An Artificial Neural Network Based Electronic-Nose System: Tea and Spice Flavour Discrimination and Drift Parameter Determination

A Thesis Submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

Kishana Ram Kashwan

Registration No. 186/2000

School of Science & Technology
Department of Electronics
Tezpur University (A Central University)
Napaam Tezpur, Assam (India) 784028

October 2005

Abstract

Electronic nose (E-nose) is an Intelligent Sensor based device, which mimic the sense of smell of human nose (olfaction). It consists of an array of chemo-resistor sensors, head space generation and sampling, high precision signal-processing unit to digitize the sensor responses for signal processing, online data acquisition unit to collect and transport odours, Artificial Neural Network (ANN) based soft computing machines for Pattern Recognition (PARC) and classification. The sensors are broadly tuned (non-specific) and are sensitive to a variety of biological and chemical materials. The output of the E-nose can be an identification of odour or estimation of concentration of odour or characteristic properties of the odour. Odourant stimulus generates a characteristic odour fingerprint (change in physical properties such as resistance) from the sensor array. Patterns or fingerprints from known odours are used to construct a database and train a pattern recognition system so that unknown odours can subsequently be classified and identified.

The fundamental to the artificial nose is the idea that each sensor in the array has different sensitivity to different odours. The pattern of response across the sensors is distinct for different odours. This distinguishability allows the E-nose to identify an unknown complex odour from the pattern of sensor response. Applications of E-nose include pharmaceutical industry, environmental monitoring, fragrance and cosmetics production, food and beverages manufacturing, chemical engineering, and more recently, medical diagnostics and bioprocesses. E-nose is classified as one of the Artificial Intelligence (AI) systems.

Tea Flavour: Tea is one of the most popular and the cheapest beverages. The industry employs over a million persons directly or indirectly. India has been producing tea commercially for over 155 years. Most of the tea factories produce orthodox black tea, where the contents of *linalool* and *geraniol* in the black tea are balanced and the aroma is mild and fine, whereas a newly adopted technique called CTC (cut, tear and curl) had brought about a revolution by increasing the cuppage of tea. The consumer in India prefers CTC tea. The CTC adopts continuous processing rather than the batch system of orthodox method. More than 500 chemical constituents are present in tea, both volatile and non-volatile. Volatile Organic Compounds (VOCs) constitute flavour and non-volatile compounds give taste. The compounds closely related to human health are flavonoids, amino acids, vitamins, caffeine and polysaccharides. Tea also contains many essential micronutrients, such as vitamins C, B, E and K. The constituents, which contribute to the tea flavour, are mainly hexenal, hexenyl, formate, linalool oxides, pyranoid, methysalcylate, geraniol, benzylalcohol, phenylethonol and ionone.

VOCs in tea, in different ratios and amount, constitute flavour of the liquor tea. The flavour also has correlation with the quality of the tea in the market. The best tea is the most aromatic and full of good tasting flavour. The aroma and flavour of tea depends upon plucking of tealeaves at appropriate time, cutting, rolling, curling, tearing, optimum fermentation and drying at appropriate temperature etc. Packaging is also important for the preservation of flavour and aroma till the time tea is consumed.

Currently, expert human panel decisions supplemented by a few chemical tests ascertain the standard of the tea flavour. These techniques are subjective, adaptation influential and have inherent faults that may even vary with time and psychology of the panel experts at times. E-nose is not only far more efficient but also very robust and fast technique when supplemented by Intelligent Artificial

Neural Network pattern recognition techniques. To test robustness of the system for non-overlapping flavours of tea, the experiment was supported by classification of five common spice flavours. These flavours of spices are also non-overlapping by nature, so it computes an analogy to the non-overlapping tea flavour in PARC domain.

Experimental setup: Four MOS based gas sensors, TGS-2611, TGS-842, TGS-822, and TGS-813-J01 from FIGARO INC, Japan are used for developing the prototype E-nose array for the experiments to discriminate Tea and Spices flavour and aroma. The set up consists of these four sensors as E-nose array and a headspace generation and sampling system with flow control valves with suitable PC control for switching between sample headspace and base room environment. The E-noses are housed in a metal chamber to shield it from electromagnetic noises. The sensors are chosen on the basis of sensitivity to different odours such as cooking vapours, alcohol, Volatile Organic Compounds (VOCs) etc. The sensor exposure is connected through plastic pipes to the flow path junction that in turn is connected to the sample and base room environment (sample and reference vessels) through two flow control pumps, which provide constant flow of sample vapour for 3 minutes and room air for 5 minutes alternately to the sensor chamber. Alternate supply of room air to the sensors was done to refresh the sensors for next cycle.

The switching circuits are designed and converted into hardware using O/E/N 57 DP-12-1C6 coil 12 VDC 150 Ω No. 9744B as relay for the alternately switching between sample head space and reference room air. Switching time was controlled by data acquisition software - GENIE based Graphic User Interface (GUI) design, which was programmed as user customizable interface. Two flow control pumps - SO-FINE-AQUA-1 are used for supplying a constant flow of sample vapour and room air as desired by the programmed options. Data acquisition and online logging of the response from E-nose sensors is achieved by PCL-208, a High Performance Data Acquisition Card from Dynalog (India), Ltd, Mumbai-400086, India. PCL-208 card has also been used to digitally control the alternate switching of the two pumps. A logical process indicator was designed to indicate whether the E-noses are responding to room air or to sample headspace. For drift parameter determination due to temperature, humidity and pressure - three separate controlling devices suitable for our experimentation were harnessed and adopted. All data processing work was done in MATLAB 6.0 environment.

Table 1: Architecture of different Neural Networks and correct classification results in % for Overlapping flavours of tea.

Neural Network	Architecture	Classification (%)
MLP	6 input Neurons, 5 hidden Neuron, 5 output Neurons, 0.3 Learning rate with Momentum 0.4	88
LVQ	6 input Neurons, variable number of nodes in the competitive layer, 5 output Neurons, Learning rate 0.01 & 0.129 and Conscious factor 1	89
PNN	6 input Neurons, 5 Neurons in output layer, neurons added until sum-squared error falls below 0.000001, competitive output layer, Spread constant 1	94
RBF	6 input Neurons, 5 Neurons in output layer, neurons added until sum-squared error falls below 0.000001, hidden radial basis layer and linear output layer, Spread constant 1	100

Neural Networks: Pattern Recognition (PARC) is used to classify data sets acquired through a series of measurements on the samples. A matrix is formed from the patterns for a number of samples and then a decision vector that divides the pattern into an assigned classification is calculated. This is then used to classify unknown patterns. The pattern recognition methods are either supervised or non-supervised. In this experimentation, both have been considered, as they are widely used for electronic noses.

PARC techniques can be enhanced by data pre-processing such as feature selection and feature extraction. A widely used feature extraction in linear and unsupervised method is principal component analysis (PCA), which reduces dimensionality with a minimum loss of information. It partly correlates data to two or three-dimensions. This is achieved by projecting the data onto fewer dimensions, which retain maximum amount of information in the smallest number of dimensions.

The ANNs are computer programs based on a simplified model of the brain, which reproduces brain's logical operation using a collection of neuron-like elementary units to perform processing. A training set consists of two parts: a training stimulus, a collection of inputs to the perceptron and a training target, the desired output for each stimulus. Training of the ANN enables it to distinguish between two or more odourous molecules with different structures. Several types of ANN are available. Advantages of ANNs include the ability to handle noisy or missing data, no equations are involved, a network can deal with previously unseen data once training has been completed, large number of variables can be handled and they provide general solutions with good accuracy.

The data sets from E-nose response were analyzed using four ANN classifier paradigms - Learning Vector Quantization (LVQ), Multiple Layered Perceptron (MLP), Probabilistic Neural Network (PNN) and Radial Basis Function (RBF). Training of the neural networks was performed with first half of the whole data set. The remaining half of the data set was used for testing the neural networks. The aim of this comparative study was to identify the most appropriate ANN algorithm, which can be trained with high accuracy to predict the aroma of tea samples. Tables 1 and 2 are enlisted with summary of results and type of Neural Networks used for experimental training and testing of overlapping and non-overlapping tea flavours respectively.

Table 2: Architecture of different Neural Networks and correct classification results in % for non-overlapping tea flavours.

Neural Networks	Architecture	Classification (%)
MLP	4 input Neurons, 6 hidden Neuron, 10 output Neurons, 0.35 Learning rate with Momentum 0.42	87
LVQ	4 input Neurons, 10 output Neurons, Learning rate 0.011 and Conscious factor 1	92
PNN	4 input Neurons, 10 Neurons in output layer, Spread constant 0.8	93
RBF	4 input Neurons, 10 Neurons in output layer, Spread constant 0.8	96

Experimentation for Tea: The first part of the experimentations was conducted on five, overlapping tea flavours that were product of faulty tea manufacturing techniques: drier mouth, drier mouth over fired, well-fermented normal fired, well fermented over fired and under fermented normal fired. The MLP, LVQ, PNN and RBF algorithms are used to classify the samples. The results are shown in Table 1. Normalised data was used for PCA analysis. First three principal components were kept which accounted for 99.8556 % of the variance of the data set. PC #1, PC #2, PC #3, accounted for 97.9693, 1.0708, 0.8155 % of variance, respectively whereas PC #4 accounted for 0.1444 %. Five distinct tea qualities of features appear to be evident with partial overlapping plots. Multivariate data analysis suggests that there is considerable spread in the data due to drifting in the sensors' responses.

The second part of the tea experiments was conducted on 60 tea samples of 10 different non-overlapping flavours (from 6 independent sources), chosen from the tea flavour wheel that defines tea flavour terminology. Tea samples were collected from the independent sources of Noorbari and Sonabhil Tea Gardens, Golden Tea Company, Tezpur, Tea garden of Science and Technology Entrepreneurs' Park, Indian Institute of Technology, Kharagpur, Green Gold Pvt. Ltd and J. Thomas Pvt. Ltd., Guwahati, India. These flavours are bakey, sour, woody, sweet, musty, poor, smokey, minty, baggy and papery. Long series of experiments were conducted to collect a huge data for pattern recognition and analysis of tea flavour. In the experiments 40 data sets for each of the ten tea flavours resulting in a total of 400 data sets, each data set consisting of 100 data vectors were generated. In a cycle of one set of 500 data vectors (200 seconds sample reading and then 300 seconds refreshing), a data set of 100 vectors of most stable part of response is used for analysis. Thus the ratio of data collected and data used is 5:1. PCA analysis accounted for 99.06% of the variance in data set. PC #1, PC #2 and PC #3, accounted for 95.52, 2.03, 1.51 % of variance respectively whereas PC #4 accounted for 0.94 % only. Architecture and results of ANN Paradigms are enlisted in Table 2. In the second part of experiments author has achieved far distinct clustering and classification results for non-overlapping flavour terms, compared to first part of experiments which had partial overlapping clustering.

Table 3: Architecture of different Neural Networks and correct classification results in % for spice Flavours.

Neural Network	Architecture	Classification (%)
MLP	4 input Neurons, 5 hidden Neuron, 5 output Neurons, 0.35 Learning rate with Momentum 0.42	86
LVQ	4 input Neurons, variable number of nodes in the competitive layer, 5 output Neurons, Learning rate 0.011 and Conscious factor 1	93
PNN	4 input Neurons, 5 Neurons in output layer, neurons added until sum-squared error falls below 0.000001, competitive output layer, Spread constant 0 8	90
RBF	4 input Neurons, 5 Neurons in output layer, neurons added until sum- squared error falls below 0.000001, hidden radial basis layer and linear output layer, Spread constant 0 8	95

Although four sensors were used for classification of the flavour of the tea, however a study on economic use of sensors for reducing both hardware and computational cost was conducted. A method called regression selection of the most significant sensor combination by PCA analysis was done. A Root Mean Square Error (RMSE) technique and a multiple linear regression (MLR) criterion was used and the most effective sensor combination was determined.

The classification techniques suggested for tea flavour discrimination are expected to be useful for quality and flavour standardisation in the Tea Industries.

Spice Aroma Discrimination: Robustness of the E-nose was tested on five distinct spice flavours. Flavour of spices is most critical and unique for delicious cuisine. These not only make food tasty but also have therapeutic uses and medicinal values. Spices are known mostly by their aromas. This research has attempted to experimentally establish that the E-nose can distinctly classify the non-overlapping characteristics of the spice flavours.

Fifteen common local varieties of spice samples of chili (6 varieties), garlic (2 varieties), ginger (2 varieties), onion (3 varieties) and turmeric (2 varieties) were collected and tested for their aroma with the help of E-nose. Samples were first, tested at room temperature and then at elevated temperature of 100° C. The PCA analysis accounted the variance in data set to 98.72 %. PC #1, #2 and #3, accounted for 84.92, 12.33, 1.47 % of variance respectively, whereas PC #4 accounted for 1.28 % only. Classification results and architecture of MLP, LVQ, RBF and PNN are shown in Table 3.

