2.51}$$

with the i^{th} output for the P^{th} training pattern expressed by:

$$Y_p(i) = \sum_{k=1}^{N+1} W_{oi}(i,k) X_p(k) + \sum_{j=1}^{N_h} W_{oi}(i,j).O_p(j) \tag{2.52}$$

In (2.52), W_{oi} *(i,k)* represents the weight from the input nodes to the output nodes and W_{oh} *(i ,j)* represents the weight from the hidden nodes to the output nodes.

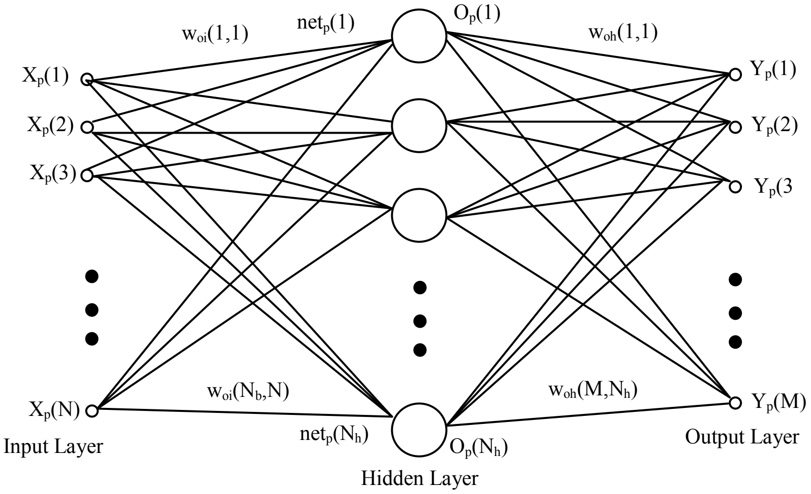

Figure 2.15: Typical three layer multilayer perceptron neural network

2.2.6. Time-Series Identification

The description dynamic input-output models are more appropriate for representing the behavior of processes with a view to process monitoring, fault detection and real time control system design. The linear model structures are discussed in this section. They can handle mild nonlinearities. They can also result from linearization around an operating point. Inputs, outputs, disturbances and state variables are denoted as *u*, *y*, *d* and *x*, respectively. The models can be in continuous time (differential equations) or discrete time (difference equations). For multivariable processes where *$u_i(t)$, $u_z(t)$, ….,u_m (t)* are the *m* inputs, the input vector *u(t)* at time *t* is written as a column vector. Similarly, the *p* outputs, and the *n* state variables are defined by column vectors:

$$y(t) = \begin{pmatrix} y_1(t) \\ \vdots \\ y_p(t) \end{pmatrix}, \qquad u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{pmatrix}, \qquad x(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix}$$

Disturbances $d(t)$, residuals $e(t) = y(t) - \hat{y}(t)$, and random noise attributed to input and output variables are also represented by column vectors with appropriate dimensions in a similar manner.

Time series models can be cast as a regression problem where the regressor variables are the previous values of the same variable and past values of inputs and disturbances.

A general linear discrete time model for a single variable *y(t)* can be written as

$$y(t) = \eta(t) + w(t) \tag{2.53}$$

where $w(t)$ is a disturbance term such as measurement noise and $\eta(t)$ is the noise free output

$$\eta(t) = G(q,\theta)u(t) \tag{2.54}$$

with rational function $G(q,\theta)$ and input $u(t)$. The function $G(q,\theta)$ relates the inputs to noise-free outputs whose values are not known because the measurements of the outputs are corrupted by disturbances such as measurement noise. The parameters of $G(q,\theta)$ are represented by the vector θ, and *q* is called the shift operator. Assume that relevant information for the current value of output *y(i)* is provided by past values of *y(t)* for n_y previous time instances and past values of *u(t)* for n_u previous instances. The relationship between these variables is

$$\eta(t)+f_1\eta(t-1)+...+f_{n_y}\eta(t-n_y)=b_1u(t)+b_2u(t-1)+...+b_{n_u}u(t-(n_u-1)) \tag{2.55}$$

where $f_i, n=1,2,...,n_y$ and $b_i=1,2,...,n_u$ are parameters to be determined from data. Defining the shift operator *q* as

$$y(t-1)=q^{-1}y(t) \tag{2.56}$$

Eq. (2.55) can be written using two polynomials in *q*

$$\eta(t)(1+f_1q^{-1}+\cdots+f_{n_y}q^{-n_y})=u(t)(b_1+b_2q^{-1}+\cdots+b_{n_u}q^{-(n_u-1)}) \tag{2.57}$$

This equation can be written in a compact form by defining the polynomials

$$F(q)=(1+f_1q^{-1}+\cdots+f_{n_y}q^{-n_y}) \tag{2.58}$$

$$B(q)=(b_1+b_2q^{-1}+\cdots+b_{n_u}q^{-(n_u-1)}) \tag{2.59}$$

Where $\eta(t)=G(q,\theta)u(t)$ with $G(q,\theta)=\dfrac{B(q)}{F(q)}$ (2.60)

Often the inputs may have a delayed effect on the output. If there is a delay of n_k sampling times, Eq. (2.55) is modified as

$$\eta(t)+f_1\eta(t-1)+...+f_{n_y}\eta(t-n_y)=b_1u(t-n_k)+b_2u(t-(n_k+1))+...+b_{n_u}u(t-(n_u+n_k-1)) \tag{2.61}$$

The disturbance term can be expressed in the same way

$$w(t)=H(q,\theta)e(t) \tag{2.62}$$

Where, *e(t)* is white noise and

$$H(q,\theta)=\frac{C(q)}{D(d)}=\frac{1+c_1q^{-1}+\cdots+c_{n_c}q^{-n_c}}{1+d_1q^{-1}+\cdots+d_{n_d}q^{-n_d}} \tag{2.63}$$

The model (Eq. 2.53) can be written as

$$y(t)=G(q,\theta)u(t)+H(q,\theta)e(t) \tag{2.64}$$

where, the parameter vector θ contains the coefficients b_i, c_i, d_i and f_i of the transfer functions $G(q,\theta)$ and $H(q,\theta)$. The model structure is described by five parameters n_y, n_u, n_k, n_c, and n_d. Since the model is based on polynomials, its structure is finalized when the parameter values are selected. These parameters and the coefficients are determined by fitting candidate models to data and minimizing some criteria based on reduction of prediction error and parsimony of the model. Auto Regressive model with eXogenous inputs (ARX), Auto Regressive moving average model with eXogenous inputs (ARMAX), Output error model (OE), non-linear version NRMAX, NRMA are the frequently used time series identification models. One of the drawbacks of these models are their limited range of applicability; i.e. the extrapolation capabilities of these models beyond the range for which they are developed become poor. Cross correlation coefficient is a tool that can be used to check whether there is sufficient impact of the process input on process output, i.e. whether two time series data are correlated.

In case of ARX model both the denominators of G and H will become same and the disturbance term C(q) becomes unity leaving the above equation like

$$A(q)y(t) = B(q)u(t) + e(t) \tag{2.64}$$

ARX model was used in the present project for the time series identification of phenol degradation process.

All the machine learning algorithms including PCA, PLS, Clustering, PNN, ART1 and time-series identification are used in the present work for process identification, process quality monitoring and fault detection purpose and they have been discussed in the present chapter with their present state of art and application.

REFERENCES:

1. Chen, J. & Liao, C. 2002. Dynamic process fault monitoring based on neural network and PCA. Journal of Process Control, 12, 277–289.
2. Zhao, C., Wang, F., Lu, N. & Jia, M. 2007. Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes. Journal of Process Control, 17, 728–741.
3. De Gaetano, A., Panunzi, S., Rinaldi, F. & Sciandrone, M. 2009. A patient adaptable ECG beat classifier based on neural networks. Applied Mathematics and Computation, 213, 243-249.
4. Meleiro, L. C., Zuben, F. V. & Filho, R.M. 2009. Constructive learning neural network applied to identification and control of a fuel-ethanol fermentation process. Engineering Applications of Artificial Intelligence, 22, 201–215.
5. Sadrzadeh, M., Mohammadi, T., Ivakpour, J. & Kasiri, N. 2009.Neural network modeling of Pb2+ removal from wastewater using electrodialysis. Chemical Engineering and Processing, 48, 1371–1381.
6. Jyh-Cheng Jeng, 2010. Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms. Journal of the Taiwan Institute of Chemical Engineers, 41, 475–481.
7. Marchitana, N., Cojocarub, C., Mereutaa, A., Ducac, Gh., Cretescud, I. & Gontaa, M. 2010. Modeling and optimization of tartaric acid reactive extraction from aqueous solutions: A comparison between response surface methodology and artificial neural network. Separation and Purification Technology, 75, 273–285.

8. Bin Shams, M.A., Budman, H. M. & Duever, T. A. 2011. Fault detection, identification and diagnosis using CUSUM based PCA. Chemical Engineering Science, 66, 4488-4498.
9. Pendashteh, A. R., Fakhru'l-Razi, A., Chaibakhsh, N., Abdullah, L.C., Sayed, S.M. & Abidin, Z.Z. 2011. Modeling of membrane bioreactor treating hypersaline oily wastewater by artificial neural network. Journal of Hazardous Materials, 192, 568– 575.
10. Chang, P., & Lai, C. 2005. A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting. Expert Systems with Applications, 29(1), 183–192.
11. Hsieh, N. 2005. Hybrid mining approach in the design of credit scoring models. Expert Systems with Applications, 28(4), 655–665.
12. Kuo, R., Kuo, Y., & Chen, K. 2005. Developing a diagnostic system through integration of fuzzy case-based reasoning and fuzzy ant colony system. Expert Systems with Applications, 28(4), 783–797.
13. Shin, H., & Sohn, S. 2004.Segmentation of stock trading customers according to potential value. Expert Systems with Applications, 27(1), 27–33.
14. Tsai, C., Chiu, C., & Chen, J. 2005. A case-based reasoning system for PCB defect prediction. Expert Systems with Applications, 28(4), 813–822.
15. Raich, A., & Cinar, A. 1996. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. AIChE Journal, 42(4), 995-1009.
16. Wold, H. 1966. Estimation of principal components and related models by iterative least squares, In: Krishnaiah, P.R., Ed., Multi Variate Analysis II. Academic Press: New York, 391-420.

17. Geladi, P., & Kowalski, B.R. 1986. Partial least-squares regression: A tutorial. Anal. Chim. Acta., 185, 1-17.

18. Qin, S.J., & McAvoy, T.J. 1992. Nonlinear PLS modeling using neural network. Comput. Chem. Eng., 16 (4), 379-391.

19. Holcomb, T.R., &Morari, M. 1992. PLS/neural networks. Comput. Chem. Eng., 16(4), 393-411.

20. Malthouse, E.C., Tamhane, A.C., & Mah, R.S.H. 1997.Nonlinear partial least squares. Comput. Chem. Eng., 21(8), 875-890.

21. Zhao, S.J., Zhang, J., Xu, Y.M., & Xiong, Z.H. 2006. Nonlinear projection to latent structures method and its applications. Ind. Eng. Chem. Res., 45, 3843-3852.

22. Lee, D.S., Lee, M.W., Woo, S.H., Kim, Y. &Park, J.M. 2006. Nonlinear dynamic partial least squares modeling of a full-scale biological wastewater treatment plant. Process Biochemistry, 41, 2050-2057.

23. Cybenko, G. 1989. Approximation by superposition of a sigmoidal function, Math. Contr. Signals Syst., 2, 303–314.

24. Hornik, K. 1991. Approximation capabilities of multilayer feed forward networks. Neural Networks, 4, 251–257.

25. Hornik, K., Stinchcombe, M., & White, H. 1989. Multilayer feed forward networks are universal approximators. Neural Networks, 2, 359–366.

26. Richard, M.D., & Lippmann, R. 1991. Neural network classifiers estimate Bayesian a posteriori probabilities. Neural Comput., 3, 461–483.

27. Specht, D.F. 1990. Probabilistic Neural Networks. Neural Networks, 3, 109-118.

28. Haykin, S. 1999. Neural Networks: A Comprehensive Foundation. 2nd Edn., New Jersey: Prentice-Hall.

29. Walter, H.D., & Michael T.M. 2005. Recent Developments in Multilayer Perceptron Neural Networks. Proceedings of the 7th Annual Memphis Area Engineering and Science Conference, MAESC 2005.

30. Hornik, K., Stinchombe, M., & White, H. 1990. Universal Approximation of an unknown Mapping and its Derivatives Using Multilayer Feed forward Networks. Neural Networks, 3(5), 551-560.

CHAPTER3- PROCESS IDENTIFICATION

The current chapter presents the identification of phenol degradation process. The organic pollutant phenol has been degraded by bacteria named *Pseudomonas putida* (ATCC: 11172). Four parameters namely temperature, pH, RPM and phenol dosage were varied systematically to produce 16 sets of useful data. Out of sixteen runs, three runs namely 4th, 6th and 16th runs were used to produce time series data which in turn was used for the identification of dynamics. The process dynamics was identified by using (ARX) and Artificial Neural Networks (ANN). A PLS regression model which can estimate the effect of each parameter on the phenol degradation process has been developed.

3.1. PHENOL AS AN ORGANIC POLLUTANT AND ITS REMOVAL

Phenol is a commonly found pollutant in industrial waste effluents, like from the factories of iron-steel, coke petroleum, pesticide, paint solvent, pharmaceutical, wood processing chemicals, and pulp and paper [1-4]. Physicochemical processes (adsorption using activated carbon, for example) and biological processes are often used for removal of phenol and its derivatives from wastewater. The drawbacks of adsorption processes include the needs of adsorbent regeneration, the compensation to adsorbent loss, and the possibility of inducing secondary pollution [5-9]. On the other hand, activated sludge is capable of removing both heavy metals and variety of organic compounds in wastewater stream at a relatively low processing cost [10-15]. Activated sludge is basically a biomass that contains mainly bacteria and protozoa. The cell wall of the bacteria primarily consists of various organic compounds including chitin, acidic polysaccharides, lipids amino acids and other cellular compounds that could adsorb both the heavy metals and various organics [16, 17]. The protozoa are unicellular, motile, relatively large eukaryotic cells, which could absorb organic compounds and lipids [1, 17]. Moreover, the *P. putida* had been identified as an effective strain for biodegradation of chlorophenol, resorcinol, and related compounds [18, 19].

3.2. IDENTIFICATION OF DYNAMICS FOR PHENOL BIODEGRADATION

Bioprocess technology is currently employed for the production of several commodity and fine chemicals. Because of the complex nature of microorganism growth and product formation or degradation of substrate in batch and fed-batch cultures, which are often used in preference to continuous cultures, the control of bioprocesses continues to pose a challenge to chemical engineers. Extensive developments in the area of bioprocess have begun, but much work remains to be done to couple model-based control methods to biochemical reactor technology.

In a first principle approach, Haldane equation [20] has been frequently used to describe the degradation of phenol in pure or mixed cultures [21-26], which will be written as

$$\mu = \frac{\mu_m S}{K_S + S + S^2 / K_i} \tag{3.1}$$

Where μ is the specific growth rate, S is the phenol concentration and K_s, K_i& μ_m are constants.

Usually parameters derived from batch experiments are used in Equation (3.1) to predict the response. Haldane equation predicts a global maximum specific growth rate (μ^*) at a phenol concentration S^* and the specific growth rate asymptotically approaches zero as the substrate (phenol) concentration increases. Equation (3.1) is also used to describe the specific phenol uptake rate (q_p) of washed cells of *Pseudomonas putida* in batch systems [27-29]. Notwithstanding its popularity, questions remain as to the adequacy of Equation (3.1) as a model of phenol degradation.

The basis of Haldane equation is a hypothetical enzyme-substrate interaction:

$$E + S \overset{K_1}{\Leftrightarrow} ES$$

$$ES + S \overset{K_2}{\Leftrightarrow} ESS$$

$$ES \overset{k}{\Leftrightarrow} E + P$$

Where, the constants K_1 and K_2 are equivalent to K_S and K_i, respectively of Equation (3.1). Similarly, the rate constant k is equivalent to μ_m/E_0, where E_0 is the total concentration of enzyme catalyzing the slowest reaction in the pathway for the consumption of substrate. Unfortunately, at a fundamental level, there is no mechanistic proof for the above scheme with respect to degradation of phenol [30]. Moreover, from a more practical perspective, prediction discrepancies exist in the literature. Although betterment of these models have been done in recent years [31, 32], the complexity is still there because of lack of knowledge in the metabolic path way of the phenol uptake and its utilization. In first principle approach if one wants to perfectly model the phenol degradation process, one has to go through the chemical mass and energy balances and should be aware of the rate limiting steps that control the degradation process of phenol, which clearly requires the knowledge of the reactions that are taking place inside the organism (nothing but metabolism). But in case of the prediction of phenol degradation using data based modeling, the perfection is achieved by improving the quality and the amount of the input output data. Thus the complexity of knowing and analyzing all the metabolic pathways and their reactions required for the development of the process model is reduced.

In this perspective, the present work attempts to propose an alternative to rate model for phenol degradation process, the data based modeling techniques like ANN, and ARX. PLS technique was used to develop an empirical model relating phenol degradation process with the variables like temperature, pH, RPM and Phenol loading at steady state.

In order to predict phenol degradation as a function of five inputs namely temperature, pH, RPM, Phenol Loading and time a Multi-Layer Perceptron (MLP) network was trained selecting suitable numbers of input and hidden neurons, adjusting weights, selecting proper training algorithms, varying number of iterations, selecting proper learning rate etc. While using ARX for prediction of phenol degradation, the order of the model had to be selected and established. The time series data used in MLP network and for ARX prediction were tested apriori for their auto correlation and partial correlation coefficient, which could reveal the non-stationarity in the time series data, if any. It's not the question of only using ARX and ANN, a thorough data preprocessing is a part of that activity. The ensemble of aforesaid activities qualifies ANN and ARX as data based models/black-box models and in the present study instead of first-principle models of the process of interest.

3.3. BIO-DEGRADATION OF PHENOL

3.3.1 STRAIN

The strain being used for phenol degradation was a heterotrophic bacterium named *Pseudomonas putida* (ATCC: 11172). The strain was obtained from National Collection of Industrial Microorganisms (NCIM), Pune. The strain was supplied in the form of dry spores. The spores were cultured in the media specified by NCIL for the preparation of inoculum. The strain was sub cultured every two weeks for maintenance.

3.3.2 LABORATORY SCALE BENCH REACTOR

A IIC-LABEAST bench-top bio-fermentor of 2.5 litter capacity has been used for the degradation purpose (Fig. 3.1). The reactor was connected to a high and low temperature thermostatic water bath to maintain constant temperature. An aeration of 1LPM was maintained for the supply of oxygen in the reactor. The fermentor was fully equipped with a pH electrode, stirrer and baffles.

3.3.3 MEDIA

The media composition was decided in such way that it supplies all the necessary nutrients for the growth of *P.putida.* The constituents are of 'basic mineral salt medium' which is widely used for the growth of microorganisms. In this context, the following media composition was used for the phenol degradation [33].

K_2HPO_4	1.5g/L
KH_2PO_4	0.5 g/L

$(NH_4)_2SO_4$	0.5 g/L
NaCl	0.5 g/L
Na_2SO_4	3.0 g/L
Yeast Extract	2 g/L
Glucose	0.5 g/L
$FeSO_4$	0.002 g/L
$CaCl_2$	0.002 g/L

The experiments were designed systematically by varying the variables namely Temperature, pH, RPM and Phenol Loading at four different levels each. A total of 16 combinations were made which can effectively give the picture of effects imposed by each parameter on the process. All the experiments were run for an incubation period of 36 hrs. Table 3.1 shows the different combinations containing four parameters at four different levels each and their corresponding phenol degradation percentage.

3.3.4 CHROMATOGRAPHIC ANALYSIS OF PHENOL

The concentration of Phenol remained in the broth solution was determined using High Performance Liquid Chromatography (HPLC). “Jasco PU-2080Plus” high performance liquid chromatography was used for the analysis. The mobile phase used for the determination of phenol is a mixture of Isopropanol, Acetic Acid and water in a ratio 20:1:79 respectively. For this mobile phase and at a flow rate of 0.6 mL/min the retention time was found to be 4.5 min. Figure 3.2shows the chromatogram which gives a peak for phenol at a retention time of 4.5 min.

3.4. IDENTIFICATION OF PROCESS DYNAMICS USING ANN AND ARX

In order to predict phenol degradation, Multi-Layer Perceptron (MLP) network and ARX were used. The time series data used in MLP network and for ARX prediction were tested apriori for their non-stationarity, if any.

3.4.1 ARTIFICIAL NEURAL NETWORK:

The inputs to the ANN model are process measurements. The measurements are weighted individually or in groups and then combined using a nonlinear 'activation function' at a node referred to as a neuron, named after the nodes in the brain which combine information sent from the natural senses. A process ANN model is often composed of an input layer, an output layer, and one or more hidden layers of nodes. The traditionally used format of ANN is the feed forward neural network (FFN). In this network, the outputs of one layer of nodes serve as the inputs to the following nodes. Given a set of process measurements, the outputs of the neural network can be estimated parameter values or process variables. The weights applied to the inputs in the model are determined through the training processes. To train the ANN, complete process information, corresponding to the neural network inputs and outputs, were required and obtained from the sets of fermentation runs. The set of process input and output measurements spanned by the experimental data is termed the 'experimental space' and the ANN could predict outputs accurately within this range. Here the artificial neural network acted as a function approximator.

The efficiency of a neural network mainly depends upon the data that were used for the training. The training data were self-sufficient in explaining the all the aspects that were considered for the modelling of the process. To develop the rate model of phenol degradation process; three out of sixteen runs (4th, 6th and 16th) were used for the production of time series data/historical database with a time interval of 15 minutes.

Taking the time series reading for every 15 minutes for 36 hrs results in (36×4=144) samples for one run. Analyzing 144 samples using HPLC for estimating the amount of phenol degraded is a big task. Even though it is enough to have one run to estimate the effect of time on the degradation process, three runs were chosen which included the max variations in the input variables, hence, enhanced the accuracy in estimating the dynamics of the process.

The data was augmented before training to make it sufficient enough to represent the effect of each parameter on phenol degradation process. The data were randomized before training to remove any bias in the weight decay. The neural network model used for the prediction was Multi-Layer Perceptron (MLP) model. The number of neurons in the hidden layer varied between 2 and 20 and best of five networks were chosen for the prediction purpose. There were a total of five inputs (i.e. Temperature, pH, RPM, Phenol Loading and Time) in the input layers and one output in the output layer (i.e. % Phenol Degradation). A randomly selected 70% of the data were used for the training, 15% were used for the testing purpose and 15% were used for the validation purpose. The performance of the best five network combinations are presented in Table 3.2.

All the networks were trained by using BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm and the error function used was SOS (Sum of Squares) function [34]. Different activation functions were used for both hidden and output layers and best five combinations were represented in Table 3.2. Figures 3.3and 3.4 depict the training and prediction performances for best five developed MLPs. The representation of MLP is in the form of ‘MLP a-b-c’, where ‘a’ is the number of units in the input layer, ‘b’ is the number of units in the hidden layer and ‘c’ represents the number of units in the output layer. For example ‘MLP 8-12-1’ represents that there are 8 units in the input layer, 12 units in the hidden layer and one unit in the output layer. MLP includes a bias unit for each and every input variable that has been given to it.

In the present ANN model, there are five input variables. The network should have 10 input units including five variables and five bias units. But there were only 8 units in the input layer because, while training the networks; the variable ‘Phenol Loading’ was defined as categorical variable as it accepted only four distinct values (namely 100, 200, 300 and 400 ppm) and did not vary with the time. Thus the network did not consider ‘Phenol Loading’ as an active input unit and a bias unit was not assigned against it. Hence the architecture had only 8 input units and one output unit.

3.4.2 AUTO REGRESSION MODELS WITH EXOGENOUS (ARX) INPUTS

The time series identification using ARX model was done by using the time series data produced in 4th run. ARX model itself considers only one data channel (run) for the identification of dynamics in the process. So only one run (4th) was considered there, either of the 6th& 16th runs could have been used.

In ARX model the disturbance term will be neglected. An Auto Regression models with eXogenous (ARX) inputs can be written as:

$$A(q)y(t) = B(q)u(t) + e(t) \tag{3.2}$$

Where,

A(q)- Transfer function developed from the output data in correspondence to the given input data

B(q)– Transfer function developed from the input data.

y(t) – Output i.e. phenol degradation percentage.

u(t) – Input continuous variable i.e. Temperature/RPM/pH

e(t) – Error

q– Shift operator given by *y(t)/y(t-1)*

Both *A(q)* and *B(q)* are the polynomials having their corresponding coefficients ($a_1,a_2,a_3,\ldots,a_n$ and $b_1,b_2,b_3,\ldots,b_n$) which are determined from the time series data. Three numbers of SISO transfer functions were developed by considering each of the three continuous variables (temperature, pH, and RPM) as inputs with percentage phenol degradation as output. The coefficients for each ARX model were given Table 3.3. The time series identification was done by using the 'Identification Toolbox' in MATLAB. Figure 3.5 shows the measured and simulated outputs and the fit of the model developed by correlating temperature with the output (phenol degradation %). The percentage fit values for the three models developed by correlating temperature, RPM and pH are 90.22%, 91.51% and 85.22% respectively. Thus the ARX models were able to identify the dynamics of the phenol biodegradation process reasonably well.

In fine, it can be acclaimed that it is not the question of only using ARX and ANN, a thorough data preprocessing is a part of that activity. The aforesaid ensemble of activities justifies the inclusion of ARX and ANN as data based models.

3.5. Partial Least Squares (PLS) Regression

Partial least squares technique was used as a function approximator or regressor. PLS attempts to find latent variables that capture the maximum variance in the data at the same time achieve maximum correlation between predictor and predicted variables. PLS regression is a recent technique that generalizes and combines features from principal component analysis and multiple regressions. This prediction is achieved by extracting from the predictors a set of orthogonal factors called latent variables which have the best predictive power.

The inputs and outputs of 16 numbers of runs were used for the PLS regression purpose. The inputs were temperature, pH, RPM and phenol dosage, whereas the output was phenol degradation percentage. It is to be mentioned that the database taken for developing the PLS model was a steady state one. Hence the regression model can only predict the final phenol degradation after the incubation period of 36 hrs for the chosen and interpolated combinations of input variables.

The PLS regression of 16 different input combinations of four variables and their corresponding phenol degradation percentage as output resulted in a regression coefficient of 0.9669 which eventually was a good fit. The prediction versus actual output using PLS model is presented in Figure 3.6. The sixteen combinations of the inputs (temperature, pH, RPM and phenol dosage) for phenol degradation process and their corresponding PLS predicted outputs revealed acquiescent resemblance with the experimental results.

In fine, it can be concluded that ANN and ARX based Phenol degradation dynamics emancipated encouraging results. PLS based empirical model developed can be helpful in designing the process with a view to sizing the equipment and utility requirements.

TABLES:

Table 3.1: Different combinations of input parameters and their corresponding phenol degradation percentages as output.

Run	Temperature	RPM	pH	Phenol Loading	% Degradation
1	34.0	210	6.0	100	62.22
2	34.0	150	7.0	300	54.76
3	28.0	150	6.0	400	49.22
4	28.0	210	7.0	200	59.68
5	31.0	240	6.0	300	53.72
6	25.0	150	5.5	100	61.34
7	28.0	240	6.5	100	62.7
8	25.0	240	7.0	400	51.26
9	31.0	150	6.5	200	56.42
10	25.0	210	6.5	300	54.46
11	25.0	180	6.0	200	56.08
12	31.0	180	7.0	100	64.56
13	34.0	240	5.5	200	55.2
14	28.0	180	5.5	300	51.8
15	31.0	210	5.5	400	48.86
16	34.0	180	6.5	400	51.54

Table 3.2: Summary of the network performances in ANN identified phenol degradation process.