Drift Compensation: Drift is a dynamical process, caused by physical changes in the sensors and the chemical backgrounds, which gives an unstable signal over the time. The drift could be both reversible and irreversible. The causes of the drift include variation in the surrounding in which the sensors and samples are placed. These factors are normally temperature, humidity and pressure. Another cause of drift is measurement history, i.e. measurement at time *t* is highly influenced by measurements at time *t-n*. This is called memory effect. The aging of the sensor is another cause of the drift. The drift causes pattern recognition models to be very short-lived. If drift correction is not made in the sensor signals, the models will have a continuous need for recalibration, which is a time consuming process. It becomes necessary to measure the precision of the sensors used for smelling

Gases, unlike solids and liquids have indefinite shape and volume. As a result, they are subject to pressure, volume and temperature changes. Real gas behavior is actually complex. An ideal gas is considered to be a point mass and collisions between ideal gases are assumed to be elastic. For a gas, pressure (P), volume (V), temperature (T) and the moles (n) are related by the following equation known as ideal gas equation:

$$PV = nRT$$

$$\text{or} \qquad n = PV / RT$$

Here, R is ideal gas law constant. Its value is 0.0821 L.atm/mol.K. When pressure is increased, at constant temperature and volume, the number of moles increases, thereby increasing the odourant molecule density or in other word Parts Per Million (ppm) increase. This increases E-nose response. For the temperature variation, volume is allowed to expand proportionally and naturally, and hence there is no significant change in ppm of the sample initially present in the air. However, increasing temperature can affect in two ways, first the evaporation in the sample molecules will be fast and more and second, the average kinetic energy of the sample molecules increases. This phenomenon possibly causes greater and deeper diffusion of the odourant molecules into the lattice of the sensor surface, which increases response in the sensor output. On the other hand water molecules decrease the sensor resistance and thus the E-nose output response increases. It is not known what exactly causes the humidity to increase the response in the E-nose. We assume that the polar nature of water molecules have tendency to make some weak physical bonds with the sensor surface molecules and thus increasing conductivity of the sensor surface.

Drift compensation is commonly done manually or by PC based Statistical Techniques. Many techniques have been developed using different ANN algorithms to compensate the drifts. ANN techniques are to be applied with training sessions of the network, which needs long time. In this research an effort has been made to determine drift coefficients for temperature, relative humidity variation in tea samples and pressure variations in kerosene vapour within a certain operating range of 25° C to 110° C, 90% to 40 % Relative Humidity (RH) and 103.3 kPa (1 atm) to 206.6 kPa (2 Atm) respectively. Pressure coefficients are determined for kerosene sample at higher pressure for electronic nose sensors since hydrocarbon gases relate more closely to pressure coefficient than tea in practical situations. The coefficients, subsequently, have been used to eliminate drift during online capturing and processing of data of E-nose response and analysed in PCA. The corrective coefficients (listed in Table 4), found by a long series of experiments, can be applied for calibration of MOS based electronic nose instrumentation systems according to varying ambient conditions in samples under experimental tests. The corrective factors as determined may also be incorporated onboard for the hand held electronic nose instruments, as custom based menu or corrections may be applied automatically in more advanced intelligent systems.

The results of drift determination experiments have established that there is a strong linear correlation with temperature, humidity and pressure of the samples used for classification by electronic nose system.

Table 4: Drift Coefficients for four MOS sensors.

Name of Sensor	Temperature Coefficient (mV / °C)	Humidity Coefficient (mV /% RH)	Pressure Coefficient (mV / kPa) Above STP
TGS-2611	0.20	0.89	10.3
TGS-842	0.28	1.62	15.0
TGS-822	1.30	7.59	12.4
TGS-813	0.72	2.93	14.4

The variation due to temperature of tea sample does not affect sensor response in a measurable form below the temperature of 40° C, however on increasing temperature further, we have observed that output response increases up to a temperature about 80° C. Beyond 80° C, sensor response does not vary significantly. Similarly experiments were conducted for humidity variations from 40% to 90% for tea samples. The sensor response increases as the humidity is increased. It is observed that excessive humidity may need long refreshing time for the sensor to return to the base line response. All coefficients are listed in the Table 4.

Drift compensation in E-nose sensors is expected to improve the precision in Tea testing and other process. It is assumed that the same concept can also be applied to other sample-sensor combination for the calibration such as Conducting Polymer Sensors.

Conclusion: A novel intelligent E-nose System has been developed and the classification of fifteen different flavours of tea in overlapping and non-overlapping flavour profile is tested with intelligent PARC techniques. The robustness to non-overlapping characteristics of flavour is justified by testing with five distinct spice flavors. Economical sensor combination selection by RMSE measure is adopted. Additionally to increase the precision, coefficients of drift due to temperature, humidity and pressure variation in samples is determined.

Scope for Future Work: Discrimination of Tea Flavour Specific to clone varieties and zonal varieties is a new field to be explored. FPGA implementation for hardware based and on line Tea Flavour Discrimination also seems to be a potential study area.

DECLARATION

I, Kishana Ram Kashwan, a research scholar in the Department of Electronics, School of Science and Technology, Tezpur University, hereby declare that the research work reported in the thesis entitled **"An Artificial Neural Network based Electronic-Nose System: Tea and Spice Flavour Discrimination and Drift Parameter Determination"** is a bona fide work carried out by me and being submitted to the Department of Electronics, Tezpur University, Tezpur (Assam), in the fulfillment of the requirements for the award of the degree of Doctor of Philosophy. It has previously not formed the basis for the award of any degree, diploma, associateship, fellowship or any other similar title of recognition.

Date: 17 October 2005

(Kishana Ram Kashwan)
Research Scholar
Registration No. 186 / 2000

TEZPUR UNIVERSITY

This is to certify that the thesis entitled **"An Artificial Neural Network based Electronic-Nose System: Tea and Spice Flavour Discrimination and Drift Parameter Determination"** submitted to the Tezpur University in the Department of Electronics under the School of Science and Technology in the fulfillment for the award of the degree of Doctor of Philosophy in Electronics is a record of research work carried out by Mr. **Kishana Ram Kashwan** under my personal supervision and guidance.

All helps received by him from various sources have been duly acknowledged.

No part of this thesis has been reproduced elsewhere for award of any other degree.

Date: 17 October 2005

Place: Tezpur (Assam)

Prof. (Dr.) M. Bhuyan
Supervisor
Professor
School of Science and Technology
Department of Electronics

TEZPUR UNIVERSITY

This is to certify that the thesis entitled **"An Artificial Neural Network based Electronic-Nose System: Tea and Spice Flavour Discrimination and Drift Parameter Determination"** submitted by Mr. **Kishana Ram Kashwan** to the Tezpur University in the Department of Electronics under the School of Science and Technology in the fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Electronics has been examined by us on ________________ and found to be satisfactory.

The committee recommends for the award of the degree of Doctor of Philosophy.

Signature of:

Principal Supervisor: **External Examiner:**

Associate Supervisor: *Not Applicable*

Co-Supervisor: *Not Applicable*

Date: ________________

ACKNOWLEDGEMENT

I am grateful to Prof. (Dr.) M. Bhuyan, Professor, Department of Electronics, School Of Science and Technology, Tezpur University (A Central University) for being my supervisor for this Doctoral Research work. His guidance was most significant leading path for me. At times, he provided most valuable instructions and ideas in the most friendly and frank environment to accomplish the research work. His encouraging guidance was a dynamical driving force to me to achieve the target of setting up an E-nose System at Tezpur University. The concept of E-nose was new to me and more so challenging due to very limited resources, however, Prof. Bhuyan encouraged me all the through constantly to work hard to achieve the goal. It will not be exaggerated if I say that I wouldn't have been able to achieve the goal without the supreme guidance of Prof. Bhuyan. I will remain grateful to Prof. Bhuyan forever.

I am equally indebted to Prof. B. D. Phukan, HOD & Chairman DRC, Department of Electronics. He was fatherly figure to advise and assess me. His encouraging guidance provided me with better knowledge and sense of understanding.

I am heartily grateful to Prof. J. W. Gardner and Dr. Ever L. Hines of School of Engineering, Department of Electronics, Warwick University, Coventry, UK. I had a great opportunity to learn the state of art in the field of E-nose and soft computing during my visit to the University of Warwick, UK, in 2002. The visit and subsequent discussions during it with Prof. Gardner and Dr. Hines on know how technology of E-nose were stepping stones for my Ph.D. research work. I am really thankful to Prof. M. Bhuyan, my guide, who provided me the opportunity for visiting Warwick University. The Almighty God was glad to take me U.K. in my first ever-abroad visit.

I am thankful to Dr. Singala, head of Metallurgical Department, CSIR Lab. Chandigrah, for providing me opportunity to learn about Sensor Technology. He was kind enough to provide me all facilities.

Back in the Department of Electronics, I am very much thankful to Dr. J. Dutta, Reader, Dr. S. Bhattacharyya, reader, Mr. P. P. Sahoo, Reader, Mr S. Sharma, Lecturer and Mr. S. Roy, Lecturer, Mr. Anukul C. Baishya, Lab. Supervisor, for their valuable help and cooperation. I am also thankful to Mr. Dwipendra Das for his help. *My sincere thanks to Mr. Metha Ram who helped me in workshop work at times. I am* greatly thankful to the Department of Chemical Sciences and Department of Physics, Tezpur University, for allowing me to carry out some of my work in their laboratories.

I am thankful to the authorities of Tea Gardens, Noorbari and Sonabhil, Tezpur, Golden Tea Company (Tea Trader), Tezpur, Tea garden of Science and Technology Entrepreneurs' Park, Indian Institute of Technology, Kharagpur, for providing the non-overlapping tea samples. I am greatly thankful to Green Gold Pvt. Ltd and J. Thomas Pvt. Ltd., Guwahati, India, for providing the tea flavours of different species. I am also grateful to the Tea Research Center Tocklai, Jorhat, Assam, India, for providing the specific tea samples of overlapping nature.

I am thankful to the Tezpur University Library for the wealth of the books and journals, which I referred for invaluable support of literature review. I sincerely thank the Departmental Library in the Department of Electronics for the support materials. I will remain thankful to the Electronic Workshop where I designed circuits.

I express my thanks to the Indian Air Force, my employer, for the cooperation and help. I will ever remain grateful to that Organization who provided me with right environment, which helped me achieving my goals.

Last but not the least, my sincerest thanks to all those who helped and guided me directly or indirectly.

Kishana Ram Kashwan
Research Scholar
Registration No. 186 / 2000

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AI	Artificial Intelligence
ANFIS	*Adaptive Neuro-Fuzzy Inference System*
ANN	Artificial Neural Network
ANOVA	**AN**alysis **Of** **VA**riance
API	Atomic Pressure Ionization
ART	Adaptive Resonance Theory
ATP	Andenosine Triphosphate
BAW	Bulk Acoustic Wave
CA	Cluster Analysis
CC	Component Correction
CP	Conductive Polymer
df	Degrees of Freedom
DOE	Design of Experiments
ENT	Ear, Nose and Throat
FCM	Fuzzy C-Mean
FIS	Fuzzy Inference System
FS	Fuzzy Set
GABA	Gamma Aminobutyric Acid
GC	Gas Chromatography
GRNN	Generalized Regression Neural Network
GUI	Graphical User Interface
IDT	Interdigital Transducer
IR	Infra Red
ISE	*Ion*-Selective Electrodes
LDA	Linear Discriminant Analysis
LSD	Ecstasy and Lysergic Acid
LVQ	Learning Vector Quantization
MISiC	Metal Silicon Carbide
MLP	Multi-Layer Perceptron
MLR	Multiple Linear Regression
MMI	Man Machine Interface
MOS	Metal Oxide Semiconductor
MOX	Metal Oxide
MS	Mass Spectroscopy

MVA	Multivariate Analysis
PARC	Pattern Recognition
PCA	Principal Components Analysis
PCB	Printed Circuit Board
PDP	Parallel Distributed Processing Models
PLS	Partial Least Square
PNN	Probabilistic Neural Networks
ppm	Parts per million
PTR	Proton Transfer Reaction
QMB	Quartz Crystal Micro-Balance
RH	Relative Humidity
RMSE	Root Mean Square Error
SAW	Surface Acoustic Waves
SBS	Sequential Backward Selection
SFS	Sequential Forward Selection
SPR	Surface Plasmon Resonance
SS	Sums of Squares
SSE	Sum-squared Error
STP	Standard Temperature and Pressure
TGS	Taguchi Gas Sensor
TSMR	Thickness Shear Mode Resonators
VBA	Visual Basic Application
VCT	Vacuum Cook-in-Bag/Tray Technology
VOC	Volatile Organic Compound
WOF	Warmed-over flavour

CHAPTER 1

INTRODUCTION

Chapter 1

Introduction

Scientists and researchers, for quite long, have shown tremendous interest in exploring the ways to transform some of the physiological phenomena and biological processes into machine form. This is better known as artificial intelligence (AI). The chief goal of AI is to develop paradigms (algorithms) that require machines to perform certain tasks similar to human brain. AI system must be capable of performing the three functions: (i) store knowledge, (ii) apply the knowledge strored to solve problems and (iii) acquire new knowledge through experience. An AI system has three main components: representation, reasoning and learning. E-nose is classified as one of the AI systems. The concept of biological processes implemented as artificial devices make certain tasks easier for human interpretation and understanding. Efforts have been made for the development of artificial systems reproducing the five senses of the body [1]. Recognitions of speech and analyzing of images have successfully been implemented. Olfaction in humans and animals is very important. To humans, the sensation of flavour is due to the three main chemoreceptor systems. These are the gustation, olfaction and trigeminal.

Developing artificial olfaction systems is quite complex since more analytical approach is needed for comparison and identification of odours. Classification of odours is important in industrial applications such as environmental monitoring, egg and fish freshness, banana ripening, Ear, Nose and Throat (ENT) bacteria detection, wine and beer flavour discrimination, safety and security, explosive detection and space shuttle air freshness including food quality determination and flavour discrimination. Flavour discrimination of tea and spice using AI based E-nose is the prime focus of this research.

1.1 Tea Processing and Tea Flavour

The quality of tea refers to the characters like colour, brightness, briskness and other liquor properties such as strength and flavour by which tea can be judged for its market value. Quality of tea is thus determined by the presence or absence of chemical compounds, which impart flavour characteristics in the infusion process. Majority of these chemicals are produced during processing of the tealeaves. It is, however, essential that the precursors for the compounds are adequately present in the green shoots (young tea leaves) taken to the factory for manufacturing tea. Biogenesis of such precursors is influenced, on the one hand, by the genetic and environmental factors, which cannot be controlled and, on the other hand, by the cultural practices adopted in the plantations, which can be controlled. Formation of chemicals responsible for quality from the precursors genetically available is thus influenced by the agronomic practices adopted in the fields as well as by the conditions of processing. With regard to the control of the process, parameters have to be precise, as the extent of cell rupture in the leaf, ambient humidity, temperature and oxygen concentration etc. in the processing area influence the quality of tea. Even the quality of water for cleaning the machinery, factory floor as well as on the water used for humiditification during fermentation affects the tea quality.