Index	Net. name	Training perf.	Test perf.	Validation perf.	Training error	Test error	Validation error	Training algorithm	Error function	Hidden activation	Output activation
1	MLP 8-12-1	0.999826	0.999809	0.999793	0.077928	0.089405	0.087374	BFGS 227	SOS	Logistic	Logistic
2	MLP 8-16-1	0.999772	0.999845	0.999781	0.102242	0.073340	0.100799	BFGS 124	SOS	Tanh	Exponential
3	MLP 8-20-1	0.999792	0.999885	0.999812	0.093302	0.054573	0.082554	BFGS 344	SOS	Logistic	Logistic
4	MLP 8-12-1	0.999811	0.999781	0.999782	0.084205	0.104174	0.092511	BFGS 282	SOS	Tanh	Tanh
5	MLP 8-20-1	0.999837	0.999875	0.999812	0.072456	0.060513	0.081934	BFGS 315	SOS	Logistic	Identity

Table 3.3: Coefficients of three SISO transfer functions for three ARX models developed.

Variable considered for ARX→		Temperature	RPM	pH
q^1	a_1	- 0.8985	- 0.8985	- 0.8931
	b_1	-4.247×10^{-15}	--	--
q^2	a_2	- 0.1227	- 0.109	- 0.1227
	b_2	$- 4.247\times10^{-15}$	--	--
q^3	a_3	- 0.03465	- 0.02249	- 0.03465
	b_3	$- 4.247\times10^{-15}$	--	--
q^4	a_4	- 0.1638	- 0.1856	- 0.1638
	b_4	$- 4.247\times10^{-15}$	--	--
q^5	a_5	- 0.08709	- 0.0995	- 0.08709
	b_5	$- 4.247\times10^{-15}$	--	--
q^6	a_6	- 0.02045	- 0.01641	- 0.02045
	b_6	$- 4.247\times10^{-15}$	-0.009252	--
q^7	a_7	0.2004	0.2055	0.2004
	b_7	--	--	--
q^8	a_8	0.03953	0.04895	0.03953
	b_8	--	--	--
q^9	a_9	0.08341	0.07851	0.08341
	b_9	--	--	--

FIGURES:

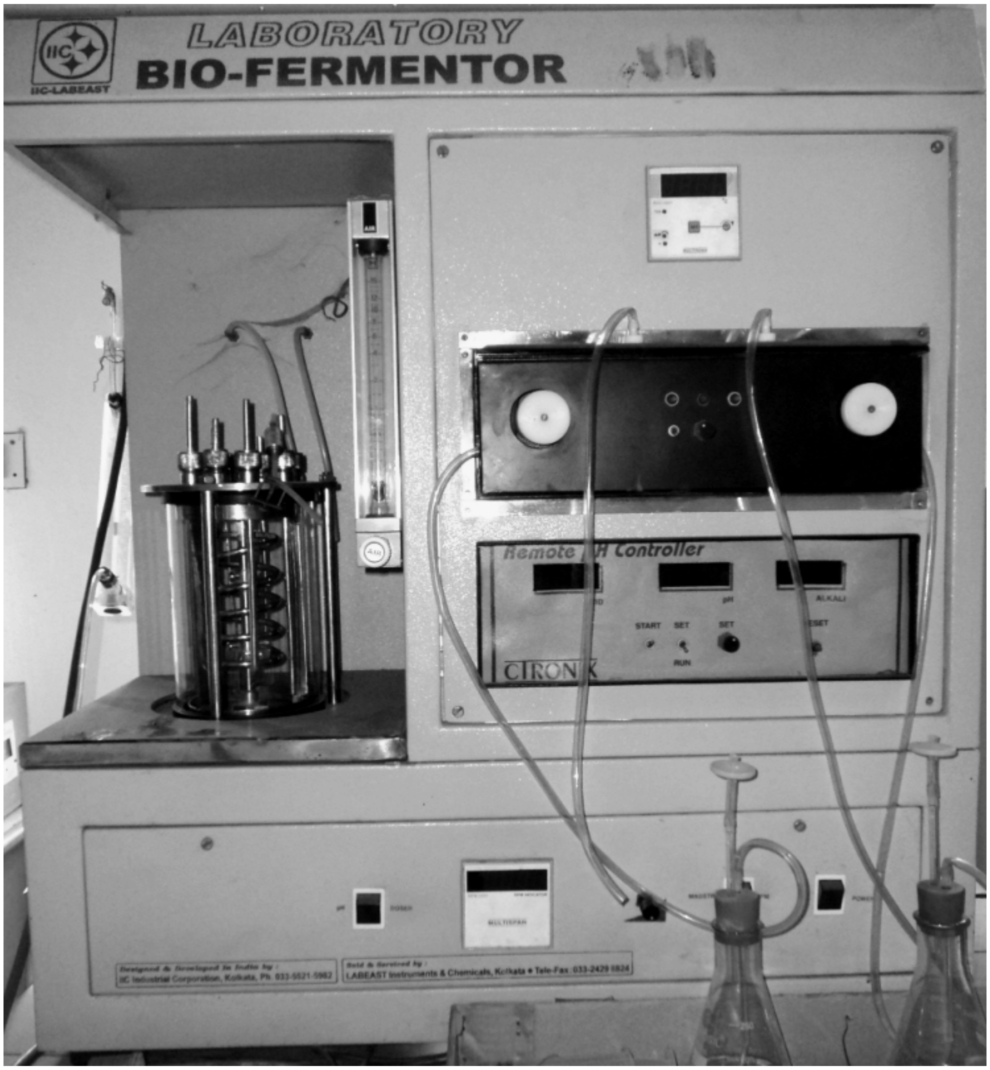

Figure 3.1: Laboratory scale Bench-top Bio-Fermentor.

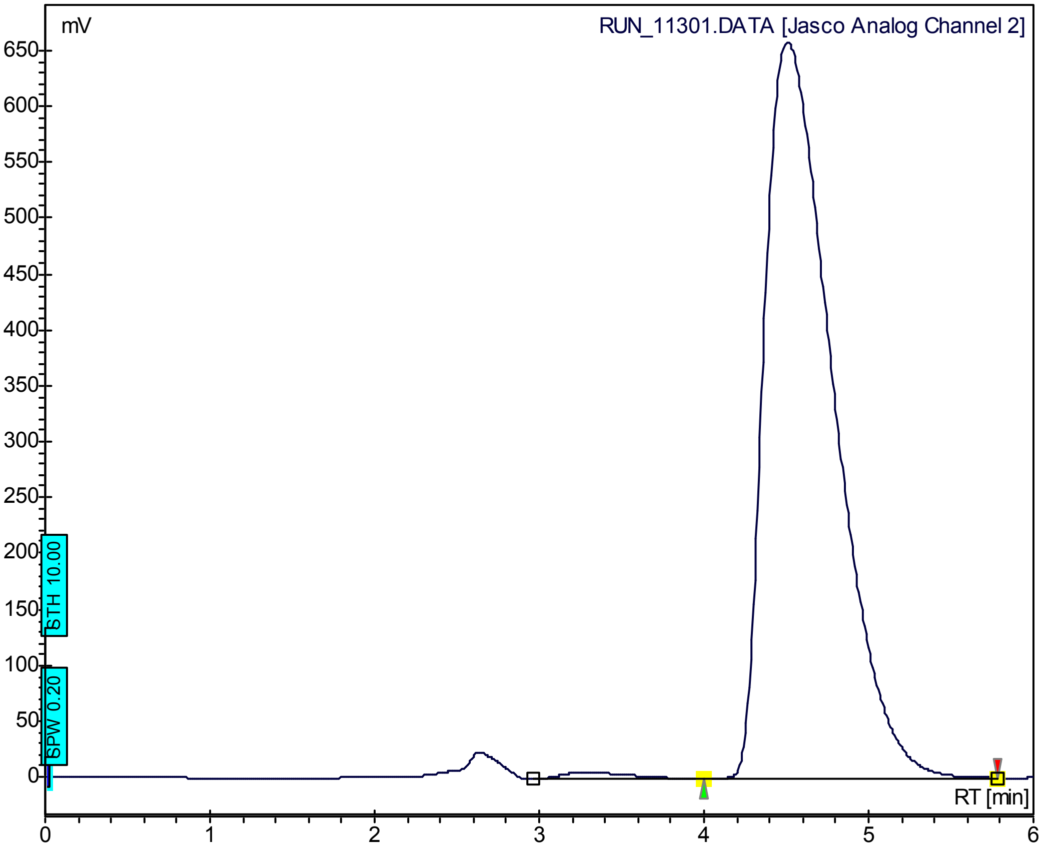

Figure 3.2: A sample chromatogram of phenol showing the peak at RT 4.5min.

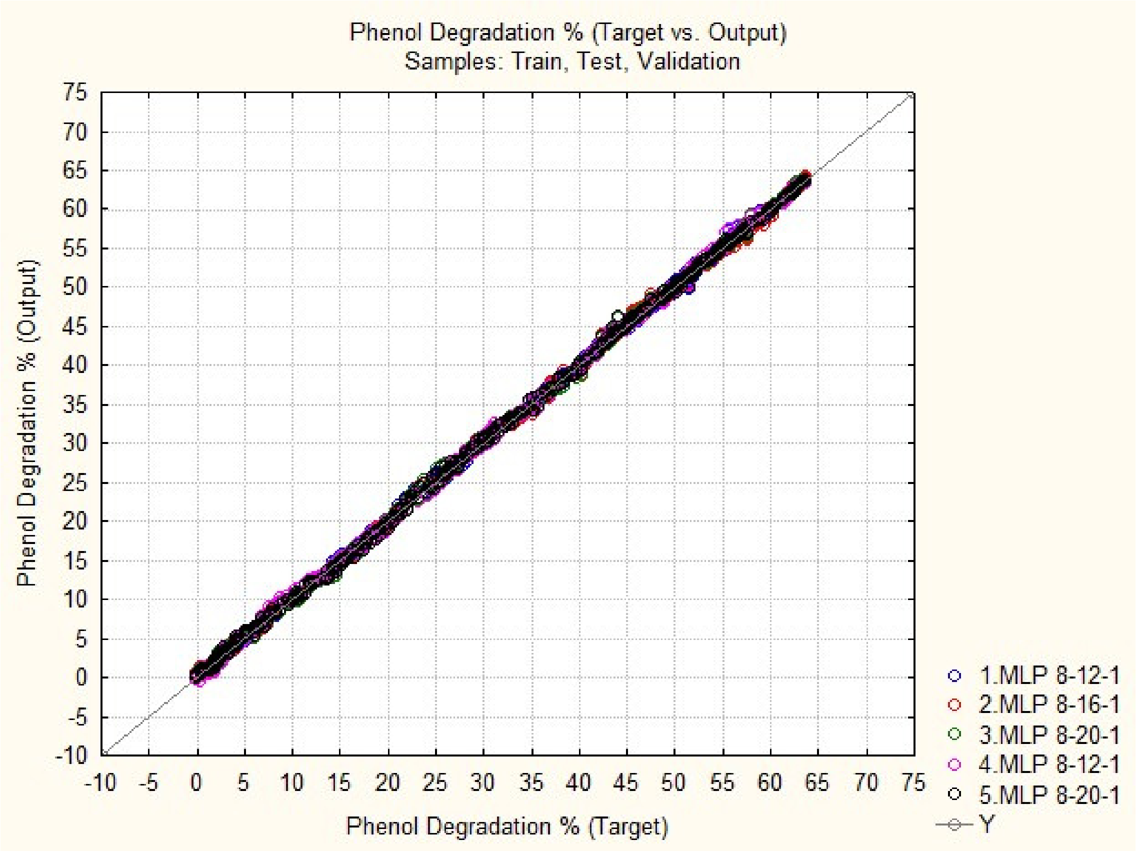

Figure 3.3: Performance of networks in ANN based dynamic model for Phenol degradation (Training, validation and testing performance combined)

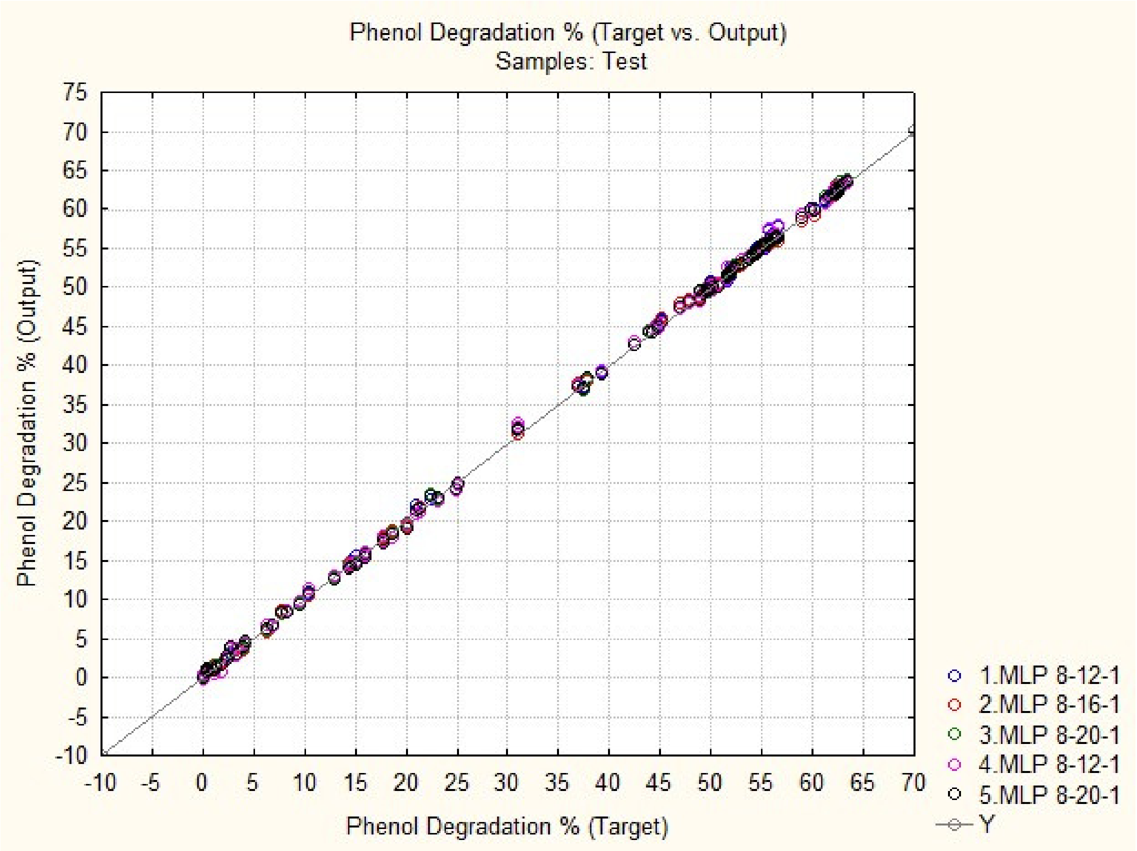

Figure 3.4: Prediction performance of the developed networks in ANN based dynamic model for Phenol degradation

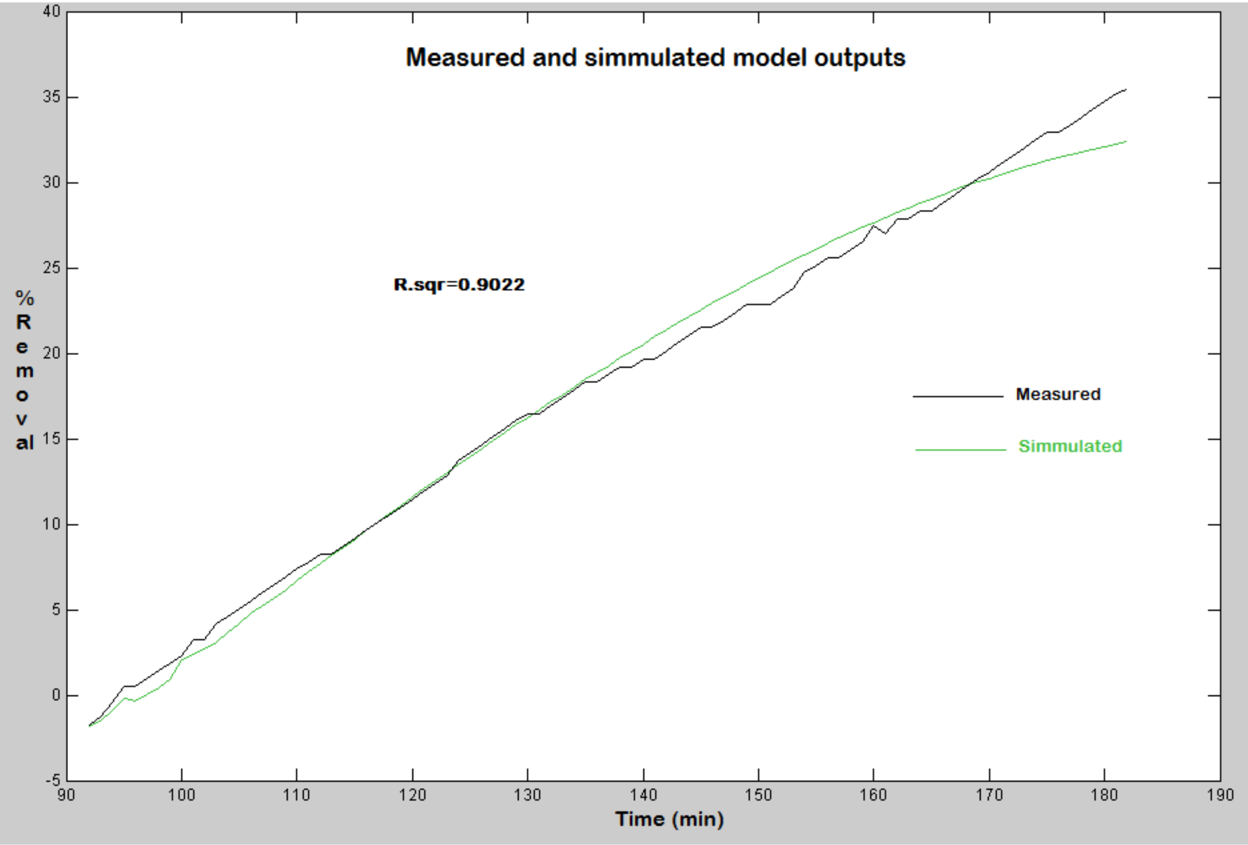

Figure 3.5: Measured and simulated outputs and the fit of the ARX model developed by Identification Toolbox in MATLAB.

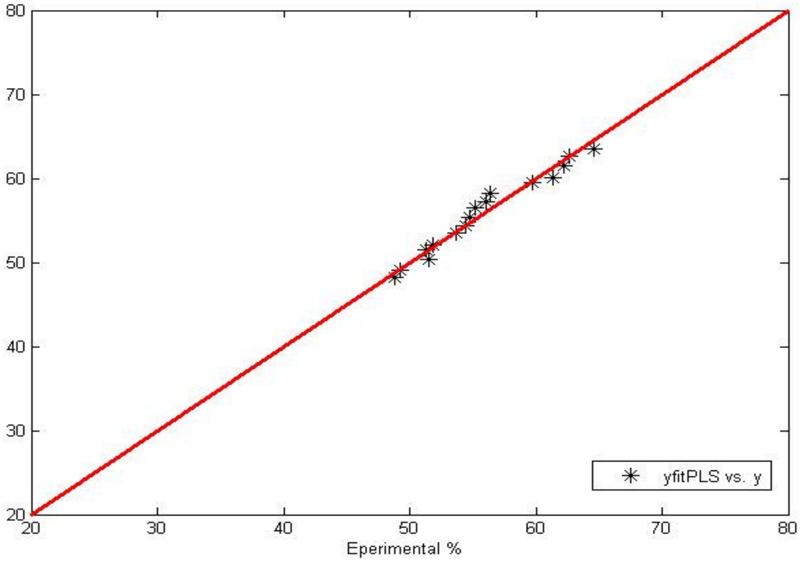

Figure 3.6: Predicted versus actual process outputs using PLS model.

REFERENCES:

1. Aksu, S. & Yener, J., 1998. Investigation of biosorption of phenol and monochlorinated phenols on the dried activated sludge. Process Biochem. 33, 649–655.

2. Patterson, J.N. 1997. Waste Water Treatment Technology. Ann Arbor Science, New York.

3. Bulbul, G. & Aksu, Z., 1997. Investigation of wastewater treatment containing phenol using free and Ca-alginated gel immobilized Pseudomonas putida in a batch stirred reactor. Turkish J. Eng. Environ. Sci., 21, 175–181.

4. Sung, R.H., Soydoa, V. & Hiroaki, O. 2000. Biodegradation by mixed microorganism of granular activated carbon loaded with a mixture of phenols. Biotechnol. Letters, 22, 1093–1096.

5. Perrich, J.R. 1981. Activated Carbon. Adsorption for Wastewater Treatment. CRC Press, Boca Raton, Florida.

6. Brasquet, C., Rouss, J., Subrenat, E. & Le Cloriec, P. 1996. Adsorption and selectivity of activated carbon fibers application to organics. Environ. Technol., 17, 1245–1252.

7. Kumar, S., Upadhyay, S.N. & Upadhya, Y.D. 1987. Removal of phenols by adsorption of fly ash. J. Chem. Technol. Biotechnol., 37, 281– 292.

8. Singh, B.K. & Rawas, N.S. 1994. Comparative sorption equilibrium studies of toxic phenols on fly ash and impregnated fly ash. J. Chem. Technol. Biotechnol., 61, 307–317.

9. Munaf, E., Zkin, R., Kurniad, R. & Kurniadi, I. 1993. The use of rice husk for removal of phenol from waste water as studied using 4- aminoantipyrine spectrophotometric method. Environ. Technol., 18, 355–358.

10. Tsezos, M. & Bell, J.P. 1989. Comparison of biosorption and desorption of hazardous organic pollutants by live and dead biomass. Water Res., 23, 563–568.

11. Chitra, S. & Chanrakasan, G. 1996. Response of phenol degrading Pseudomonas pictorium to changing loads of phenolic compounds. J. Environ. Sci. Health, A31, 599–619.

12. Vaker, D., Connell, C.H., Wells & W.W. 1967. Phosphate removal through municipality waste water treatment at San Antonjo Texas. J. Water Pull. Cont. Fed., 39, 750–771.

13. Zumriye, A., Derya, A., Elif, R. & Burcin, K. 1999. Simultaneous biosorption of phenol and nickel from binary mixtures onto dried aerobic activated sludge. Process Biochem., 35, 301–308.

14. Bux, F., Akkinson, B. & Kasan, K. 1999. Zinc biosorption by waste activated and digested sludges. Water Sci. Technol., 39 (10–11), 127– 130.

15. Ulku, Y., Goksel, N.D. & Celal, F.G. 1999. Effect of chromium (VI) on the biomass yield of activated sludge. Enzyme Microbiol. Technol., 25, 48–54.

16. Takahiro, K. & Eiichi, M. 1995. Survial of a non-flocculating bacterium Thiobacillus thioparus TK-1 inoculated to activated sludge. Water Res., 29, 2751–2754.

17. Brandt, C., Zeng, A. & Deckwer, W. 1997. Adsorption and desorption of pentachlorophenol on cells of M. chlorophenolicum Pep-I. Biotechnol. Bioeng., 55, 480–489.

18. Shular, M.C. & Karge, F. 1992. Bioprocess Engineering. Basic Concepts, Prentice Hall, New Jersey.

19. Clessceri, C.S., Greenberg, A.B. & Trussel, R.R. 1985. Standard methods for the determination of water and wastewater. APHA, Washington, DC 5.48–5.53.

20. Haldane, J.B.S. 1930. Enzymes. Longmans, London.

21. Andrews, J.F. 1968. A mathematical model for the continuous culture of micro organisms utilizing inhibitory substrates. Biotehnol. Bioeng., 10, 707-723.

22. D' Adamo, P.D., Rozich, A.F., & Gaudy Jr., A.F. 1984. Analysis of growth data with inhibitory substrate. Biotechnol. Bioeng., 26, 397-402.

23. Edwards, V.h., Ko, C.R., & Balogh, A. 1982. Dynamics and control of continuous microbial propagators to subject substrate inhibition. Biotechnol. Bioeng., 14, 939-974.

24. Pawlowsky, U., & Howell, J.A. 1973. Mixed culture biooxidation of phenol: I. Determination of kinetic parameters. Biotechnol. Bioeng., 28, 965-971.

25. Sokol, W. 1988. Dynamics of continuous stirred-tank biochemical reactor utilizing inhibitory substrate. Biotechnol. Bioeng., 31, 198-202.

26. Yang, R.D., & Humphrey, A.E. 1975. Dynamic and steady state studies of phenol biodegradation in pure and mixed cultures. Biotechnol. Bioeng., 17, 1211-1235.

27. SoKol, W. 1987. Oxidation of an inhibitory substrate by washed cells (oxidation of phenol by *Pseudomonas Putida*). Biotechnol. Bioeng., 30, 921-927.

28. Sokol. 1988. Uptake rate of phenol by *Pseudomonas putida* grown in unsteady state. Biotechnol. Bioeng., 32, 1097-1103.

29. Sokol, W., & Howell, J.A. 1981. Kinetics of phenol degradation by washed cells. Biotechnol. Bioeng., 23, 2039-2049.

30. Li, J., & Humphrey, A.E. 1989. Kinetic and flourimetric behavior of a phenol fermentation. Biotechnol. Lett., 11(3), 177-182.

31. Wang, S.J., & Loh, K.C. 1999. Modeling the role of metabolic intermediates in kinetics of phenol biodegradation. Enzyme Microb. Technol., 25, 77–184.

32. Alper, N., & Beste, Y. 2004. Modeling of phenol removal in a batch rector. Process Biochemistry, 40, 1233-1239.

33. Allsop, P.J., Chisti, Y., Moo-Young, M. & Sullivan, G.R. 1993. Dynamics of Phenol Degradation by *Pseudomonas putida*. Biotechnology and Bioengineering, 41, 572-580.

34. Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford: University Press.

CHAPTER 4- PROCESS MONITORING& FAULT DETECTION

The present chapter explores both product quality monitoring and process monitoring (fault detection of the process). For product quality monitoring purpose a wine dataset has been considered as case study and been explored with multivariate statistics, hence data based models in developing product quality monitoring methodologies. The unsupervised techniques like PCA and *K*-means clustering and supervised PLS technique were used to design statistical PLS based classifiers useful for wine quality monitoring. Artificial Adoptive Resonance theory network (ARTI) and Probabilistic neural network (PNN) were also used to design neural classifiers. In order to detect process faults (process monitoring), a time series data produced by phenol degradation was being used. CUMSUM and X-bar charts along with Moving Range and Range charts were used for the univariate process monitoring while PCA was used for the multivariate process monitoring in order to detect process abnormalities, if any.

4.1. WINE QUALITY MONITORING

4.1.1 WINE DATA SET

Wine dataset contains the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars which represents three diferent qualities of wine. A chemical analysis of 178 Italian wines from three different cultivars yielded 13 measurements. This dataset is often used to test and compare the performance of various classification algorithms. The chemical analysis determined 13 constituents found in each of the three types of wines. These are:

1) Alcohol

2) Malic acid

3) Ash

4) Alcalinity of ash

5) Magnesium

6) Total phenols

7) Flavanoids

8) Nonflavanoid phenols

9) Proanthocyanins

10)Color intensity

11)Hue

12)OD280/OD315 of diluted wines

13)Proline

4.1.2 Development of Statistical Classifier

The use of sensor arrays for producing features followed by the multivariate data analysis (MVDA) and different clustering techniques to discriminate among various samples paves the way to a successful design of a classifier. The use of various decision rules qualifies the classifier to be used for authentication purpose. Discrimination and classification of the feature variables produced from multisensory array owe a profound debt to the multivariate statistics these days. In these procedures, an underlying probability model must be assumed in order to calculate the posterior probability upon which the classification decision is made. One major limitation of the statistical models is that they work well only when the underlying assumptions are satisfied. The effectiveness of these methods depends to a large extent on the various assumptions or conditions under which the models are developed. Wine dataset containing178 numbers of wine samples possessing 13 numbers of feature variables was used to develop PLS based classifier, which is supposed to be very important development for on-line monitoring of wine samples.