While the characteristics of majority of the compounds imparting flavour have been recognized only relatively recently and their linkages to typical tastes of tea have been established. Tea industry has unknowingly exploited them successfully throughout history for commercial use of the tea plant and established precise cultural practices and processing conditions. However, through application of new knowledge, it is becoming increasingly possible to optimize productivity, quality and uniformity of tea by suitably amending agronomic practices and processing conditions [2].

Due to a large number of organic compounds present in tea, it is difficult to process tea to an absolute standard. The volatile compounds present in tea determine its quality. Appendix-3 describes different volatile compounds present in black tea [2]. In conventional tasting, it is very difficult to keep a consistency in the standard of tea quality from batch to batch during the production process. The quality is ensured by a human taste panel, which may vary due to different factors. The aroma and flavour are two quality factors of tea, which depend upon the number of volatile compounds present and their ratios. .Human panel tasting is inaccurate, laborious and time consuming due to adaptation, fatigue, infection and state of mind. An E-nose can be a better alternative to the conventional methods for tea tasting and quality monitoring during production process. An E-nose is increasingly fast, reliable and robust technology. Tea industries all over the world presently use certain standard terminology of tea flavour, however, there is no mention about a quantitative description or score on these flavour terms. The Tocklai Tea Research Association, Assam (India), has adopted standard terminology but some of them overlap [3]. A tea flavour wheel [4] has been developed to indicate the tea flavour zones, which is shown in appendix A-2. In this flavour wheel twenty-five non-overlapping flavour terms have been identified out of about 40 generally used flavour notes [5].

1.2 Olfaction

The smell sensation is a chemical and neural process [6] wherein odourant molecules stimulate the olfactory receptor cells that are located high up in the nose, in the olfactory epithelium. The odourant molecules enter the human nose along with inhaled air. These molecules are received by olfactory receptor cells in epithelium and chemical signals are generated which reach to the olfactory bulb. After converging and modification, the signal travel to the human brain through nerve systems. The brain performs processing and identification operation, which subsequently results in a suitable decision for the next course of action. The sensation of flavour in humans is due to three main chemoreceptor systems. These systems are:

Gustation – sense of taste by the tongue

Olfaction – sense of smell by the nose

Trigeminal – sense of irritation by skin

Taste is used to detect non-volatile chemicals, which enter the mouth while the sense of smell is used to detect volatile compounds. Receptors for the trigeminal sense are located in the mucous membranes and in the skin. They also respond to many volatile chemicals. In the perception of flavour, all three chemo-receptor systems are involved but olfaction plays by far the greatest role with the other two senses contributing much less to our overall perception.

Odours are believed to be of two types, simple and complex. The nature of stimulus and not the quality of sensation distinguish these. A simple odour is one which consist of only one type of odourant molecule whereas a complex odour is a mixture of many, different types of odourant molecules. All naturally occurring odours are complex mixtures. Odourants are typically small hydrophobic, organic molecules containing one or two functional groups. The size, shape and polar nature of the molecules determine its odour properties. The odourant must possess certain molecular properties such as water solubility, high vapour pressure, low polarity, able to dissolve in fat (lipophilicity) and surface activity to provide sensory properties

Broad patterns of responses are shown by the mammalian olfactory system consisting of a large number of non-specific receptors, about 300 different olfactory binding proteins – in a total of about 50 million. These cells send their signals to the secondary modes and then to cells located in the olfactory bulb. There is a great deal of convergence at this stage, between 1000 to 20,000 primary receptor cells, connecting to each secondary cell followed by limited divergence [6]. This suggests that the secondary cells are involved in the integration of information, i.e. impulses from many input cells add up simultaneously. The nature of the primary cells is non-specific in their responses whereas the secondary cells respond to the distinct categories of the odours. The secondary cells interact with each other and with higher cells as well. Thus, the system is a complex and non-linear with both excitation and local inhibition helping to produce a high degree of sensitivity and specificity.

1.3 Electronic Nose

Electronic-Nose (E-nose) is an intelligent instrumentation system, which comprises of an integrated chemical sensor array, together with interfacing electronic circuitry and a pattern recognition (PARC) software paradigm [7]. It is designed to detect and discriminate among simple and complex odours. The sensory array consists of broadly tuned (non-specific) sensors that are treated with a variety of odour-sensitive biological or chemical materials. Hardware components and sensory arrays are used to collect and transport odours to the soft computing machines. Interfacing electronics circuitry is used to digitize and store the sensor responses for signal processing. An odour stimulus generates a characteristic fingerprint (or smell-print) from the sensory array. Patterns or fingerprints from known odours are used to construct a database and train a pattern recognition system so that unknown odours can subsequently be classified and identified. The functional block diagram for biological olfaction and E-nose as a comparison is depicted in figure 1.1. In the recent past, E-nose sensors have generated much interest at international level for its potential to solve a wide variety of problems in fragrance and cosmetics production, food and beverages manufacturing, chemical engineering, environmental monitoring and more recently, medical diagnostics and bioprocesses. E-nose sensors have numerous applications and potential markets in near future. In this research the E-nose has been used to discriminate the flavours of black tea and common spices.

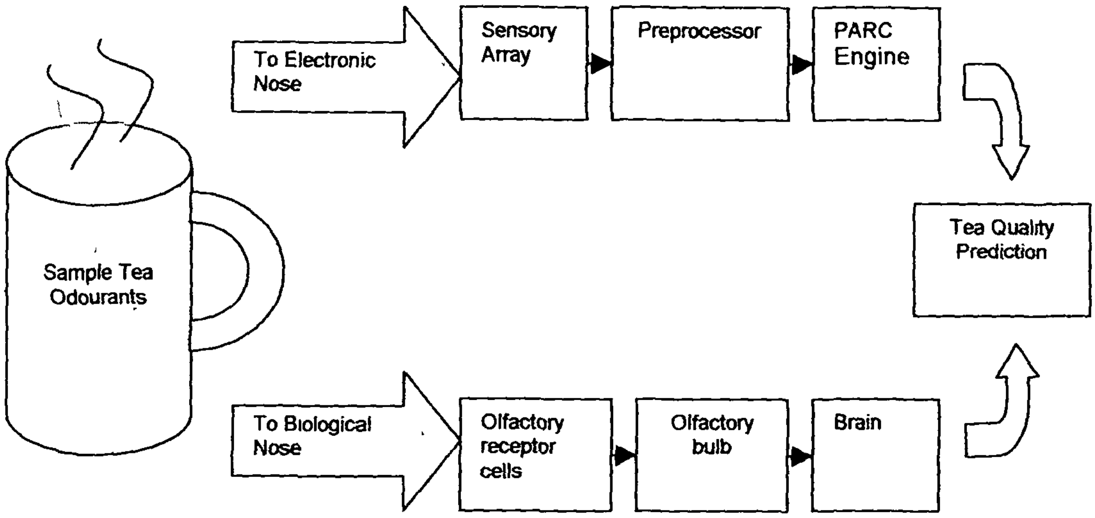


Figure 1.1 Comparison of the basic elements of an artificial olfactory system with the human olfactory system.

Array of sensors in E-nose acts as primary receptors of the odourant molecules. When odourant molecules react with sensor surface, its conductivity changes which in turn produces signals called E-nose responses. These responses are preprocessed and statistically operated by various data processing methods, which converts data in a compatible form for soft computing. Figure 1.1 illustrates the basic parts of human olfactory system and corresponding components of an electronic nose [8].

An E-nose is both a chemical sensing and a data analysis system that can discriminate among odours. The E-nose is developed as a match model for the biological nose comprising the various stages between the sensing of a volatile odourant and its recognition. Interaction, signal generation and processing of identification stages are depicted in the parallel schematic between the biological and artificial nose in figure 1.1.

When E-nose sensors are exposed to the odourant molecules, some of its properties are altered. The odourant molecules enter the lattice of the sensor surface and react chemically in a reversible reaction. The change in the properties of chemical sensor results in increased conductivity. Increase in conductivity is highly dependent upon type of odourant molecules. The variations in increase in conductivity in EN sensor enable it to generate separate electronic signature for each type of smell. Hence, E-nose is an intelligent system that can generate different patterns (electronic signature) for each type of smell. These patterns, belonging to different classes, can be classified and recognized with the help of PARC employing artificial neural networks and neural fuzzy systems.

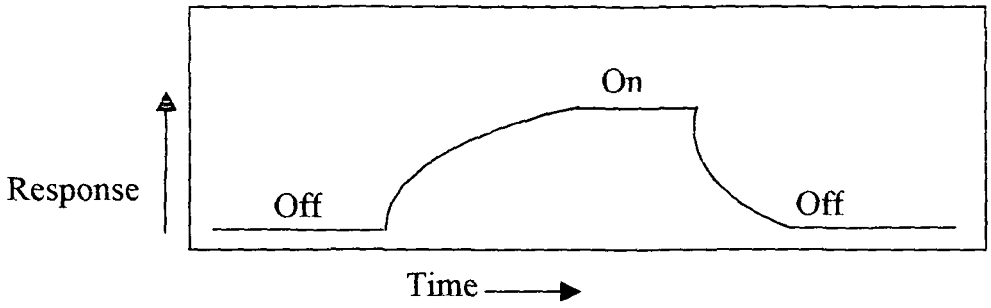


Figure 1.2 Response curve of Electronic Nose Sensor to the odourant molecules. In presence of smell, response (volt) of sensor increases, on removal of smell response reduces to initial value.

As shown in the figure 1.2, electronic-nose sensor has low response initially when it is not exposed to any smell. On exposure to smell environment its response begins to increase. Response soon becomes stable but higher than its initial value. When smell environment is removed, response begins to reduce and soon reaches its initial value [8].

Many types of materials have been developed for E-nose sensors for various applications and subsequently are used for odour detection. These include metal oxides (SnO_2), lipid layers, phthalocyanines [$(C_6H_4C_2N)_4N_4$] which is a cyclic blue-green organic pigment and conducting polymers [9].

1.4 E-nose for Tea and Spice Flavours

In conventional tasting, it is very difficult to keep the consistency in standard in tea quality from batch to batch during production process. The aroma and flavour are two quality factors of tea, which depends upon number of volatile compounds and their ratios of presence.

Indian spices are very popular and well known all over the world. Proper ripening of the spices is utmost important for a good aroma in them. Agronomical conditions such as type of soil, hybridised seeds and water irrigation and ecological conditions such as climatic changes and tropical or non-tropical etc. can have strong correlation with the quality and aroma of the spice. It has recently been proved that smells may have an important role for ailment healing. Spices are well known for the health benefits for centuries. Presently it is not known that what standard methods for determination of flavours of spice are used. Perhaps, as assumed, expert human conceptual decisions and a few chemical tests might be used to ascertain the standard of the spice flavour. These techniques have similar drawbacks as mentioned in tea tasting process. The other methods that might be used are gas chromatography (GC) and Mass Spectrometry (MS).

GC consists of a small capillary tube with an interior diameter of about 25 to 250 μm and a length of 1 to 30 meters. The molecules eluting from the column can be fragmented in a MS to produce a characteristic pattern that aids in their identification. In MS, each constituent compound molecule is ionized, typically by an electron beam, and the energy absorbed by the molecule in effect breaks it into fragments. As odourants travel down the tube, the film interacts with the mixture. The transport time-delays of the various molecular constituents of the mixture vary with their vapour pressure and solubility, causing them to separate as they emerge from the outlet. The mass spectrum that results can be used to identify the original molecule. The spectra obtained are compared to spectra contained in a large chemical database to help identification of the specific chemical compound in each tea sample.

GC and MS techniques have certain limitations and drawbacks over electronic nose sensors. These drawbacks are slowness, bulky and expensive, prone to wear and tear and short life span.

Electronic nose technology is far superior to the above mentioned techniques. EN is an intelligent system and very fast, thus likely to replace GC and MS completely in near future. An E-nose may provide a more objective platform to augment the conventional methods for tea tasting and quality monitoring during production process [10].

E-nose performs continuously in real time monitoring of an odour at specific site in the field over hours, days, weeks or even months. It can also circumvent many other problems associated with the uses of human panels.

The functional components of electronic nose operate serially on an odourant vapour from tea sample, a tea sample handler, an array of conducting semiconductor sensors and a signal processing software system. The output of electronic nose can be the identity or the characteristic properties of tea quality that human might perceive. Each sensor in E-nose has a different sensitivity. It has already been established that electronic nose identifies tea sample and estimates its quality [11]. However, spice flavour discrimination using E-nose has not been reported so far. Non-parametric analysis techniques such Principal Components Analysis (PCA) and Cluster Analysis (CA) are used to discriminate the response of an Electronic-Nose for tea and spice samples of complex odours.

1.5 Artificial Neural Networks

There are millions of very simple processing elements or neurons in the brain, linked together in a massively parallel manner. This is believed to be responsible for the human intelligence and discriminating ability. Artificial Neural Networks provide crude emulation of the brain [12]. These connectionist models implement certain important aspects of a pattern recognition system such as robustness, adaptivity, speed and learning. A neural network can learn, through examples, the discriminating characteristics of various pattern classes by automatically discovering the inherent relationship among them in a data-rich environment and no rules need to be specified beforehand. This bears an analogy to how a baby learns to recognize objects or perhaps learns to speak.

There are numerous methods adopted for the evaluation of odour data collected from E-noses. These are basically pattern recognition (PARC) paradigms. Patterns are the means by which we interpret the world. We recognize the objects around us, and move and act in relation to them. Typical examples include recognizing the voice of a friend over the phone or the flavour of tea, spice and an ice-cream, reading a newspaper, driving a car, diagnosing a disease and distinguishing a piece of music played on a *sitar* from that on a *sarod.*

Fuzzy set theory [13] tries to mimic the human reasoning and thought processes, while the neural networks attempt to emulate the architecture and information representation scheme of the human brain. These two are combined for augmenting each other in order to build more efficient intelligent information system, in neuro-fuzzy computing paradigm with having recognition performance better than those obtained by the individual technology. For example, one can have a neural classifier that can learn even with linguistic, incomplete, imprecise or vague examples. One can also have a fuzzy classifier with the capability of learning intractable classes more accurately and rapidly through neural networks.

Genetic algorithm [13], another biologically inspired tool which provides suitable adaptive, robust and fast search techniques for designing efficient pattern recognition systems through evolution based on the mechanism of natural genetics. The challenge is, therefore, to devise powerful recognition methodologies and systems by symbiotically combining these tools. The systems should have the capability of flexible information processing to deal with real life ambiguous situations and to achieve tractability, robustness and low-cost solutions.

This research is based on only sensory recognition and thus discussion is limited to the sensory recognition methods only. The problem of classification is basically one of partitioning the feature space into regions, one region for each category of input. When it is determined that an object from a sample *P* (e.g. tea) belongs to a known sub-sample *S* (e.g. woody), we say that pattern recognition is done. Classification is the process of grouping objects together into classes (sub-samples) according to their perceived likenesses or similarities. The subject area of pattern recognition includes both classification and recognition and belongs to the broader field of machine intelligence – that is, the study of how to make machines learn and reason to make decisions as do humans.

Artificial intelligence (AI) is a sub field of computer science concerned with the concepts and methods of symbolic inference by computer and symbolic knowledge representation for use in making inferences. AI can be seen as an attempt to model aspects of human thought on computers. It is also sometimes defined as trying to solve faster by computer any problem that a human can also solve. Stanford Professor John McCarthy, a leading AI researcher, coined the term.

Examples of AI problems are computer vision (building a system that can understand images similar to humans) and natural language processing (building a system that can understand and speak a human language similar to humans). Memory is the lifeline of intelligence. For AI, an associative memory is the process or device that memorizes an association between an input pattern and an output pattern. The output may be similar to the input pattern (autoàssociative), in which case the process of association can eliminate noise on the input. In other systems it may be very different (heteroassiciative)

Machine learning is an area of artificial intelligence concerned with the development of techniques which allow computers to 'learn'. More specifically, machine learning is a method for creating computer programs by the analysis of data sets. Machine learning overlaps heavily with statistics, since both fields study the analysis of data. Some machine learning systems attempt to eliminate the need for human intuition in the analysis of the data, while others adopt a collaborative approach between human and machine. Human intuition cannot be entirely eliminated since the designer of the system must specify how the data are to be represented and what mechanisms will be used to search for a characterization of the data.

Machine learning has a wide spectrum of applications including search engines, medical diagnosis, stock market analysis, classifying DNA sequences, speech and handwriting recognition, game playing and robot locomotion.

Machine learning algorithms (same as ANN) are organized into a taxonomy based on the desired outcome of the algorithm [14]. Common algorithm types include:

Supervised learning: where the algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem.

Unsupervised learning: In unsupervised learning there is no feedback path or in other words there is no teacher to oversee the learning process.

Reinforcement learning: where the algorithm learns a policy of how to act when given an observation of the world. Every action has some impact in the environment and the environment provides feedback that guides the learning algorithm.

Learning to learn: where the algorithm learns its own inductive bias based on previous experience.

The performance and computational analysis of machine learning algorithms is a branch of statistics known as computational learning theory. It will be discused later in details.

Data may be qualitative, quantitative or both; they may be numerical, linguistic, pictorial or any combination thereof [14]. Data structures in pattern recognition are of two types:

Object data: These are numerical vectors of *n* features, represented by $x=\{x_1,x_2,x_3,..,x_N\}$, a set of *N* feature vectors in the *n*-dimensional measurement space.

Relational data: These data are a set of N^2 numerical relationships, say, $\{r_{ii'}\}$, between pairs of objects. In other words, $r_{ii'}$ represents the extent to which object *i and i'* are related in the sense of some binary relationship.

In pattern recognition, features are the individual measurable properties of the phenomena being observed. Choosing discriminating and independent features is key to any pattern recognition algorithm for successful classification. Feature selection is the process of selecting a map of the form $x' = f(x)$, by which a sample $x = (x_1, x_2, ...x_n)$ in an *n*-demensional measurement space is transformed into a point $x' = x'_1, x'_2, ...x'_{n'}$ in an *n'*- dimensional feature space, where $n' < n$. While different areas of pattern recognition obviously have different features, once the features are recognized, they are classified into smaller set of algorithms. These include near neighbourhood classification in multiple dimensions, neural networks or statistical techniques, most commonly Baysian probabilities. The main objective of feature selection is to retain the optimum salient characteristics necessary for the recognition process and to reduce the dimentions of the sample space.

Classification is also a process of separating individual samples (e.g. response data from E-nose for woody or minty tea sample) from a universal sample space (e.g. response data from E-nose for 10 non-overlapping tea samples). The classification process sorts out samples to the nearest matching class or even to the absolute class of the sample according to the flexibility of algorithm rules.

The nearest neighbour algorithm in pattern recognition is a method for classifying phenomena based on observable features. In this algorithm, each feature is assigned a dimension to form a multidimensional feature space. A training set of objects with apriori known class are processed by feature extraction and plotted within the multi-dimensional feature space. The offsets in each dimension are referred to as the feature vector known as the training or learning stage. Because the engine can be retrained to classify various phenomena thus pattern recognition is a part of machine learning.The testing phase begins with phenomena to be classified (the class not being known apriori) and extracts the same set of features. The geometric distance is computed between the new feature vector and each apriori feature vector from the training set. The shortest distance thus computed is the nearest neighbour. Obviously this algorithm will be more computationally intensive as the size of the training set grows. Many optimisations have been given over the years which generally seek to reduce the number of distances actually computed. Some optimisations involve partitioning the feature space and only computing distances within specific nearby volumes where as others perform more general in nature.Other variations of the algorithm include the *N*-Nearest Neighbour algorithm where several of the nearest feature vectors are computed and the classification is made with the highest confidence only if all of the nearest neighbours are of the same class. The nearest neighbour algorithm is a heuristic algorithm.

Bayesian pattern recognition involves a similar multi-dimensional feature space but computes probabilities of the phenomena being in various classes rather than just distance from known samples [14]. In general, Bayesian classification produces somewhat better results at a higher computational cost.

The fundamental unit or building block of the artificial neural network is neuron. The general neuron has a set of n inputs x_j, where the subscript j takes values from 1 to n and indicates the source of the input signal. Each input x_j is weighted before reaching the main body of the neuron, e.g. x_j is multiplied by w_j. In addition, it has a bias term w_0, a threshold value Θ that has to be reached or exceeded for the neuron to produce a signal, a nonlinearity function F that acts on the produced signal R and an output O. In a network the neuron is called *node*. For m nodes in a network, an additional subscript, I, is needed to distinguish a single neuron. Figure 1.3 depicts the basic model of the neuron.

The transfer function of the basic model is described by the relation

$$O_i = F_i(\sum w_{ij} x_{ij})$$

The neuron firing condition is $\sum w_{ij} x_{ij} \geq O_i$ where the index *i* represents the neuron in question and *j* represents the inputs from other neurons. The purpose of the nonlinearity function is to ensure that the neuron's response is bounded. The actual response of the neuron is conditioned or damped as a result of large or small activating stimuli and thus is controllable. In biological process conditioning of stimuli is continuously done by all sensory inputs. Different nonlinearity functions [14] are used, depending on the ANN paradigms and the algorithms used. Two most popular nonlinearities are the hard limiter and the sigmoid, which are explained later.

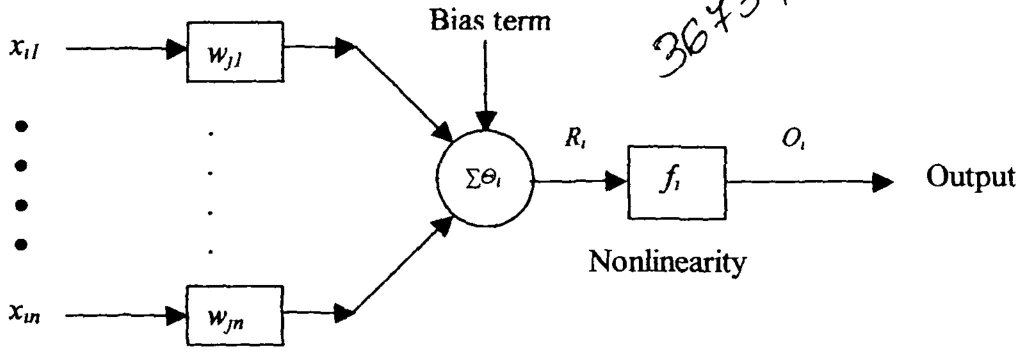


Figure 1.3 Basic neuron model

1.6 Drifts in E-nose Sensors

Drift is a dynamical process, caused by physical changes in the sensors and the chemical backgrounds, which gives an unstable signal over the time [15]. The drift could be both reversible and irreversible. The causes of the drift include variation in the surrounding in which the sensors and samples are placed. These factors are normally temperature, humidity and pressure. One another cause of drift is measurement history, called memory effects. This means that measurement at time *t* is highly influenced by measurements at time *t-n*. This leads that the same gas mixture will not give one well-defined pattern [15]. The aging of the sensors is another cause of the drift. The drift causes pattern recognition models to be very short-lived. MOS sensors for E-nose are operated at predefined temperature range due to inbuilt heating mechanism subjected to power dissipation rate, defined by manufactures. Inbuilt heating allows these sensors to operate at temperature much higher than the surrounding. Humidity and pressure of the sample gas also plays a role strictly according to the type of the sample used. However, there had always been a difficulty to analyse and eliminate the drifts due to temperature, humidity and pressure variation in the surrounding environment in which samples naturally reside i.e. for field experimentations, such tea quality monitoring during manufacturing using E-nose sensors. If drift correction is not made in the sensor signals, the models will have a continuous need for retraining.

Drift parameters of MOS based E-nose sensors due to temperature, humidity and pressure of the sample have not been adequately addressed so far. These parameters can be used as additional inputs to Neural Networks, which can compensate to certain accuracy in accordance with set algorithms for the particular sensor and sample pair. It, however, appears that drifts in sensors due to temperature, humidity and pressure are highly sample specific. It, therefore, becomes necessary to determine the nature of drifts in electronic nose sensors for a particular type of sensor and sample combination, each time an experiment is done for the given sample. It is realized that it would be useful if drifts for MOS sensors were determined specifically for tea flavours to be tested in the field and plant environments.

Increasing refreshing time duration between the sample readings might, in most of the cases, eliminate drifts related to memory effects. Aging problem in the sensors can be evaluated by the replacement of old sensors. However, principal causes of the drift are temperature, humidity and pressure. These factors are to be duly analyzed for the drift reduction. Statistical and analytical techniques are quite often suggested for the reduction in the drift, however, these methods are general in nature and lack efficiency. The four E-nose sensors, which were used in the experiments on tea and spices, were calibrated for the drift deviations. The calibration, once carried out, can be expected to be fairly constant over a long time use of the sensors for the same samples. This can effectively eliminate the burden of the frequent and long training sessions of the E-nose system [15].

1.7 Sensor Selection Criterion

The recent advances in technology generated new sensors, which are tuned to different gases of interest, although they are still non-selective. Many sensors from the same family can be used, although their responses are only slightly different [16]. The wide spectrum of sensor types is needed, but it generates a selection problem when a specific application has to be developed. Economic or technological aspects are also to be considered. Many research areas are focused on sensor technology and powerful classification algorithms development but often, the sensor selection, an intermediate step is forgotten. However, it is well known that in most of the cases, a subset of sensors provides better recognition rate than the whole set of sensors and therefore the applications can be simplified by using simpler classifiers or by using fewer sensors. In addition, every extra sensor amounts additional cost and also the fewer sensors, the less training points are needed for the classifier. Therefore, the number of sensors used for a certain application should be minimized. Hence, a systematic method for selecting sensors in order to optimize a sensor array is desired. This research work had also focused to devise a method for selecting the optimum number of sensors from a group of four MOS sensors, TGS-2611, TGS-842, TGS-822 and TGS-813-J01 from FIGARO INC, Japan [17], for discrimination of tea and spice flavour quality.

The problem of the selection of the best set of sensors arises from the fact that most of the classifiers do not provide the best result if irrelevant or redundant sensors are used [9]. This problem is usually prevented by using methods of dimensionality reduction e.g. feature extraction or feature selection. With the feature extraction method, a mathematical transformation is implemented. This transformation maps the original sensor space to a new space, usually of lower dimensionality where the performance of the system improves. Among these transformations, linear methods as Principal Component Analysis (PCA) [18] and Linear Discriminant Analysis (LDA) are the usual choice. In the case of feature selection, just a subset of features (sensors) of the original space is selected. This selection is guided by the improvement of a certain function, such as RMSE minimization or usually the correct recognition rate. Among the different algorithms for feature selection two main groups, linear projection and search methods have been considered [19]. From the sensors' data, the steadiest part of 4 sensors' response was selected for the following analysis methods.

Projection methods: Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) have been used to map the n-dimensional sensors space onto 2-dimensional feature space. New space sensor selection can be achieved by using the rule that the features with minimum loading must be eliminated. PCA use eigenvectors and eigenvalues to define the subspace orthogonal base from the data covariance, trying to preserve the variance presented by the original N dimensional data into subspace.