4.1.2.1 Identification of Classes Present in the Data Using PCA and K-means Clustering

A data matrix (178 ×13) was generated from the wine dataset considered. Eigenvector decomposition was done on the data matrix. It was found that the first three principal components captured 2/3rd of the total variance. Table 4.1 shows the percentage variance covered by each principal component and their corresponding Eigen values. Scores were generated along the first and second principal component direction. Figure 4.1 represents the scores along principal component 1 (PC1) versus principal component 2 (PC2). The projection of all the data points of 13 dimensions into two- dimensional plane makes us to visualize the patterns present in the whole data set. From the Figure 4.1we can see that the data is distributed into three distinct groups. The possible distinct groups have been apparently separated in the figure. For confirming this finding, the K-means clustering has been applied by defining different number of clusters. The mean squared error (MSE) of miss classification also represented that three numbers of clusters is optimum with the least MSE. The stable K-means statistics of the scores along PC1-PC2 are presented in Table 4.2, which presents all the 3 cluster centroids, the number of data points pertaining to each cluster. As a part of hierarchical clustering, the distance matrix or a dissimilarity matrix was generated, which was symmetric along the diagonal (all the diagonal elements are zero). A hierarchical cluster tree was then created with that distance matrix to form two dendrogram (Figure 4.2) originated from the score vectors along PC1-PC2. In a dendrogram, the grouping of the branches down the tree represents the formation of clusters as the distance between the cluster centers becomes very low. Only 30 number of data points of whole data set were represented in the figure so that one can visualize the data points distinctly. Including more data points will make the dendrogram look clumsy. In the dendrogram we can see three groups;

first group forming at an inter cluster distance above 4.5 and the second at around 4.0 and third group around at inter cluster distance of 3.5.

Once the patterns were found and confirmed that there are clearly three distinct groups in the whole dataset, K-means clustering has been applied in 13 dimensions with a predefined cluster number of three. The K-means has formed three groups and assigned all the data points of wine samples to any one of those three groups. Initially the data was having only the information about the 13 attributes, but now the data is having 13 attributes of all the wine samples (inputs) along with their corresponding group numbers (outputs). This information, provided by K-means was used for the development of different types of classifiers.

4.1.2.2 PLS Based Classifier Development & its Performance

PLS based classifier falls under the category of statistical classifier. This methodology has successfully been used in a large number of areas like metabonomic [1] and transcriptomic [2] studies. In the present problem PLS has been used as a classifier.

The wine dataset considered was successfully clustered in to 3 groups by k-means clustering. The 3 different classes of wine samples were represented as three numbers of vectors and they were amalgamated to give raise the predictor X matrix. A corresponding response $\boldsymbol{Y}$ matrix (178×3) indicating the wine class was generated. In the $\boldsymbol{Y}$ matrix, '1' represents the presence of an individual class and 0 represents its absence. Each of the 178 row vectors of the $\boldsymbol{Y}$ matrix is either 1 0 0 or 0 1 0 or 0 0 1. Each of the wine samples was having randomly chosen 105 numbers of data (chosen out of 178 numbers) for training and 73 numbers for testing containing 13 feature variables. Three classes of X vectors were regressed by PLS to three numbers of characteristic$Y$vectors.

Each of the regressed Y vectors of the response matrix was given a class membership by choosing an entry most close to 1.0; uniquely existing among any one of the 3 columns of them. The designed PLS classifier was then used for predicting Y s' representing unknown sample classes corresponding to unknown X samples. The Nonlinear Iterative Partial Least Squares (NIPALS) algorithm originally proposed by Chiang et. al. (1992) was adapted here and used in the present work for detecting unknown wine class is as follows [3]:

1. Formation of training/ predictor matrix $X_{103\times13}$: Out of the row vectors, row vector 1 - 45 = sample class 1; row vectors 46 - 90 = sample class 2; row vectors 91-103 = sample class 3.
2. Assigning response matrix $Y_{103\times3}$ as class identification matrix consisting of 1's and 0's only.
3. Relating X and Y by PLS regression; hence determination of the matrix of regression coefficient and loading matrices corresponding to X and Y data.
4. Formation of test X matrix of dimension(73×13)
5. Prediction of 73 numbers of $(1 \times 3)Y$ vectors corresponding to 73 numbers of test vectors of the test X dataset using the developed model in step 4.
6. Determination of 73 numbers of (1×3) dimensional abs $(Y - 1)$ vectors.
7. Detection of outlier; A sample not among the considered wine classes: Detection of the minimum entry value among all the 3 numbers of columns for each of 73 numbers of (1×3) dimensional abs $(Y - 1)$ vectors. If any of the minimum entry from the 3 number of columns (abs$(Y - 1)$) for those 73 numbers of test vectors is > (± 15 % of 1.0), most likely; the sample class corresponding to that is an outcaste.

8. Generation of class membership: Detection of the minimum entry value among all the columns for each of the 73 numbers of (1×3) dimensional abs $(Y-1)$ vectors is synonymous to find out the entry close to 1.0 among all the columns for each of the 73numbers of (1×3) dimensional PLS regressed abs (Y) vectors. The column number corresponding to the minimum entry is the class the $identificationnumber$ of that k^{th} test vector (among 73 test vectors). In this way all the 73 numbers of vectors of the test X dataset occupied a class membership ranging from 1 -3.

   If the

$$identificationnumber = 1, \qquad detectedclass = 1$$

$$identificationnumber = 2, \qquad detectedclass = 2$$

$$identificationnumber = \ 3, \qquad detectedclass = 3$$

The classifier operated with almost 100 % efficiency. Figure 4.3shows the wine classification performance with 2 misclassifications over 73 numbers of samples.Misclassification means categorizing of one class of sample as other class (misclassification rate percentage is ((number of misclassifications/ total number of samples)*100). It was found to be 2.0 % for the present case). One of the samples belonged to class '1' type wine was classified as wine class '2' and the other sample belonged to sample class '2' was classified as wine class '3' by the designed PLS classifier. This development seems to be a potential one so far on-line monitoring of beverage quality is concerned. The code for the statistical classifier was developed using MATLAB.

4.1.3 Development of Neural Classifier

Neural networks have been successfully applied to a variety of real world classification tasks in industry, business and science [4]. Applications include bankruptcy prediction [5,6], handwriting recognition [7,8], speech recognition [9,10], product inspection [11,12], fault detection [13,14], medical diagnosis [15,16], and bond rating [17,18]. A number of performance comparisons between neural and conventional classifiers have been made by many studies [19-21]. In addition, several computer experimental evaluations of neural networks for classification problems have been conducted under a variety of conditions [22,23].

Two network architectures namely Probabilistic Neural Networks (PNN) and Adaptive Resonance Theory (ART) networks were employed for the classification among wine samples in the current study. Their performances were presented below.

4.1.3.1 Probabilistic neural Network (PNN) Based Classifier Development & its Performance

PNN act as a better classifier than a function approximator. For developing a PNN based classifier the target vector which represents the cluster number of the each sample, has been converted from indices (1,2 and 3) to vectors ([1 0 0], [0 1 0] and [0 0 1]) as mentioned in the section 4.1.2.2. Once the output vectors were formed for all the samples, the whole dataset containing 178 samples of 13 attributes along with their cluster numbers has been redistributed into six randomly selected datasets containing 20%, 30%, 40%, 50%, 60% and 70% of data. The networks were trained by using those randomly selected data sets and tested against the remaining percentage of the dataset. The random selection of data was done using the method described by Box and Muller (1958) and Devroye (1986) [24, 25]. The networks were trained using ANN toolbox of MATALB. The performances of developed PNN networks as a classifier are presented in Table 4.3. The results clearly indicated that maximum efficiency was achieved when the training was carried out using 70% of the data. Thus one can conclude that sufficient data is required for PNN to predict the posterior probabilities accurately.

4.1.3.2 ART1 Network Based Classifier Development & its Performance

Processing the data according to the demand of ART1 network architecture plays an important role in their successful implementation. Normalization of data matrix and conversion of scaled data matrix into binary matrix are the major steps in this regard. The data has been normalized and converted into binary data according to the procedure mention in section 2.5.1.2.3 of Chapter 2. The 8%, 20%, 30%, 44% , 56%, 70%, 80% and 92% of 178 binary data samples were randomly chosen as training data sets as well as target data sets (n×14 matrix) for the ART1 network. The first 13 columns of the data set formed the 178 number of input feature vectors and 14th column serves as 178 numbers of targets or class tags. 3 different classes of training and testing pools were created out of 178 samples or feature vectors to design three different classifiers as ART1-1, ART1-2, & ART1-3. In a particular data pool; either training or testing, the presence of any one of the three classes of feature vectors are targeted as '1' and any other class of feature vectors apart from that category in that pool are targeted as '0'. The ART1 networks developed were very robust as reflected by its classification efficiency of 100 % for all combinations of training and testing vectors. A randomly selected 20 % of the data from the data base were used for training each of the ART1-1, ART1-2, and ART1-3 networks (The two vigilance parameters (ρ) of 0.4 & 0.7 with 100 iterations were used for the training of the networks) and simulation for three trained networks were done with the corresponding randomly selected samples containing 20%, 30%, 40%, 50%, 60% & 70% data. A representative (10×10 matrix, including only 9 feature columns + 1 target column) normalized data matrix is shown in

Table 4.4. Table 4.5 is a representative (10×12 matrix, including only 9 feature columns + 3 target columns) binary data matrix. Table 4.6, 4.7 & 4.8 represent the training time & efficiency of the networks.The code for the ART1 based classifier was developed using MATLAB.

From the above one can clearly say that adaptive resonance theory has its own superiority over other classifiers. This is mainly due to the plasticity of the ART1 network. Due to this nature, an ART1 network accepts an input vector and classifies it as one of the categories depending upon which of the stored pattern it resembles within a specified tolerance otherwise a new category is created by storing that pattern as an input vector. It makes ART1 networks superior than conventional back propagation neural networks like PNN. The low performance of PNN can be attributed to the fact that it can learn nothing with continuously modifying their weights without a respite of getting a stationary setting.

4.2. ONLINE PROCESS MONITORING OF PHENOL DEGRADATION

Monitoring of the process parameters which affect the product quality and keeping them in control will ensure the quality of the product. At the same time detection of process abnormalities or faults and their diagnosis is very important so far as the maintenance of the product specification is concerned. The successful functioning of any plant needs proper monitoring of the important process parameter and earlier detection of abnormal operating condition, which may avoid run away situations. The time series data produced during the phenol degradation experiments have been used to monitor three variables namely temperature, pH and RPM.

4.2.1 Monitoring of Process parameters using univariate statistics

CUSUM integrated with Moving Range Charts and $\overline{x}$chart integrated with R charts used to monitor the process in a univariate manner. Figures 4.4, 4.5 & 4.6 show the CUSUM and Moving Range charts of temperature, pH and RPM respectively. Figures 4.7, 4.8 & 4.9 show the X-bar and Range charts of temperature, pH and RPM respectively. The data produced in the 4th run have been used for the monitoring of process parameters. The control limits for these charts have been customized by altering the multiples of σ values. The multiples of sigma were taken in such a way that the control limits will represent the 95% confidence intervals. From the figures one can observe that the deviation between the R-charts and Moving range charts are prominent. These are due to the fluctuations in the subset values i.e. triplicate values that have been taken for readings. Anyway, only X-bar and CUSUM charts were considered for the process fault detection purpose. Both X-bar charts and CUSUM charts indicated certain common operating points where the parameters were out of control in both the charts. CUSUM chart for temperature had shown 3 numbers of instances where the temperature went out of control whereas X-bar chart had shown 8 numbers of such instances having one commonality. The CUSUM & X-bar charts for pH had shown 4 numbers of deviations from normal operating condition having two numbers of common deviations. The CUSUM chart for RPM produced 5 outliers whereas X-bar produced 2 outliers having no common deviations. 23 numbers of faulty situations were found over four numbers of process parameters. All the deviation points were noted and checked whether the multivariate statistical process control can identify the same abnormalities.

4.2.2 Monitoring the process parameters using multivariate statistics

The same time series data has been used for the Multivariate Statistical Process Monitoring (MSPC) purpose using PCA. In MSPC all three variables were considered at a time and their corresponding principal components were recognized. The data points were projected onto these new dimensions considering two components at a time i.e. PC1 & PC2, PC2 & PC3 and PC1 & PC3. As the principal components are the directions in which maximum variance is present, all the data points should fall within a cluster. Points that deviate from the normal operating conditions should fall apart. Figures 4.10, 4.11 & 4.12 present the projections of all the process variables/features on to PC1&PC2, PC1&PC3 and PC2&PC3 respectively. The Ellipses in all the figures represent the 95% confidence levels of the axes values. The major and minor axes of the ellipse are defined as:

$$Centroid=(\mu_x,\mu_y)$$

$$Length\ of\ Major\ axis\ (a)=\mu_x\pm 0.95R_x/2$$

$$Length\ of\ Minor\ axis\ (b)=\mu_y\pm 0.95R_y/2$$

Where, μ is the mean value of the corresponding coordinates and R is the range for the same. The points out of the ellipse represent the outliers. Here one can alter the confidence levels to maintain stringent control limits. Projections on PC1&PC2, PC1&PC3 and PC2&PC3 have produced 11, 15 and 12 outliers, respectively, producing a total of 26 outliers excluding the common points. In comparison to the traditional SPM, the MSPC yielded 2 new outliers or abnormal operating situations namely at 3rd and 102nd instant. Thus multivariate approach in process monitoring always helps in detecting abnormal conditions in a process where individual process variables/features may seem to be under control but their combination may produce an abnormal situation affecting the process in an adverse way. One of the major characteristics of multivariate data is that the variables being measured are almost never independent, but rather, they are highly correlated with one another at any given time. Hence multivariate process monitoring will be the best approach to monitor a process where many inter related process variables are involved.