Search methods: In the search methods the maximization or minimization of an objective function guides the selection. The objective function used for the problem presented above is the root mean square error (RMSE) [20]. In principle, an exhaustive search of the best result can be done, but the number of combinations to evaluate is 2^N, where N is the number of features. If N is high, this exhaustive search is impossible in practice. In the case exposed, we may use exhaustive search as a reference procedure to check the performance of the different algorithms. Among the all possibilities, the best result is a configuration with only fewer sensors that present a recognition rate fairly high. If we order the all-possible subsets according the recognition rate, the configuration that uses all sensors is at position far from the best. At this point it must be highlighted that the best recognition rate only reaches a certain value and this value has to be taken as reference for the other algorithms. The recognition rate depends strongly with the problem and with the classifier used.

Sequential forward and sequential backward selection: Among the different search algorithms, the Classical Sequential Forward Selection (SFS) and the Classical Sequential Backward Selection (SBS) are appreciable. With SFS the selection algorithm starts from the empty space, and a feature is added if it provides the best improvement in the objective function. SBS uses the opposite strategy i.e. the algorithm begins from the whole feature space and a feature is extracted if it provides the less descent in the objective function.

1.8 Outline of the Thesis

The thesis is organized into 5 chapters and 7 appendices. Chapter-1 and 2 describe brief introduction on the research work and review of comprehensive literature. In these chapters, E-nose and biological olfaction have been described as a comparative study and merits of E-nose over olfaction are presented. A detailed review of ANN and artificial intelligence literature has been included. Introduction of fuzzy system and regression method is introduced.

Chapter-3 has bee described with the complete and detailed procedure for design and development of E-nose setup. Brief history of E-nose followed by technology involved has been narrated. Types of sensors used for E-nose are explored and applications of E-nose are described. E-nose is represented as an intelligent instrumentation system with procedural description of design and development including materials, Printed Circuit Boards (PCBs) and PC interfacing. E-nose data handling and data acquisition are described including feature selection and graphical user interface (GUI) procedure. Lastly the drift problem in E-nose sensors is discussed and the chapter is concluded with results, conclusion and reference list.

Chapter-4 four is devoted to the tea and spice flavour discrimination. Sample collection, testing procedure, analysis of data collected from E-nose sensors using ANN techniques are described and presented illustriously. The robustness of the system is checked by using additional flavour samples of common spices. ANN analysis is presented for the spice flavour classification and quality determination. Lastly optimum sensor selection criterion is discussed and RMSE and MLR techniques are used to select best possible combination of the sensors. At the end results are discussed and concluded with reference list.

Chapter-5 is presented with detailed procedure of determination of the drift parameters. Causes and remedial actions for various types of drifts due to temperature, humidity and pressure are described. Validation of the drift is carried out by implementing compensation for the drifts in accordance with drift coefficients determined. It is proved that drift compensation is helpful to improve the flavour classification rate and quality determination. Results, conclusion and reference list are presented at the end of the chapter.

At the end, 7 appendices are included to describe tea flavour and neural network glossary, tea flavour wheel, chemical composition of the tea, Leading E-nose manufacturers list, data acquisition card and sensors specification. Lastly source codes in MATLAB for programs for PCA and ANN classification paradigms are included in appendix-7.

List of references is included at the end of each of the chapters.

1.9 Conclusion

The electronic-nose is an intelligent instrumentation system comprising an array of electronic chemical sensors with partial specificity and a Neural Network based pattern recognition intelligent system that is capable of recognizing both simple and complex odours. This implies that electronic nose is a kind of intelligent sensor system that is used to sense odourant molecules in an analogous manner of the human nose. The Tea Industry is expected to benefit from this proposed artificial neural network based E-nose sensor system, which can be used to predict the quality of the tea. It will make possible for an improved testing procedure compared to old techniques [21, 22] to determine flavour quality of tea grades. Implementation of automation and intelligent system will enable the Tea Industry to be independent of human constraints and limitations.

References

[1]. Stamtios V. Kartalopoulos, *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications,* IEEE Press, PHI, 2000.

[2]. N. K. Jain, *Global Advances in Tea Science*, Aravali Books Intc. (P) Ltd, New Delhi, 1999

[3]. Ritaban Dutta, E. L. Hines, J. W. Gardner K. R. Kashwan, M. Bhuyan, *Tea quality prediction using a tin oxide based electronic nose*: Elsevier Science Journal: Sensors and Actuators B 94 (2003) Volume 94, Issue 2 (1 September 2003) Pages: 228-237. Elsevier Science Ltd. Oxford, UK.

[4]. M. Bhuyan and S. Borah, *Use of Electronic Nose in Tea Industry,* EAIT, 2001, IIT, Kharagpore.

[5]. Ritaban Dutta, K. R. Kashwan, M. Bhuyan, E. L. Hines, J. W. Gardner *Electronic Nose based tea quality standardization,* Neural Networks Volume 16, Issue 5-6 (June 2003) 2003 Special issue: Advances in neural networks research IJCNN'03, Elsevier Science Ltd, UK

[6]. D.G. Laing, R. L. Doty and W. Breiohl, *The Human Senses of Smell,* New York: Springer-Verilag, 1991

[7]. K. R. Kashwan, M.Bhuyan, *Tea Flavour Discrimination by using Electronic-Nose Sensors and Artificial Neural Network Pattern Recognition Techniques,*

BIOMED-2005, National Conference on Emerging Trends in Biomedical Instrumentation, Birla Institute of Technology, Mesra, Ranchi, Apr 29-30 2005

[8]. Persaud K.C. and Pelosi P., *An approach to an electronic nose: Artificial Internal Organs,* 31: 297 300

[9]. Prasanna Chandrasekhar, *Conducting Polymers, fundamentals and Applications: A Practical Approach,* Kluwer Academic Publishers © 1999.

[10]. Philip N. Bartlett, Joe M. Elliott, and Julian W. Gardner, *Electronic nose and their application in the Food Industry,* FOOD TECHNOLOGY, Dec 1997, Vol. 51, No. 12

[11]. Ritaban Dutta, E. L. Hines, J. W. Gardner K. R. Kashwan, M. Bhuyan, *Determination of tea quality by using a Neural Network based Electronic Nose,* International Joint Conference on Neural Network (IJCNN 2003), Portland, Oregon, USA. Conference sponsored by IEEE Neural Network Society and International Neural network Society, July 20-24 Vol. 1, pp.404-409, 2003

[12]. J. W. Gardner, E. L. Hines and M. Wilkinson, *Application of artificial neural networks to an electronic olfactory system,* Measurement Science and Technology, 1990, 1, 446-451

[13]. Sankar K. Pal and Sushmita Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing,* Wiley Series on Intelligent Systems, John Wiley & Sons, Inc.

[14]. Simon Haykin, *Neural Networks: A Comprehensive Foundation* Second Edition, Pearson Education Asia 1999.

[15]. K. R. Kashwan, M.Bhuyan, *Determination of Drift due to Temperature, Humidity and Pressure in the Samples for MOS based Electronic-Nose Sensors,* BIOMED-2005, National Conference on Emerging Trends in Biomedical Instrumentation, Birla Institute of Technology, Mesra, Ranchi, Apr 29-30 2005

[16]. J. W. Gardner and P. N. Bartlett *A brief history of electronic nose* Sensors and Actuators B, 1994, 18-19, 211-220.

[17]. Production Information *Tin Oxide Sensors* Figaro Engineering Inc. (Japan), Users' guide.

[18].. R. Dutta, E. L. Hines, J. W.Gardner, D. D. Udrea, P. Biolot, *Non-destructive egg freshness determination: an electronic nose based approach,* Measure. Science Technol. (IoP),14 (2003) P: 190 – 198

[19]. T.C. Pearce, J. W. Gardner, S. Friel, P. N. Bartlett, N. Blan, *Electronic nose for monitoring the flavour of beers,* Analyst118 (April) (1993)

[20]. A. N. Choudry, T. M. Hawkins and P. J. Travers, *A Method for Selecting an Optimum Sensor Array,* Elsevier Science Journal: Sensors and Actuators B 69 (2000) Volume 69, Pages: 236 - 242. Elsevier Science Ltd. Oxford, UK.

[21]. M. Bhuyan, *An Integrated PC based Tea Process Monitoring and control System* Ph.D. Thesis, Guwahati University, October 1997.

[22]. S. Borah and M. Bhuyan *Non-destructive testing of tea fermentation using image processing* Insight (Journal of British Institute of Non-destructive Testing) Vol. 45, No. 1, January 2003.

§§

CHAPTER 2

OLFACTION AND ARTIFICIAL INTELLIGENCE

Chapter 2

Olfaction and Artificial Intelligence

Smell and taste are generally classified as visceral senses because of their close association with gastrointestinal function. Physiologically, they are related to each other. The flavours of various foods are in large part a combination of their taste and smell. Consequently, food may taste different if one has a cold that depresses the sense of smell. Both taste and smell receptors are chemoreceptors that are stimulated by molecules in solution in mucus in the nose and saliva in the mouth. However, these two senses are quite different. The smell receptors are distance receptors (teleceptors). The taste pathways pass up the brain stem to the thalamus and project to the postcentral gyrus along with those for touch and pressure sensibility from the mouth. The olfactory sense is able to distinguish among a practically high number of chemical compounds at very low concentrations [1].

Research on artificial neural networks, commonly referred to as *neural networks*, has been motivated right from its beginning by the recognition that the human brain computes exclusively in a different way from the conventional digital computer. The brain is highly complex, nonlinear and parallel computer (information processing system). It has the capability to organize its structural constituents, known as neurons, so as to perform certain computation, e.g. pattern recognition, perception and motor control etc, many times faster than the fastest digital computer in existence today [2]. For example, the brain accomplishes perceptual recognition tasks (e.g. recognizing a familiar face embedded in an unfamiliar scene) in approximately 100-200 ms, whereas tasks of much lesser complexity may take days on a conventional computer.

In case of *sonar* of a bat, the complex neural computations needed to extract all this information from the target echo, occur within a brain the size of a plum. Indeed an echolocating bat can pursue and capture its target with a facility and success rate that would be the envy of radar or sonar engineer. At birth, a brain has great structure and the ability to build up its own rules through what we usually refer to as 'experience'. Indeed, experience is built up over time, with the most dramatic development (i.e. hard wiring) of the human brain that takes place during the first two years from birth; but the development continues well beyond that stage. A "developing" neuron is synonymous with a plastic brain: *Plasticity* permits the developing nervous system to adapt to its surrounding environment.

2.1 The Olfactory System

The sense of smell is a primal sense for humans as well as animals. From an evolutionary point of view, it is one of the most ancient of senses. Smell (Olfaction) allows vertebrates and other organisms with olfactory receptors to identify food, mates and predators provide both sensual pleasure (the odour of flowers and perfume) as well as warnings of danger (e.g. spoiled food and chemical dangers etc.). For both humans and animals, it is one of the important means by which our environment communicates with us. Dogs and horses can smell fear in humans [3]. Grammer, in Vienna, has recently demonstrated that the smell of fear can be detected by women in the armpit secretions of people who watched a terrifying film [4]. The implication of this work is that a chemical signal is secreted in sweat that communicates the emotion. Smells are detected in the nose by the specialized receptor cells of the olfactory epithelium. These are called olfactory receptor neurons.

2.1.1 General Physiology of Olfaction

Odourants are volatile chemical compounds that are carried by inhaled air to the regio olfactoria (olfactory epithelium) located in the roof of the two nasal cavities of the human nose, just below and between the eyes (figure 2.1).

The olfactory region of each of the two nasal passages in humans is a small area of about 2.5 square centimeters containing in total approximately 50 million primary sensory receptor cells. The olfactory region consists of cilia projecting down out of the olfactory epithelium into a layer of mucous, which is about 60 microns thick [1], as depicted in figure 2.2. This mucous layer is a lipid-rich secretion that covers the surface of the receptors at the epithelium surface. The mucous layer is produced by the Bowman's glands, which reside in the olfactory epithelium. The mucous lipids assist in transporting the odourant molecules as only volatile materials that are soluble in the mucous can interact with the olfactory receptors and produce the signals that our brain interprets as odours [5, 6, 7]. Each olfactory receptor neuron has 8-20 cilia that are whip-like extensions 30-200 microns in length. The olfactory cilia (figure 2.2) are the sites where molecular reception with the odourant occurs and sensory transduction (i.e. transmission) starts [8].

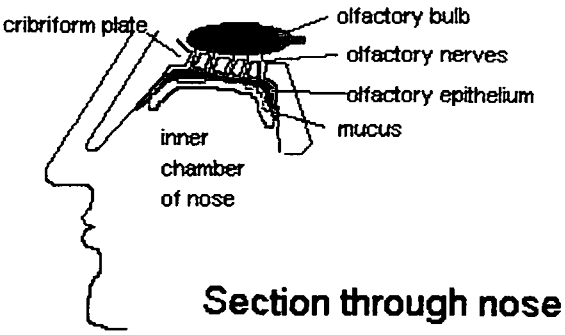


Figure 2.1 Location of nasal cavity

Above the mucous layer is the base olfactory epithelium which consists partially of basal cells located in the lowest cellular layer of the olfactory epithelium that are capable of mitotic cell division to form olfactory receptor neurons when functionally mature. The epithelium also contains pigmented cells that are light yellow in humans and dark yellow to brown in dogs. The depth of color seems to be correlated with olfactory sensitivity. While the olfactory receptor neurons extend through the epithelium to contact odourants in the atmosphere, on the opposite side within the epithelium, the neuronal cells form axons that are bundled in groups of 10-100 to penetrate the ethmoidal cribriform plate of bone, reaching the olfactory bulb where they converge to terminate with post-synaptic cells to form synaptic structures called glomeruli. The glomeruli are connected in groups that converge into mitral cells. For example, in rabbits, there are 26,000 receptor neurons converging onto 200 glomeruli, which then converge at 25:1 onto each mitral cell. The total convergence is estimated to be about 1000:1. Physiologically, this convergence increases the sensitivity of the olfactory signal sent to the brain. From the mitral cells the message is sent directly to the higher levels of the central nervous system in the brain (via the olfactory nerve tract) where the signaling process is decoded and olfactory interpretation and response occurs [8, 9].

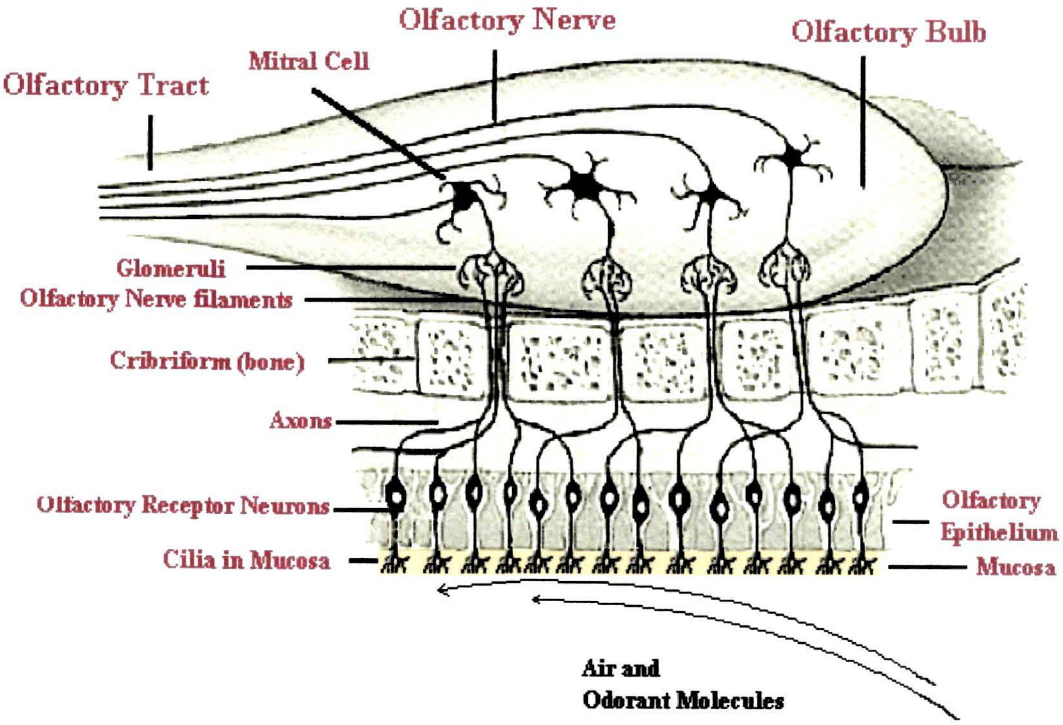


Figure 2.2 Structure of Human Olfaction System

Two American scientists who solved the enigma of how people can smell 10,000 different odours and recall them later were awarded the 2004 Nobel Prize in Physiology / Medicine [10]. The winners are Dr. Richard Axel, 58, a professor at Columbia University, and Dr. Linda B. Buck, 57, of the Fred Hutchinson Cancer Research Center and a professor at the University of Washington in Seattle. The scientists discovered a large gene family, made up of 1,000 different genes [11]. The olfactory genes give rise to an equivalent number of olfactory receptors located on 5 million cells in a small area in the upper part of the nostrils. The cells are highly specialized to detect molecules of a few inhaled odors. The number of odorant receptors varies among species. For their studies, Dr. Axel and Dr. Buck used mice, which have about 1,000 odorant receptors. Humans have about 350 such receptors [12].

2.1.2 Odourant Receptors

In 1991, Linda Buck [12] and Richard Axel [13] both discovered the family of transmembrane proteins that were believed to be the odour receptors and some of the genes that encode them. For this work both, Buck and Axel won the Nobel Prize in 2004. They cloned and characterized 18 different members of an extremely large multi-gene family that encodes the seven transmembrane proteins whose expression were restricted to the olfactory epithelium. This was a decisive breakthrough in potential understanding of the olfactory system [14]. The proteins found with all helical transmembrane structure contained sequence similarity to other members of the *G-protein* linked receptor family. It is now known pseudogenes (only 40% genes are functional in humans, the rest are known as pseudogenes) that there are about 350 odourant receptor genes and about 560 odourant receptor in humans [9, 15, 16, 17]. In 1998, Firestein [18] and coworkers at Columbia University effectively demonstrated that genes coded to produce olfactory receptors could be inserted into the rat olfactory system and that specific odourant chemicals would generate significantly higher signaling as measured by the electrical activity in the neurons. He monitored the electrical activity in the neurons, producing a chart called an electro-olfactogram. Electrical activity was highest when the nerve cells were exposed to octanal, an aldehyde that smells meaty to humans [18]. Note that most flavourists & perfumers would describe octanal as being fatty-fruity with citrus-orange notes. In this work researchers evaluated 74 odourants on a specific odour receptor. A long-standing question as to whether individual receptors recognize multiple odourants, or do single neurons have multiple receptors now appears clarified [19]. It appears that to reconcile the ability of organisms to detect far more than 1,000 discrete odours, the odours must participate in some kind of combinatorial processing: i.e., one receptor must be able to interact with several discrete odourants. Conversely, an odour molecule must be capable of interacting with multiple receptors. By inference, an individual odour will activate multiple glomeruli in the olfactory bulb. Firestein's findings may be summed up as follows:

(i) Each olfactory neuron expresses only a single type of receptor.

(ii) Gene probes for a single type of receptor bind to only 1 in a 1000 sensory neurons in a normal olfactory epithelium.

(iii) However, rats made to express a single type of receptor in large numbers of their olfactory neurons responded much more vigorously to a single type of odourant than to any of the other 73 tested.

(iv) Cells taken from these rats and placed in tissue culture also responded to only that one type of odourant molecule.

(v) Each receptor is probably capable of binding to several different odourants, some more tightly than others. The cells described above also responded although more weakly.

(vi) Each odourant is capable of binding to several different receptors.

This provides the basis for *combinatorial diversity*. This can be explained by assuming that odourant A binds to receptors on neurons #1, #2 and #3 and odourant B binds to receptors on neurons #2, #4 and #5. The brain then would interpret the two different patterns of impulses as separate odours. This mechanism is capable of discriminating among millions of different odourants. Recently, Doron Lancet [16] and co-workers at the Weizmann Institute of Science Crown Human Genome Center have constructed a database of human olfactory receptor (OR) genes by a highly automated data mining system [15, 20, 21]. This is a non-redundant dataset, which includes 906 human olfactory receptor genes.

G-proteins are signaling machines that transduce messages from receptors from extra-cellular stimuli into cellular responses mediated by effectors (enzymes) or ion channels. G-protein coupled receptors are known to have been present in the acoelomate flatworms of the Precambrian era over 800 million years ago. This ancient organism (believed to be the ancestor of all bilaterally symmetric metazoans) contained a surprisingly rich collection of cellular signaling mechanisms. [17]. G-protein-coupled receptors are a pharmacologically important protein family with approximately 450 genes identified to date. The olfactory receptors are one of the largest groups of G-protein coupled receptors described to date. Olfactory G-protein linked receptors trigger the biochemical synthesis of neurotransmitters, which open cation channels, which ultimately leads to action potentials and signaling [22].

2.1.3 Theories of olfaction

(i) Molecular shape: Chemists noted that chains of certain aldehydes / alcohols had strong odours and the odour changed as the chain length increased. Benzene ring altered its smell greatly according to where the side chains were situated. In 1952, Amoore suggested that there were seven primary classes of odours. The basic idea was that there were seven kinds of olfactory receptor site, places whereon odourant molecules could lodge when adsorbed on the olfactory sensitive area, and that the odourous molecules had shapes and sizes that were complementary to the shape and size of the seven olfactory receptors.

(ii) Diffusion pore: This theory of Davies and Taylor (1959) suggests that the olfactory molecule diffuses across the membrane of the receptor cell forming an ion pore in its wake. The diffusion time and affinity for the membrane receptor determine thresholds. But, it is difficult to explain the different qualities of smell. The problem of frequency coding and stimulus intensity is difficult to resolve. The different odour would cause a different size pore and therefore a different receptor potential, giving rise to a particular firing rate - but in olfaction, stimulus intensity is frequency coded and not the different quality of the odour. However, many odourants are organic molecules that will dissolve in membranes and alter their properties.

(iii) Piezo effect: Rosenberg et al (1968) proposed this theory. They believed that the carotenoids (vitamin A, in the pigment of the olfactory cells) combine with the odourous gases giving rise to a semiconductor current. They argued that this current could activate the olfactory neurons. This theory followed from an earlier paper by Briggs and Duncan who proposed "it seems reasonable to assume that protein bound carotenoids of the olfactory epithelium are the receptors of energy from olfactant molecules entering the nasal cavity". Rosenberg and colleagues tested the idea and found a reversible concentration-dependent increase in current of up to 10,000,000 times and proposed a weak-bond complex formation which increased the number of charge carriers. However, there were problems with this theory; (1) receptor cells do not contain the pigment and (2) weakly odourous short chain alcohols gave a greater increase in semiconductor current than smellier long-chain alcohols

(iv) Molecular vibration: The frequency of many odours is in the range of infrared (IR) spectrum. This resonance is associated with the odour, which was suggested by Dyson (1938). Male moths are drawn to candles because the flickery IR emission is identical to that of the female moth's pheromone. Different frequencies of IR could give rise to different smells. If the whole vibrational range were used, up to 4000 cm^{-1}, the detection of functional groups would be explained since many compounds with distinctive odours vibrate at around 1000 cm^{-1}. The problem is that frequency coding is proportional to stimulus intensity in olfaction, so different frequencies of IR could not be converted into different nerve firing frequency.

(v) The nose as a spectroscope: This theory, proposed by Luca Turin (1996), originates from the work of Dyson who suggested that the olfactory organs might detect molecular vibrations. Turin has proposed that when the olfactory receptor protein binds an odourant, electron tunneling can occur across the binding site if the vibrational mode equals the energy gap between filled and empty electron levels. The electron tunneling then activates a G-protein cascade. Receptors are therefore tuned to the vibrational frequency of particular odourants, rather like cones are tuned to particular wavelengths of light.

2.2 Biological Neuron

If an animal has a backbone it also has a brain. Dogs, lizards, frogs, fish and even birds have brains. But none of these creatures demonstrate the same capacity for learning, language, emotion and abstract thought that distinguishes the human species. Weighing in at around 1.5 kg [2], brain is one of the largest organs in the human body. It consists of a complex and utterly tangled nerve cells or neurons. It is housed inside the skull immersed in a fluid that cushions it from sudden impacts to the head.

The human system may be viewed as a three-stage system, as depicted in figure 2.3. Central to the system is the brain, represented by the neural (nerve) net, which continually receives information, perceives it and makes appropriate decisions. The arrows are, pointing from left to right indicate the forward transmission of information-bearing signals through the system and the arrows pointing from right to left signify the presence of feed-back in the system. The receptors convert stimuli from the human body of the external environment into electrical impulses that convey information to the neural net (brain). The effectors convert electrical impulses generated by the neural net into discernible responses as system outputs.

Ramon Cajal introduced the idea of neurons as structural constituents of the brain. Typically, neurons are five to six orders of magnitude slower than silicon logic gates [2]. However, the brain makes up for the relatively slow rate of operation of a neuron by having a truly staggering number of neurons with massive interconnections between them. It is estimated that there are approximately 10 million neurons in the human cortex and 60 trillion synapses [2]. The net result is that the brain is an enormously efficient structure. Specifically, the energetic efficiency of the brain is approximately 10^{-16} joules per operation per second compared to computer, which has about 10^{-6} joules (about 10^{10} times more than brain) [23]. Since neuron is the building block of the brain, it is presented in detail in next section.

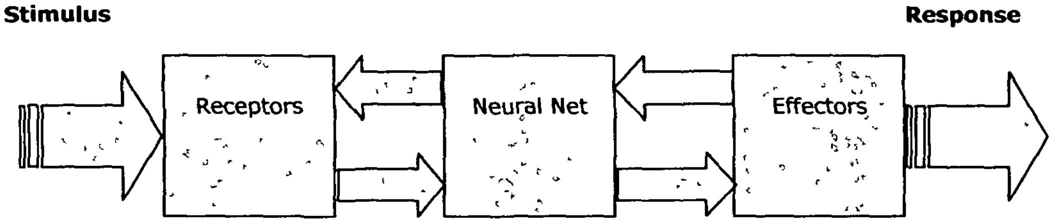


Figure 2.3 Block diagram representation of nervous system.

2.2.1 The Physiology of Neuron

The brain is a collection of large number of interconnected neurons. Each neuron is a cell (figure 2.4) that uses biochemical reactions to receive process and transmit information. A neuron's dendritic tree is connected to a thousand neighbouring neurons. If any one of the neurons fires, a positive or negative charge is received by one of the dendrites. The strengths of all the received charges are added together through the processes of spatial and temporal summation. Spatial summation occurs when several weak signals are converted into a single large one, while temporal summation converts a rapid series of weak pulses from one source into one large signal. The aggregate input is then passed to the soma (cell body). The primary function of soma is to perform the continuous maintenance required to keep the neuron functional. If the aggregate input is greater than the axon hillock's threshold value, then the neuron *fires*, and an output signal is transmitted down the axon. The strength of the output is constant, regardless of the input signal value. The output strength reaches each terminal button with the same intensity it had at the axon hillock. This uniformity is critical in an analogue device such as a brain where small errors can grow quickly and error correction is difficult in brain than in a digital system. Each terminal button is connected to other neurons across a small gap called a 'synapse'. The physical and neurochemical characteristics of each synapse determine the strength and polarity of the new input signal.

The brain is the most flexible and the most vulnerable. Changing the constitution of various neurotransmitter chemicals can increase or decrease (excitatory or inhibitory) the amount of stimulation that the firing axon imparts on the neighbouring dendrite. Many drugs such as alcohol and LSD (Ecstasy and Lysergic Acid) have dramatic effects on the production or destruction of these critical chemicals. Once a neuron fires, it keeps on triggering all the neurons in the vicinity.