TABLES:

Table 4.1: Eigen values and % variances captured by the Principal Components

Principal Component	Eigen value	%Total Variance	Cumulative Eigen values	Cumulative %
1	4.705850	36.19885	4.70585	36.1988
2	2.496974	19.20749	7.20282	55.4063
3	1.446072	11.12363	8.64890	66.5300
4	0.918974	7.06903	9.56787	73.5990
5	0.853228	6.56329	10.42110	80.1623
6	0.641657	4.93582	11.06276	85.0981
7	0.551028	4.23868	11.61378	89.3368
8	0.348497	2.68075	11.96228	92.0175
9	0.288880	2.22215	12.25116	94.2397
10	0.250902	1.93002	12.50206	96.1697
11	0.225789	1.73684	12.72785	97.9066
12	0.168770	1.29823	12.89662	99.2048
13	0.103378	0.79521	13.00000	100.0000

Table 4.2: Statistics of K-means Clustering (Score along PC1-PC2)

Cluster Identity	No. of Samples	Cluster Centroid
1	49	1.2613,0.7622
2	61	-1.0521,0.6057
3	68	0.0349,-1.0955

Table 4.3: Classification performance of the developed PNN networks.

		Training Set					
		20%	**30%**	**40%**	**50%**	**60%**	**70%**
% Accuracy for test sets	**20%**	--	55.88235	69.69697	66.66667	78.78788	90.90909
	30%	43.75	--	64.58333	70.83333	70.83333	83.33333
	40%	48.78049	56.09756	--	71.95122	79.26829	84.14634
	50%	47.31183	57.6087	70.65217	--	72.82609	83.69565
	60%	46.46465	49.49495	71.71717	69.69697	--	85.85859
	70%	47.72727	54.54545	68.18182	71.21212	74.24242	--

Table 4.4: Representative normalized data (10×10) matrix for training ART1 and PNN networks

Sample No	Alcohol	Ash	Alcalinity	Magnesium	Phenols	Color	Hue	Dilution	Proline	Cluster #
1	0.641	0.290	0.397	0.449	1.000	0.222	0.561	0.353	0.667	1
2	0.024	0.347	0.000	0.000	0.129	0.111	0.255	0.235	0.019	1
3	0.000	0.815	0.726	0.755	0.161	0.222	0.745	0.471	1.000	1
4	0.725	0.484	0.493	0.571	0.548	1.000	1.000	0.118	0.596	2
5	0.048	1.000	1.000	1.000	0.710	0.222	0.184	1.000	0.365	1
6	0.623	0.331	0.425	0.408	0.516	0.570	0.898	0.706	0.462	1
7	0.737	0.419	0.425	0.347	0.000	0.000	0.010	0.471	0.468	1
8	0.539	0.645	0.644	0.653	0.806	0.074	0.000	0.529	0.000	3
9	1.000	0.234	0.041	0.286	0.032	0.222	0.480	0.412	0.468	1
10	0.419	0.000	0.178	0.490	0.065	0.356	0.653	0.000	0.385	3

Table 4.5: Representative binary data (10×12) matrix for training ART1 networks

Sample No	Alcohol	Ash	Alcalinity	Magnesium	Phenols	Color	Hue	Dilution	Proline	Cluster 1	Cluster 2	Cluster 3
1	1	0	0	0	1	0	1	0	1	1	0	0
2	0	0	0	0	0	0	0	0	0	1	0	0
3	0	1	1	1	0	0	1	0	1	1	0	0
4	1	0	0	1	1	1	1	0	1	0	1	0
5	0	1	1	1	1	0	0	1	0	1	0	0
6	1	0	0	0	1	1	1	1	0	1	0	0
7	1	0	0	0	0	0	0	0	0	1	0	0
8	1	1	1	1	1	0	0	1	0	1	0	1
9	1	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	1	0	0	0	0	1

Table 4.6: Performance of the ART1 networks with vigilance parameter 0.4 & 100 iterations

Training vector %	Testing vector%	ART1-1		ART1-2		ART1-3	
		Training time (s)	Efficiency%	Training time(s)	Efficiency%	Training time(s)	Efficiency%
8	**92**	0.177	100	0.199	100	0.200	100
20	**80**	0.275	100	0.371	100	0.305	100
30	**70**	0.384	100	0.360	100	0.509	100
44	**56**	1.214	100	1.244	100	1.309	100
56	**44**	1.44	100	1.419	100	1.59	100
70	**30**	1.29	100	1.442	100	1.816	100
80	**20**	1.248	100	1.495	100	1.200	100
92	**8**	2.646	100	2.690	100	2.015	100

Table 4.7: Performance of the ART1 networks with vigilance parameter 0.7& 100 iterations

Training Vector %	Testing vector%	ART1-1		ART1-2		ART1-3	
		Training time (s)	Efficiency	Training time (s)	Efficiency	Training time (s)	Efficiency %
8	**92**	0.169	100	0.216	100	0.220	100
20	**80**	0.317	100	0.356	100	0.266	100
30	**70**	0.456	100	0.376	100	0.577	100
44	**56**	1.02	100	0.964	100	0.986	100
56	**44**	1.551	100	1.589	100	1.459	100
70	**30**	1.558	100	1.591	100	1.808	100
80	**20**	1.697	100	1.604	100	1.913	100
92	**8**	2.811	100	2.775	100	2.684	100

Table 4.8: Performance of ART1 networks trained with randomly selected 20% data

			Test Data set					
			20%	**30%**	**40%**	**50%**	**60%**	**70%**
ART1-1	**Computation Time (s)**	**ρ=0.4**	0.875	1.0046	0.9709	1.0073	0.872	0.9029
		ρ=0.7	0.955	0.9366	1.0522	0.8738	1.0361	0.9216
	Efficiency	**ρ=0.4**	100%	100%	100%	100%	100%	100%
		ρ=0.7	100%	100%	100%	100%	100%	100%
ART1-2	**Computation Time (s)**	**ρ=0.4**	0.892	0.8673	1.0057	0.966	0.9605	0.9626
		ρ=0.7	1.0505	1.1874	1.0369	1.0149	1.004	0.9506
	Efficiency	**ρ=0.4**	100%	100%	100%	100%	100%	100%
		ρ=0.7	100%	100%	100%	100%	100%	100%
ART1-3	**Computation Time (s)**	**ρ=0.4**	1.0115	0.9378	1.0767	0.8299	0.9825	0.8963
		ρ=0.7	1.0596	0.9239	0.9363	0.9807	1.074	1.0218
	Efficiency	**ρ=0.4**	100%	100%	100%	100%	100%	100%
		ρ=0.7	100%	100%	100%	100%	100%	100%

FIGURES:

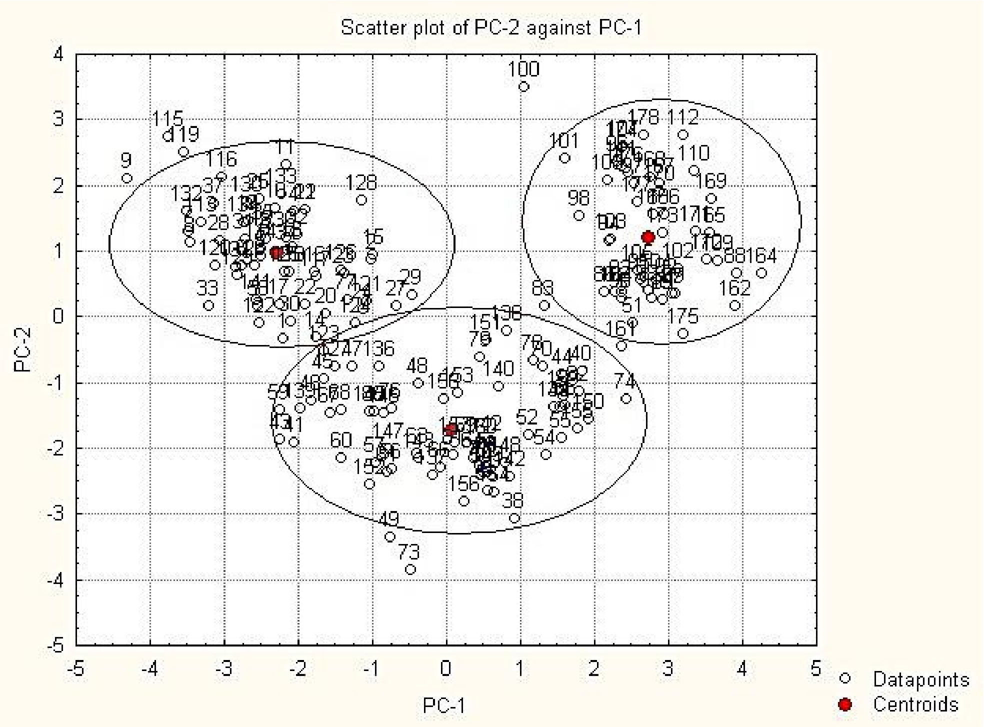

Figure 4.1: Discrimination and clustering of scores along PC1-PC2

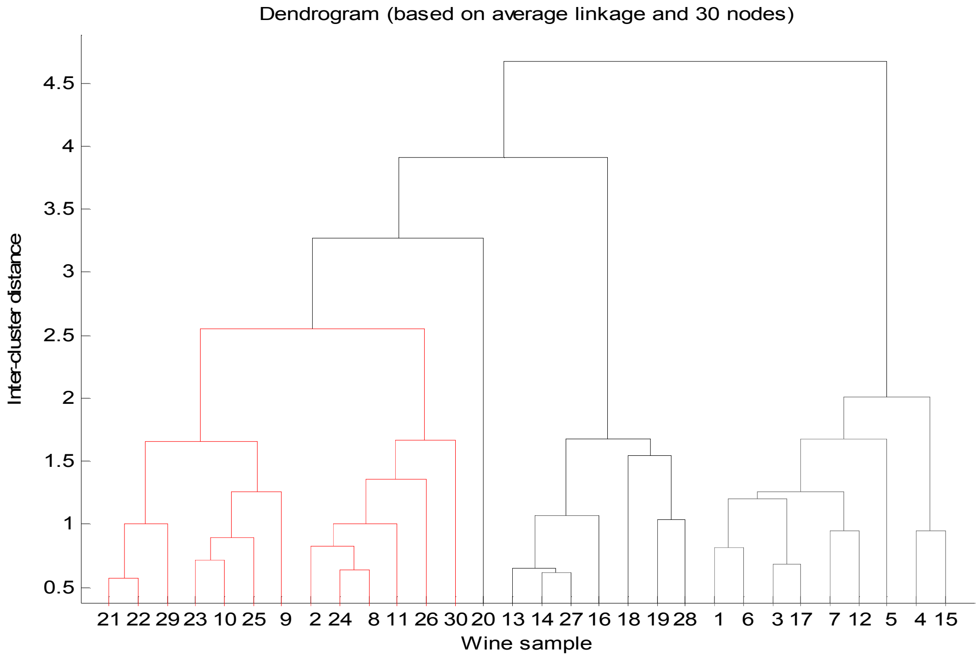

Figure 4.2: Dendrogram on score along PC1-PC2

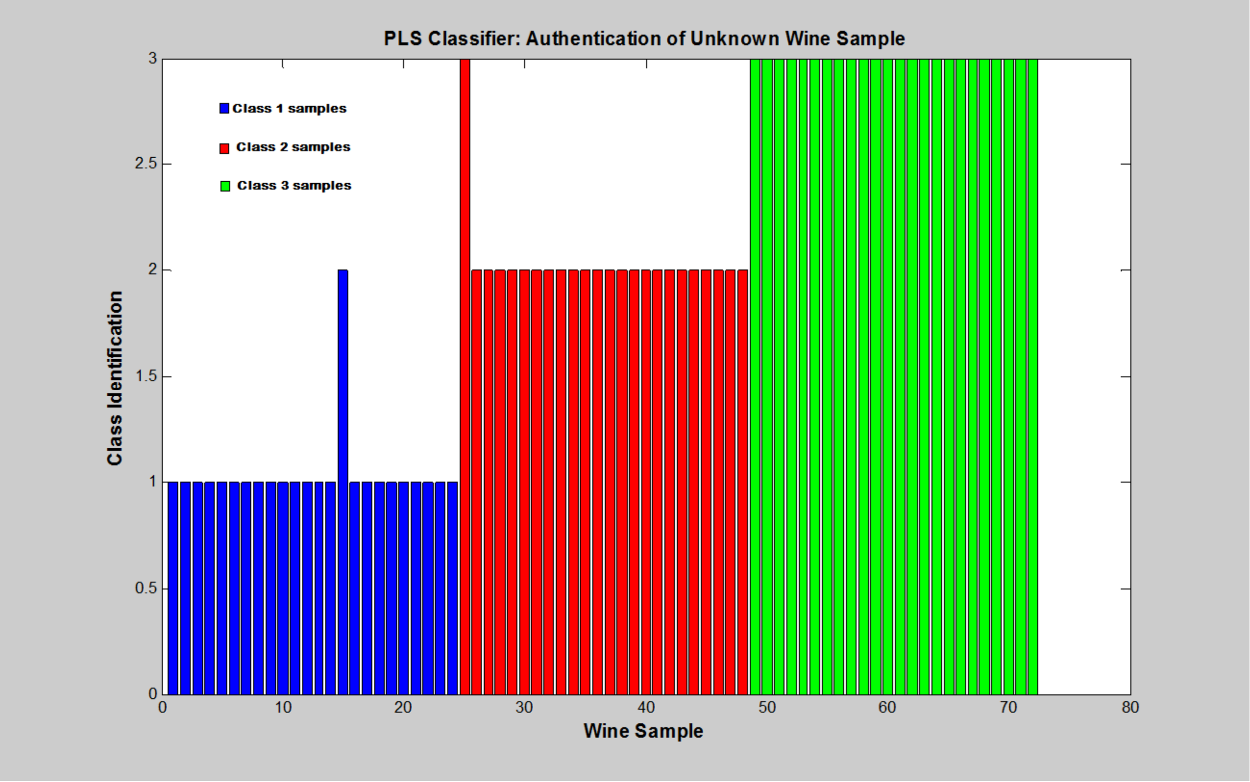

Figure 4.3: Performance of the developed PLS based classifier

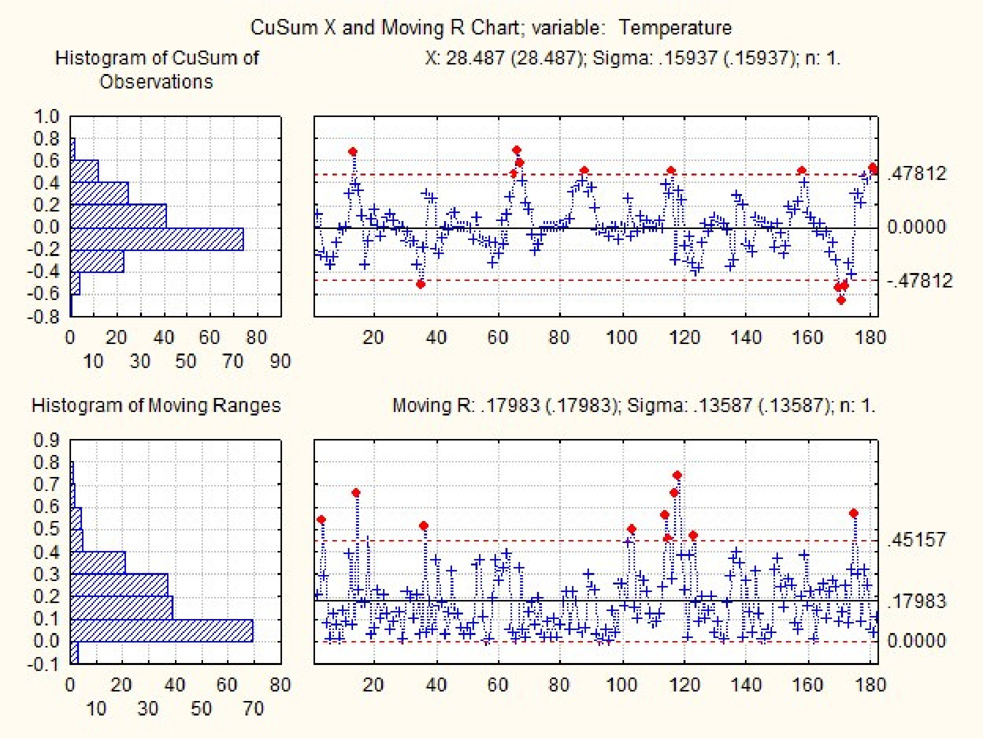

Figure 4.4: CUSUM and Moving range chart of temperature.