The human nervous system consists of billions of nerve cells (or neurons) in addition to supporting (neuroglial) cells. Neurons are able to respond to stimuli (such as touch, sound, light, and so on), conduct impulses and communicate with each other (and with other types of cells like muscle cells) through a membrane potential established across the cell membrane. In other words, there is an unequal distribution of ions (charged atoms) on the two sides of a nerve cell membrane. A resting potential (about –70 mV) occurs when a membrane is not being stimulated, in other words, it's resting.

An action potential is a very rapid change in membrane potential that occurs when a nerve cell membrane is stimulated. Specifically, the membrane potential goes from the resting potential (typically-70 mV) to some positive value (typically about +30 mV) [23] in a very short period of just a few milliseconds.

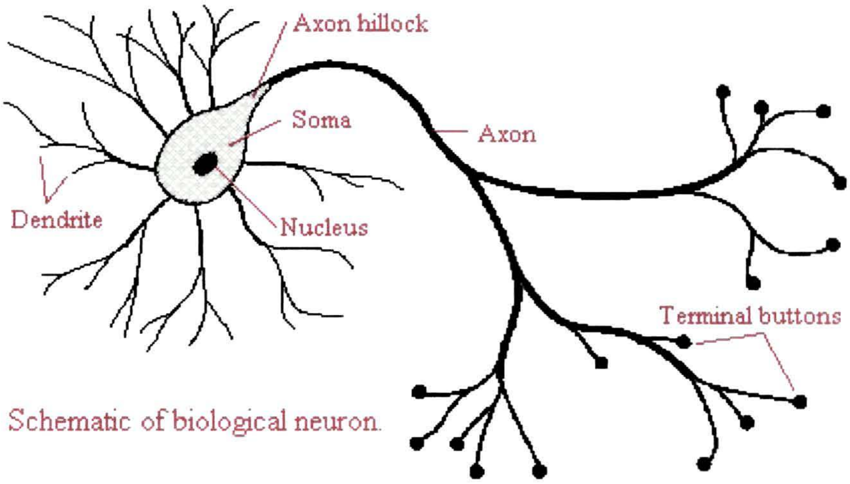


Figure 2.4 Structure of biological Neuron.

The stimulus causes the sodium gates (or channels) to open and because there's more sodium on the outside of the membrane, it diffuses rapidly into the nerve cell. All these positively charged sodium ions rushing in causes the membrane potential to become positive (inside). The sodium channels open only briefly and then close again. The potassium channels then open and because there is more potassium inside the membrane than outside, positively charged potassium ions diffuse out. As these positive ions go out, the inside of the membrane once again becomes negative.

2.2.2 Synapse

It is a point of impulse transmission between neurons. Impulses are transmitted from pre-synaptic neurons to post-synaptic neurons (figure 2.5). Synapses usually occur between the axon of a pre-synaptic neuron and a dendrite of a post-synaptic neuron. At a synapse, the end of the axon is 'swollen' and referred to as synaptic knob. Synaptic knob has many synaptic vesicles that contain neurotransmitter chemicals and mitochondria that provide Andenosine Triphosphate (ATP) to make more neurotransmitter. The gap between synaptic knob and the dendrite of the post-synaptic neuron is referred as the synaptic cleft. Pre and post-synaptic membranes, therefore, do not come in contact and hence the impulse cannot be transmitted directly. It is transmitted by the release of chemicals called chemical transmitters or neurotransmitters.

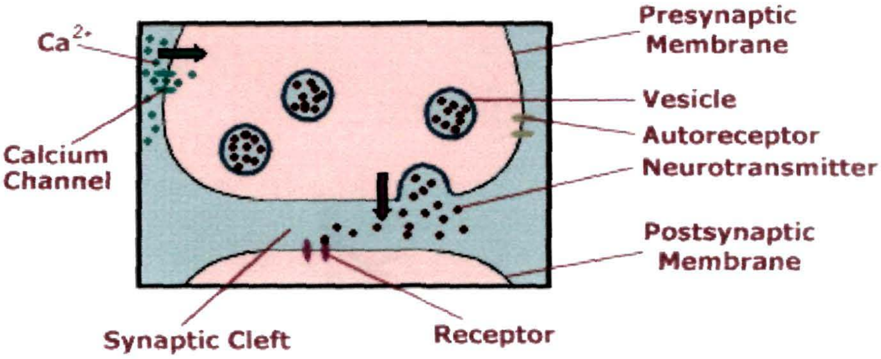


Figure 2.5 Synapse Process at cleft.

When an impulse arrives at the synaptic knob, membrane becomes more permeable to calcium. Calcium diffuses into the synaptic knob and activates enzymes that cause the synaptic vesicles to move toward the synaptic cleft (figure 2.5). Some vesicles fuse with the membrane and release their neurotransmitter e.g. *exocytosis*. The neurotransmitter molecules diffuse across the cleft and enter the receptor sites in the postsynaptic membrane. When these sites are filled, sodium channels (also called chemically gated ion channels) open and permit an inward diffusion of sodium ions as illustrated by figure 2.5. This, of course, causes the membrane potential to become less negative or in other words, it approaches the threshold potential. If enough neurotransmitter is released and sodium channels are opened then the membrane potential will reach threshold. If so, an action potential occurs and spreads along the membrane of the post-synaptic neuron or in other words, the impulse will be transmitted. Of course, if insufficient neurotransmitter is released, the impulse will not be transmitted. Impulses typically travel along neurons at a speed of anywhere from 1 to 120 meters per second. The speed of conduction can be influenced by the diameter of a fiber, temperature, and the presence or absence of *myelin*.

2.2.3 Neurotransmitters

Movement of chemicals accomplishes communication of information between neurons across synapse [24]. Chemicals or neurotransmitters are released from one neuron at the presynaptic nerve terminal and then cross the synapse to enter the next neuron at a specialized site called a receptor. The action that follows activation of a receptor site may be either depolarization (an excitatory postsynaptic potential) or hyperpolarization (an inhibitory postsynaptic potential). A depolarization makes it more likely that an action potential will fire; a hyperpolarization makes it less likely that an action potential will fire.

Neuroscientists have set up certain criteria for a chemical to be a neurotransmitter:

- The chemical must be found or produced within a neuron.
- When a neuron is stimulated (depolarized), it must release the chemical.
- When a chemical is released, it must act on a post-synaptic receptor and cause a biological effect.
- After a chemical is released, it must be inactivated. Inactivation can be through an enzyme that stops the action of the chemical.
- If the chemical is applied on the post-synaptic membrane, it should have the same effect as when it is released by a neuron.

Neurotransmitters are made in the cell body of the neuron and then transported down the axon to the axon terminal. Neurotransmitters are stored in small packages called vesicles (figure 2.5) and are released from the axon terminal when their vesicles 'fuse' with the membrane, spilling the neurotransmitter into the synaptic cleft. Neurotransmitters will bind only to specific receptors on the postsynaptic membrane that recognize them.

Otto Loewi discovered the first neurotransmitter hypothesized that electrical stimulation of the vagus nerve released a chemical neurotransmitter called acetylcholine [24].

There are two types of neurotransmitters [2], according to their nature of generating action potential across the membrane.

(i) Excitatory - Neurotransmitters that make membrane potential less negative (via increased permeability of the membrane to sodium), therefore, tend to 'excite' or stimulate the postsynaptic membrane

(ii) Inhibitory - Neurotransmitters that make membrane potential more negative (via increased permeability of the membrane to potassium), therefore, tend to 'inhibit' (or make less likely) the transmission of an impulse. One example of an inhibitory neurotransmitter is gamma aminobutyric acid (GABA) [25].

2.3 Artificial Intelligence

Artificial intelligence is the development of paradigms or algorithms that require machine to perform cognitive tasks similar to humans. An AI system has three key components: representation, reasoning and learning, as depicted in fig 2.6.

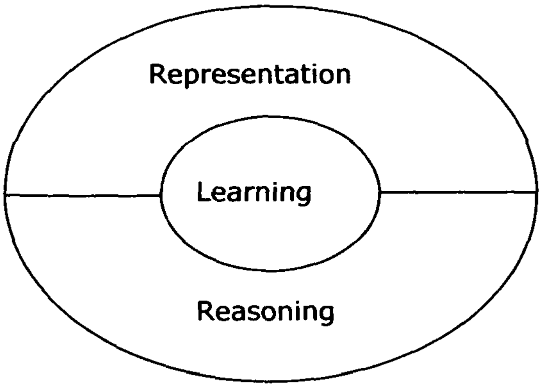


Figure 2.6 Illustration of the three key components of an AI system

(i) Representation: The most distinctive feature of AI is the pervasive use of a language of *symbol* structure to represent both general knowledge about a problem domain and specific knowledge about solution to the problem. The symbolic representation of AI is easy to understand by a human user and well suited for human-machine communication. The knowledge represents data. It may be of a declarative or procedural kind. In a declarative representation, knowledge is a static collection of facts, with a small set of procedures used to manipulate the facts. In a procedural representation, on the other hand, knowledge is embodied in an executable code.

(ii) Reasoning: In its most basic form, reasoning is the ability to solve the problems. For a system to qualify as a reasoning system, it must solve problems, make explicit and implicit information and must have control mechanism [26]. Problem solving may be viewed as a *searching* problem. A common way to deal with 'search' is to use *rules, data* and *control* [27]. The rules operate on the data, and the control operates on the rules. In many situations encountered in practice the available knowledge is incomplete or inexact. In such situations, *probabilistic reasoning* procedures are used, thereby permitting AI systems to deal with uncertainty [25, 28] more efficiently.

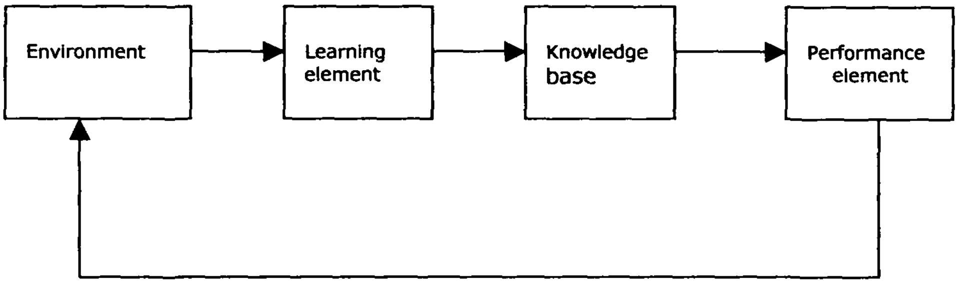


Figure 2.7 Simple model of machine learning

(iii) Learning: In the simple model of machine learning depicted in figure 2.7, the environment supplies some information to a *learning element.* The learning element then uses this information to make improvements in a *knowledge base,* and finally the *performance element* uses the knowledge base to perform its task. The kind of information supplied to the machines by the environment is usually imperfect, with the result that the learning element does not know in advance how to fill in missing details or to ignore the details that are unimportant. The machine therefore operates by guessing and then receiving *feedback* from the performance element. The feedback mechanism enables the machine to evaluate its hypotheses and revise them if necessary. Machine learning involves two different kinds of information processing: inductive and deductive. In *inductive* information processing, general patterns and rules are determined from raw data and experience. In *deductive* information processing general rules are used to determine specific facts. Similarity based learning uses induction, whereas the proof of a theorem is a deduction from known axioms and other existing theorems.

2.3.1 Comparison of AI and ANN

AI and ANN can be compared in the level of explanation, style of processing and representational structure [29].

(i) Level of explanation: In classical AI, the emphasis is on building *symbolic representations*. AI models cognition as the *sequential processing* of symbolic representations [30]. The emphasis in neural networks is on the development of *parallel distributed processing models* (PDP). The models assume that information processing takes place through the interaction of a large number of neurons, each of which sends excitatory and inhibitory signals to other neurons in the network.

(ii) Processing style: In classical AI, the processing is *sequential*, as in typical computer programming. The inspiration for sequential programming comes from natural language and logical inference, as much as from the structure of the Von Neumann Machine. In contrast, *parallelism* is essential to the processing of information to the neural networks. Parallelism may be massive (hundreds of thousands of neurons), which gives neural networks a remarkable form of robustness and flexibility. Noisy or incomplete inputs may still be recognized, a damaged network may still be able to function satisfactorily and learning needn't to be perfect.

(iii) Representational structure: The expressions of classical AI are generally complex, built in a systematic fashion from simple symbols or by virtue of the *compositionality* of symbolic expressions. The nature and structure of representations is, however, a crucial problem for neural networks. Neural networks are parallel distributed processors with a natural ability to learn. More potentially useful approach needs to build structured connectionist models or hybrid systems that integrate AI and ANN together.

2.4 Artificial Neural Network and Fuzzy Systems

Artificial neural networks (ANN) and fuzzy logic work together [2] in that the ANN classify and learn rules for fuzzy logic and fuzzy logic infers from unclear neural network parameters. The latter is a network with fast learning capabilities that produces intelligent and crisp output from fuzzy input and from fuzzy parameters and avoids time-consuming arithmetic manipulation. Incorporating fuzzy principles in a neural network gives more user flexibility and more robustness in the system [23].

2.4.1 Introduction to Neural Networks

Neural networks are composed of simple elements called neurons operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. Normally, the neural networks are adjusted or trained so that a particular input leads to a specific target output. Such a structure is shown in figure 2.8, where the network is adjusted based on the comparison of the output and the target, until the network output matches the target [23]. Typically many such input-target pairs are used in a supervised learning to train the networks.