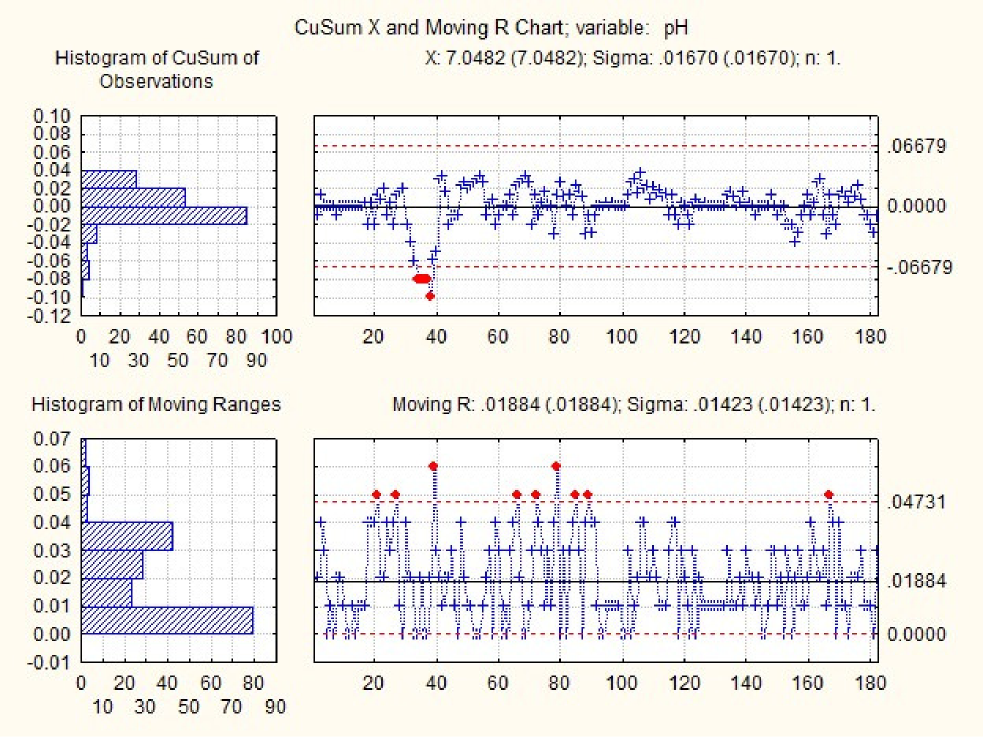

Figure 4.5: CUSUM and Moving Range chart of pH.

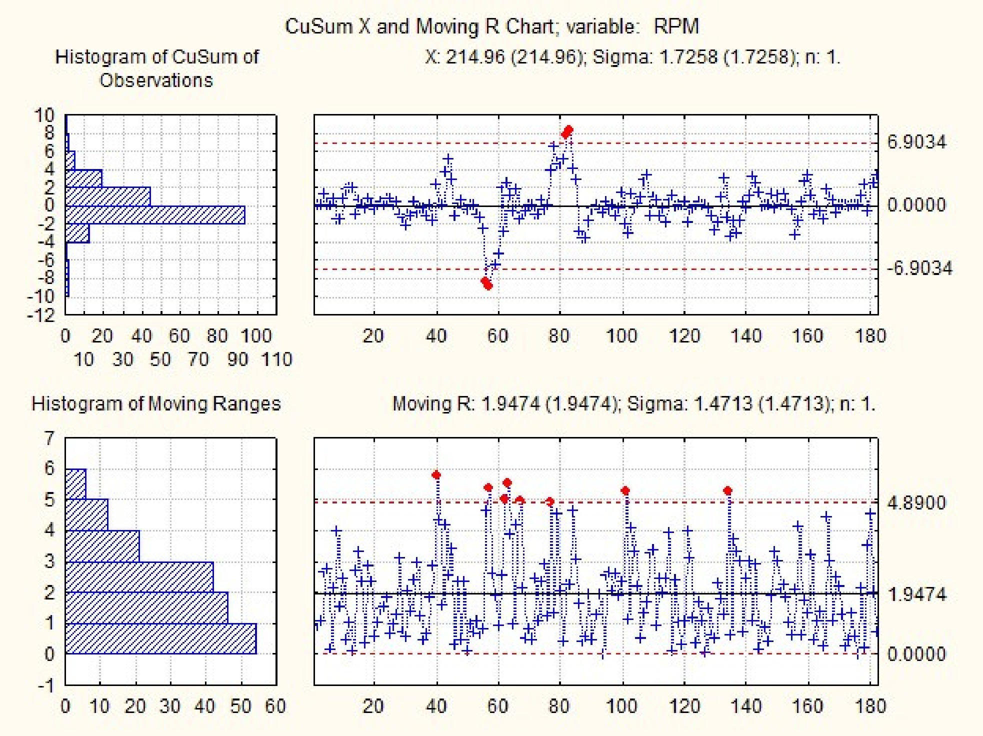

Figure 4.6: CU-SUM and Moving R Charts of RPM.

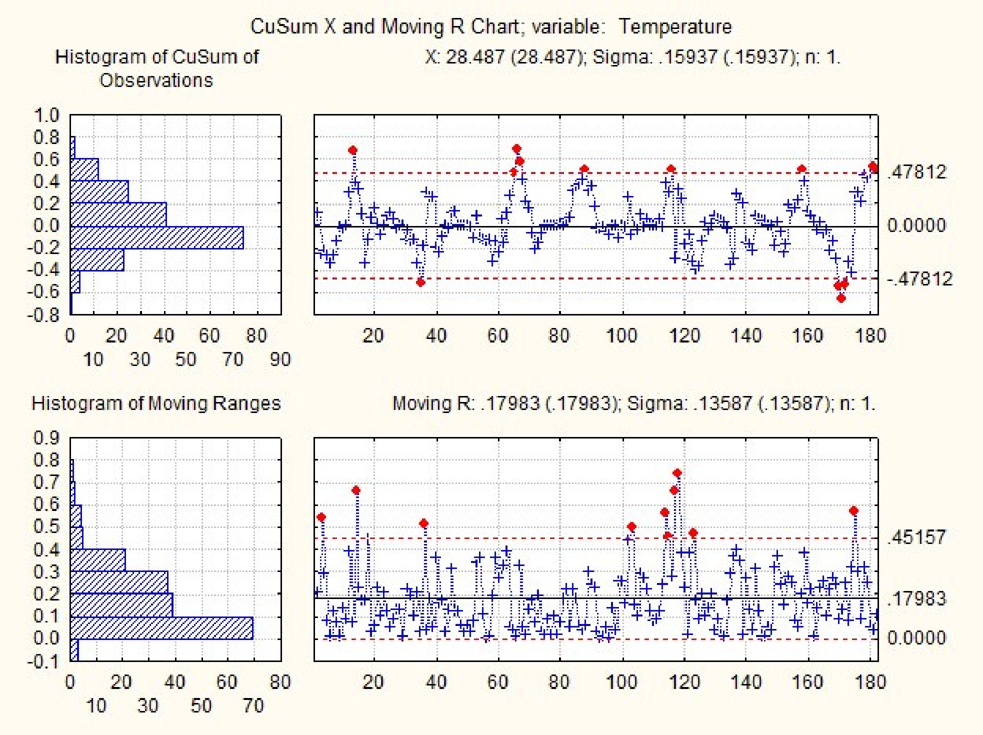

Figure 4.7: X-bar and Range chart of temperature.

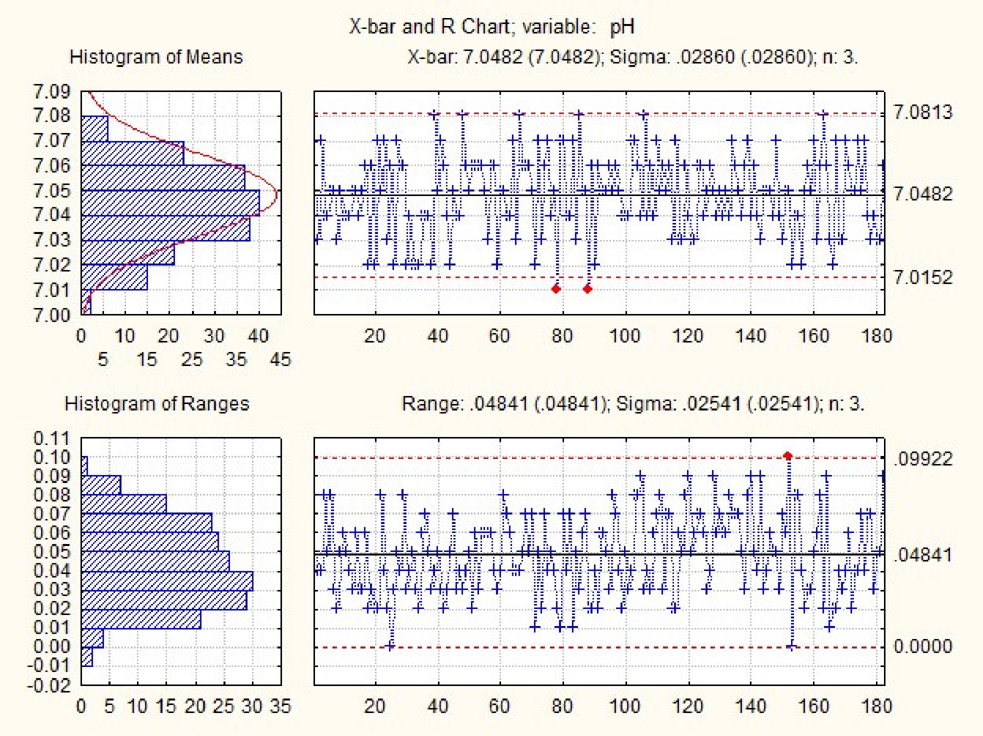

Figure 4.8: X-bar and Range chart of pH.

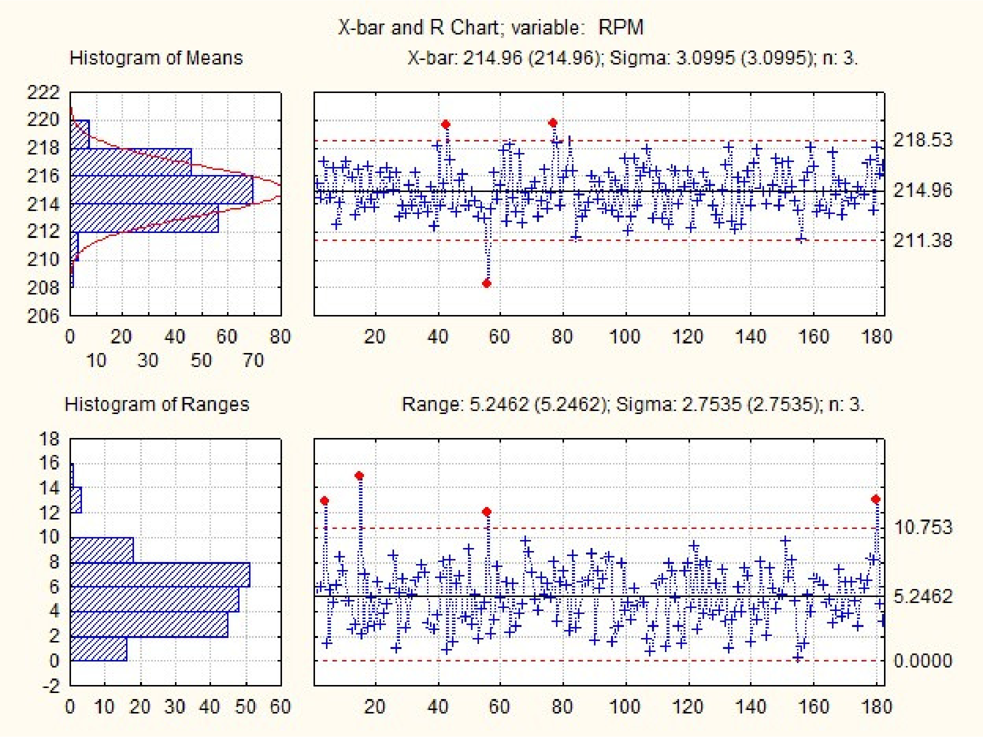

Figure 4.9: X-bar and Range chart of RPM

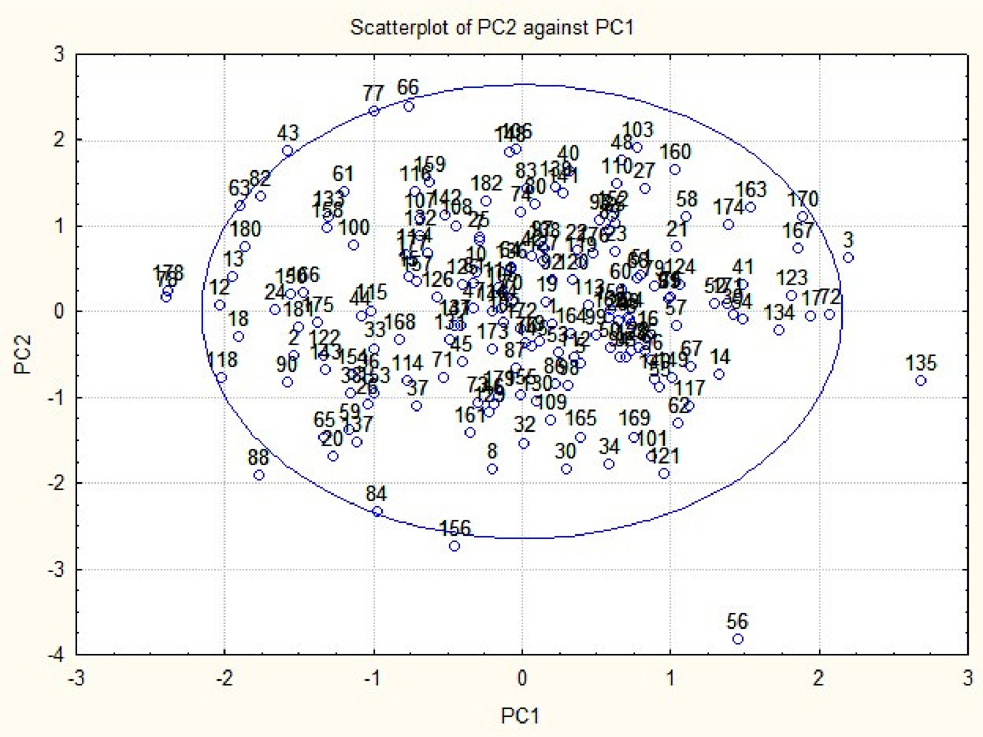

Figure 4.10: Projections on PC-1 and PC-2 with ellipse representing 95% confidence.

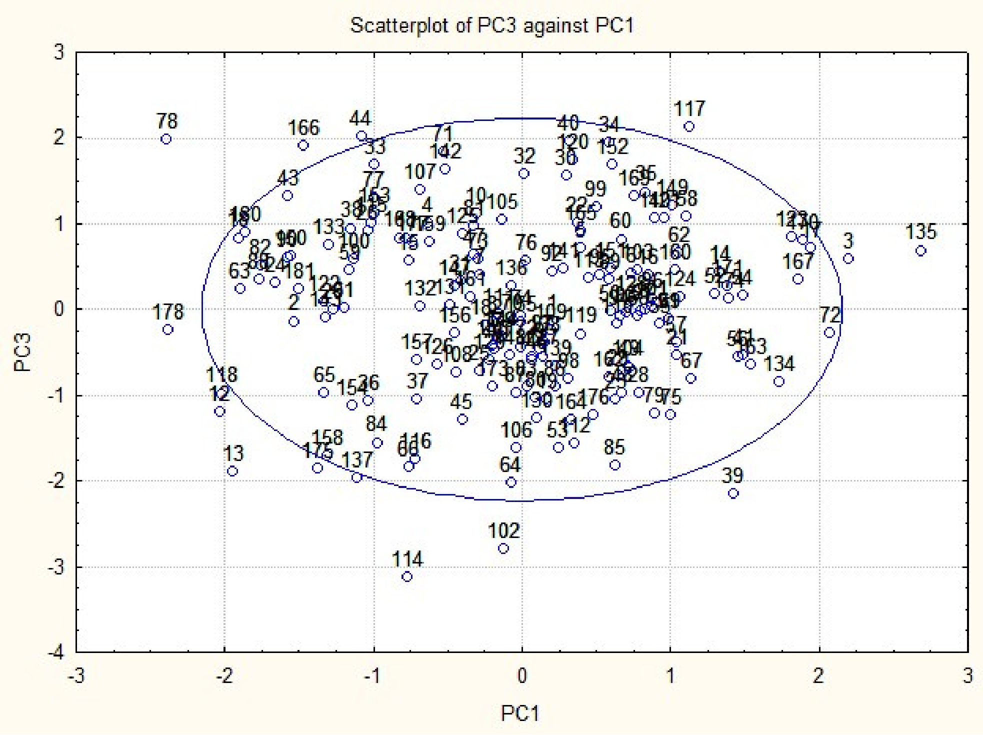

Figure 4.11: Projections on PC-1 and PC-3 with ellipse representing 95% confidence.

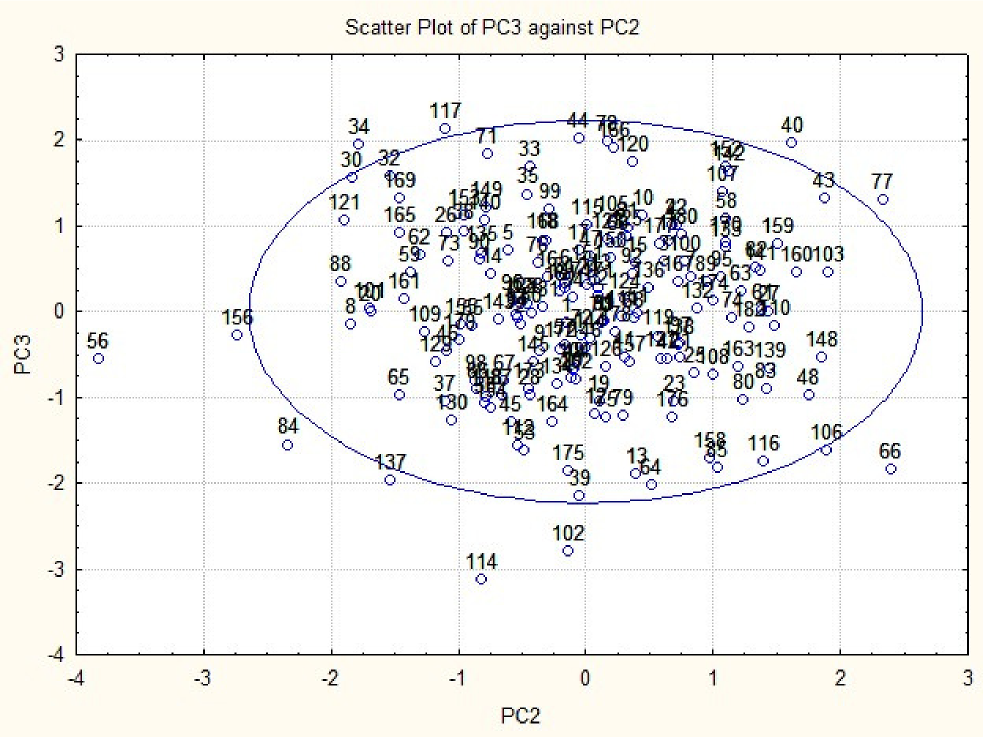

Figure 4.12: Projections on PC-1 and PC-3 with ellipse representing 95% confidence.

REFERENCES:

1. Jonsson, P., Bruce, S.J., Moritz, T., Trygg, J., Sjo¨stro¨m, M., Plumb, R., Granger, J., Maibaum, E., Nicholson, J.K., Holmes, E., & Antti, H. 2005. Extraction, interpretation and validation of information for comparing samples in metabolic LC/MS data sets. Analyst, 130, 701–707.
2. Perez-Enciso, M., & Tenenhaus, M. 2003. Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (PLS-DA) approach. Hum. Genet., 112, 581–592.
3. Chiang, Y.Q., Zhuang, Y.M & Yang, J.Y. 1992. Optimal Fisher discriminant analysis using the rank decomposition. Pattern Recognition, 25, 101-111.
4. Widrow, B., Rumelhard, D.E., & Lehr, M.A. 1994. Neural networks: Applications in industry, business and science. Commun. ACM, 37, 93–105.
5. Altman, E.I., Marco, G., & Varetto, F., 1994. Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (theItalian experience). J. Bank. Finance, 18, 505–529.
6. Lacher, R.C., Coats, P.K., Sharma, S.C., & Fant, L.F. 1995. A neural network for classifying the financial health of a firm. Eur. J. Oper. Res., 85, 53–65.
7. Guyon, I. 1991. Applications of neural networks to character recognition. Int. J. Pattern Recognit. Artif. Intell., 5, 353–382.
8. Knerr, S., Personnaz, L., & Dreyfus, G. 1992. Handwritten digit recognition by neural networks with single-layer training. IEEE Trans. Neural Networks, 3, 962–968.
9. Bourlard, H., & Morgan, N. 1993. Continuous speech recognition by connectionist statistical methods. IEEE Trans. Neural Networks, 4, 893–909.

10. Lippmann, R.P. 1989.Review of neural networks for speech recognition. Neural Comput., 1, 1–38.

11. Lampinen, J., Smolander, S., & Korhonen, M. 1998. Wood surface inspection system based on generic visual features, in Industrial Applications of Neural Networks, Soulie, F.F., &Gallinari, P., Eds, Singapore: World Scientific, 35–42.

12. Petsche, T., Marcantonio, A., Darken, C., Hanson, S.J., Huhn, G.M., & Santoso, I. 1998. An autoassociator for on-line motor monitoring, in Industrial Applications of Neural Networks, Soulie, F.F., & Gallinari, P., Eds, Singapore: World Scientific, 91–97.

13. Barlett, E.B., & Uhrig, R.E. 1992. Nuclear power plant status diagnostics using artificial neural networks. Nucl. Technol., 97, 272–281.

14. Hoskins, J.C., Kaliyur, K.M., & Himmelblau, D.M. 1990. Incipient fault detection and diagnosis using artificial neural networks, in Proc. Int. Joint Conf. Neural Networks, 81–86.

15. Baxt, W.G. 1990. Use of an artificial neural network for data analysis in clinical decision-making: The diagnosis of acute coronary occlusion. Neural Comput., 2, 480–489.

16. Baxt, W.G. 1991. Use of an artificial neural network for the diagnosis of myocardial infarction. Ann. Internal Med., 115, 843–848.

17. Dutta, S., & Shekhar, S. 1988. Bond rating: A non-conservative application of neural networks, in Proc. IEEE Int. Conf. Neural Networks, 2, San Diego, CA, 443–450.

18. Surkan, J., & Singleton, J.C. 1990. Neural networks for bond rating improved by multiple hidden layers, in Proc. IEEE Int. Joint Conf. Neural Networks, 2, San Diego, CA, 157–162.

19. Curram, S.P., & Mingers, J. 1994. Neural networks, decision tree induction and discriminant analysis: An empirical comparison. J. Oper. Res. Soc., 45(4), 440–450.

20. Huang, W.Y., & Lippmann, R.P. 1987. Comparisons between neural net and conventional classifiers, in IEEE 1st Int. Conf. Neural Networks, San Diego, CA, 485–493.

21. Michie, D., Spiegelhalter, D.J., & Taylor, C.C., Eds. 1994. Machine Learning, Neural, and Statistical Classification, London, U.K.: Ellis Horwood.

22. Patwo, E., Hu, M.Y., & Hung, M.S. 1993. Two-group classification using neural networks. Decis. Sci., 24(4), 825–845.

23. Subramanian, V., Hung, M.S., & Hu, M.Y. 1993. An experimental evaluation of neural networks for classification. Comput. Oper. Res., 20, 769–782.

24. Box, G.E.P., & Muller, M.E. 1958. A note on the generation of random normal deviates. Annals of Mathematical Statistics, 29, 610-611.

25. Devroye, L. 1986. Non-uniform random variate generation. Springer, New York.

CHAPTER 5- CONCLUSION

5.1. CONCLUSION

Present work addressed several problems related to process identification, process quality monitoring and process fault detection with the help of machine learning algorithms. The statistical and neural techniques including PCA, *K*-means clustering, PLS, PNN, and ART1 were used to approach the best possible solutions of the problems taken up. Efficient pre-processing of process data, design of efficient algorithms have been the key steps for those data based models developed. Efficient models thus developed could identify and predict the process without a priori knowledge of the process like Phenol degradation by *Pseudomonas putida* (ATCC: 11172). The time series data generated in this work were used to develop ANN and ARX based predictive models relating phenol degradation process with the variables like temperature, pH, RPM and Phenol loading at any particular instant. PLS technique was used to develop an empirical model relating phenol degradation process with the variables like temperature, pH, RPM and Phenol loading at steady state.

Classical analytical techniques such as various chromatography, spectrometry, and etc are used for the determination of different characteristics of food and beverage samples. However they are time-consuming, expensive, and laborious which can hardly be done on-site or on-line. Development of a feature based classifier could circumvent the problem of monitoring food quality without relating instrumental analysis to biological sensing like ageing and spoilage of the product. A wine data set containing 178 samples and their corresponding 13 features was taken as a case study. The unsupervised technique like PCA and *K*-means clustering were used to reduce the dimensionality and classify the samples into three groups followed by the development of supervised classifiers using various machine learning algorithms like Artificial Adoptive Resonance theory network (ARTI), Probabilistic neural network (PNN) and Partial least squares (PLS). All the designed classifiers emancipated encouraging performance.

The univariate and multivariate statistical monitoring of phenol degradation process were carried out to detect process abnormalities. Different SPC charts and PCA were used monitoring the process of phenol degradation; hence identification of process faults, if any. A biodegradation process of an organic pollutant phenol was used for the fault detection purposes. The detection of fault followed by its diagnosis is extremely important for effective, economic, safe and successful operation of a process.

5.2. FUTURE RECOMMENDATION

The following are the future recommendations:

- Development of data based model for identification of more complex processes.
- Development of ART2 and PLS based classifiers for different beverage and water classification with a view to on-line process quality monitoring.
- Monitoring, detecting, and diagnosing faults in more complex processes using hierarchical and multi-way PCA.