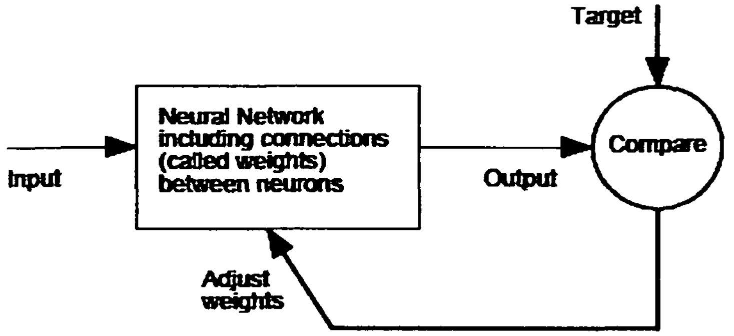

Target
Neural Network including connections (called weights) between neurons
Input
Output
Compare
Adjust weights

Figure 2.8 Basic Neural Network Model

A brief introduction to the neural networks is already presented in section 1.5. Here, some more features are introduced in the following sub sections.

2.4.1.1 Transfer Functions

Three mainly used transfer function are shown in figure 2.9. A hard-limit transfer function limits the output of the neuron to either *0*, if the net input argument *n* is less than *0*, or *1*, if *n* is greater than or equal to *0*. This function is used in Perceptron to create neurons that make classification decisions. Linear transfer function is used as linear approximator in the 'Linear Filters' [23]. The sigmoid transfer function takes an input, which may have any value between plus and minus infinity and squashes the output into the range from *0* to *1*.

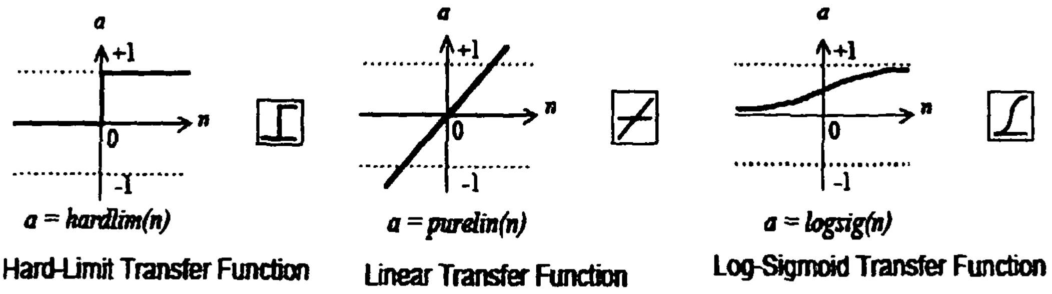


Figure 2.9 Transfer functions used for neural networks

2.4.1.2 Neuron with vector input

As shown in the figure 2.10, the inputs $p_1, p_2, \ldots p_R$ are multiplied by the weights $w_{1,1}, w_{1,2}, \ldots w_{1R}$ and the weighted values are fed to the summing junction. Their sum is simply *W.p*, the dot product of the matrix *W* and the vector *p* (single row). The neuron has a bias *b*, which is summed with the weighted inputs to form the net input *n*. The sum, *n*, is the argument of the transfer function *f*.

$$n = w_{1,1}p_1 + w_{1,2}p_2, + \ldots + w_{1R}\, p_R + b \qquad (2.1)$$

This expression can, of course, be written in MATLAB code as:

$$n = W^*p + b \tag{2.2}$$

and $$a = f(n) \tag{2.3}$$

2.4.1.3 Network Architectures

Two or more of the neurons can be combined in a layer and a particular network could contain one or more such layers. These are known as single layered or multilayered architecture of the neurons depending upon number of layers. Note that the outputs of each intermediate layer are the inputs to the following layer. A layer that produces the network output is called an *output layer*. All other layers are called *hidden layers*. It is common for different layers to have different number of neurons. Multiple-layer networks are quite powerful. For instance, a network of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function arbitrarily well. These networks are discussed in the next sections in details.

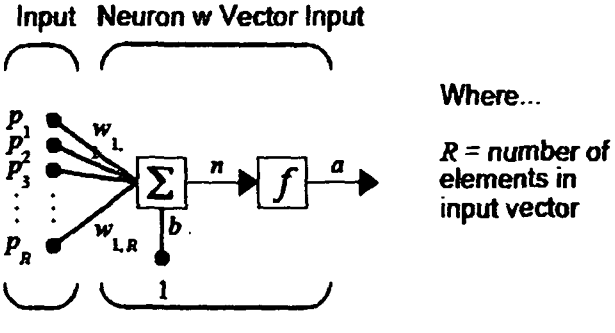


Figure 2.10 Neuron with vector input

2.4.1.4 Training Styles

There are two different styles of training. In *incremental* training the weights and biases of the network are updated each time an input is presented to the network. In *batch* training the weights and biases are only updated after all of the inputs are presented. Incremental training can be applied to both static and dynamic networks, although it is more commonly used with dynamic networks, such as adaptive filters. Batch training can be done using either *adapt* or *train* functions, although *train* is generally the best option, since it typically has access to more efficient training algorithms. Incremental training can only be done with *'adapt'* and *'train'* can only perform batch training.

2.4.2 Cluster Analysis

Cluster analysis, also called segmentation analysis or taxonomy analysis, is a way to partition a set of objects into groups or clusters in such a way that the profiles of objects in the same cluster are very similar and the profiles of objects in different clusters are quite distinct [23]. For example, a data set might contain a number of observations of subjects in a study where each observation contains a set of variables. Cluster analysis can be used to find two similar groups for the experiment and control groups in a study. In this way, if statistical differences are found in the groups, they can be attributed to the experiment and not to any initial difference between the groups. The clustering features can be explained as follows:

- **Terminology and Basic Procedure:**

(i) Similarity or dissimilarity between every pair of objects in the data set is evaluated by calculating the *distance* between objects.

(ii) The objects are grouped into a binary, hierarchical cluster tree. The distance information is used to determine the proximity of objects to each other. As objects are paired into binary clusters (clusters made up of two objects), the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed.

(iii) The objects in the hierarchical tree are divided into clusters using the cluster function. The cluster function can create clusters by detecting natural groupings in the hierarchical tree or by cutting off the hierarchical tree at an arbitrary point.

- **Finding the Similarities between Objects**

The distance between every pair of objects in a data set is calculated. For a data set made up of *m* objects, there are pairs in the data set. The result of this computation is commonly known as a similarity matrix (or dissimilarity matrix). There are many ways to calculate this distance information for example, the Euclidean distance between objects.

- **Defining the Links Between Objects**

Once the proximity between objects in the data set has been computed, objects in the data set should be grouped together into clusters, using the linkage function. The linkage function takes the distance information generated by link pairs of objects that are close together into binary clusters. The linkage function then links these newly formed clusters to other objects to create bigger clusters until all the objects in the original data set are linked together in a hierarchical tree.

- **Evaluating Cluster Formation:**

After linking the objects in a data set into a hierarchical cluster tree, it is to be verified that the tree represents significant similarity groupings. One way to measure the validity of the cluster information generated by the linkage function is to compare it with the original proximity data. If the clustering is valid, the linking of objects in the cluster tree should have a strong correlation with the distances between objects in the distance vector. Correlation coefficients are used to compare the results of clustering the same data set using different distance calculation methods or clustering algorithms.

- **Creating Clusters:**

After creating the hierarchical tree of binary clusters, hierarchy is divided into larger clusters. This can be done by finding the natural divisions in the data set and specifying arbitrary clusters. In the hierarchical cluster tree, the data set may naturally align itself into clusters as groups of objects are densely packed in certain areas and not in others. The inconsistency coefficient of the links in the cluster tree can identify these points where the similarities between objects change. In arbitrary clustering, instead of letting the cluster function create clusters determined by the natural divisions in the data set, the desired number of clusters can be specified. In this case, the value of the cutoff argument specifies the point in the cluster hierarchy at which to create the clusters [23].

2.4.3 Principal Component Analysis

In some situations, the dimension of the input vector is large, but the components of the vectors are highly correlated (redundant). It is useful in this situation to reduce the dimension of the input vectors. An effective procedure for performing this operation is Principal Component Analysis (PCA). This technique has three effects:

(i) it orthogonalises the components of the input vectors (so that they are uncorrelated with each other), (ii) it orders the resulting orthogonal components (principal components) so that those with the largest variation come first and (iii) it eliminates those components that contribute the least to the variation in the data set.

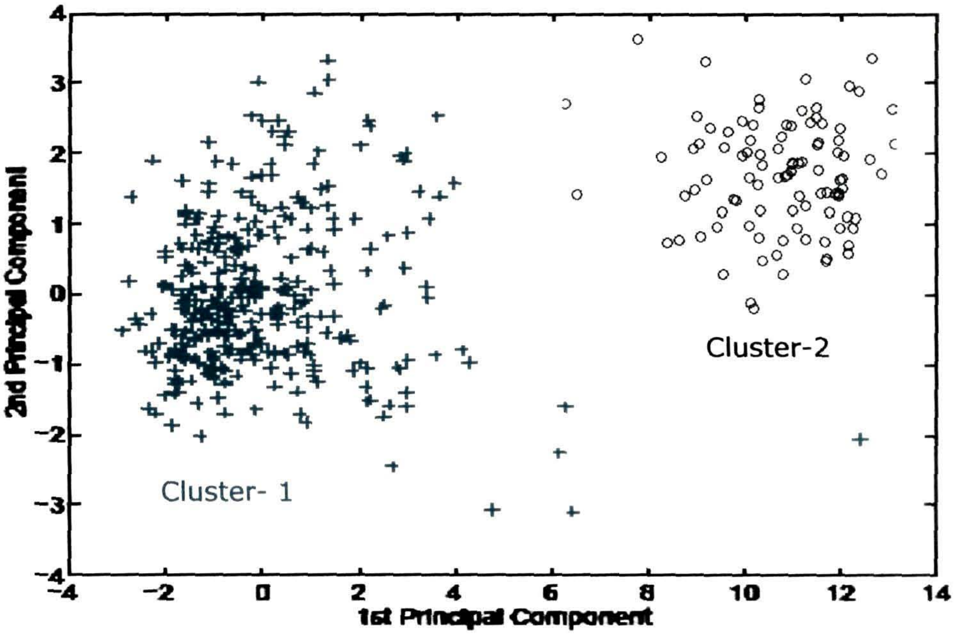


Figure 2.11 PCA plot using first two components only

Principal components analysis is a quantitatively rigorous method for achieving simplification. Each principal component is a linear combination of the original variables. The first principal component is a single axis in space. When each observation is projected on that axis, the resulting values form a new variable. And the variance of this variable is the maximum among all possible choices of the first axis. The second principal component is another axis in space, perpendicular to the first (figure 2.11). Projecting the observations on this axis generates another new variable. The variance of this variable is the maximum among all possible choices of this second axis. The full set of principal components is as large as the original set of variables. But it is common for the sum of the variances of the first few principal components to exceed 80% of the total variance of the original data [23].

By examining plots of these few new variables, researchers often develop a deeper understanding of the driving forces that generated the original data. Sometimes it makes sense to compute principal components for raw data. This is appropriate when all the variables are in the same units. Standardizing the data is reasonable when the variables are in different units or when the variance of the different columns is substantial.

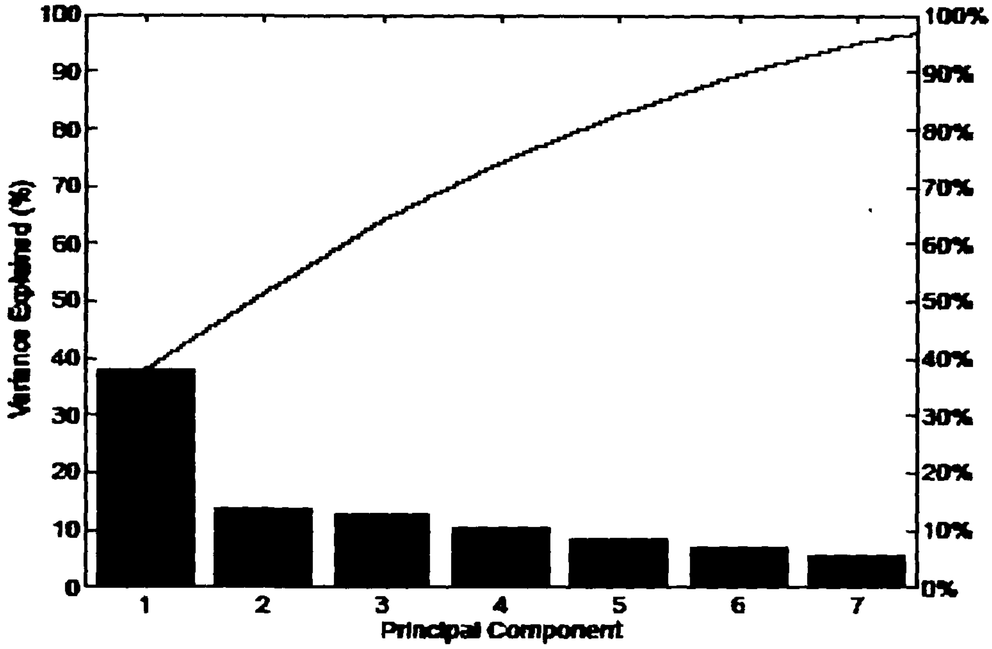


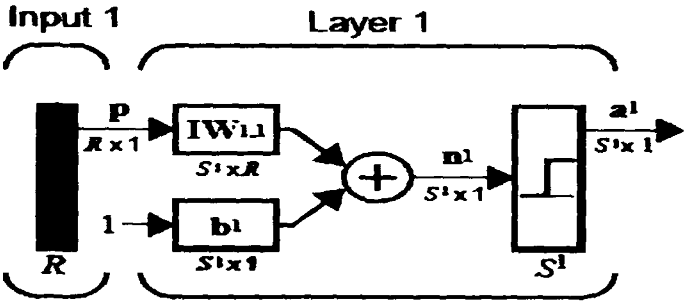
Figure 2.12 First three PC contributes nearly two third of variance

It can be seen in figure 2.12 that the first three principal components explain roughly two thirds of the total variability in the standardized ratings. For most of the applications these first three components are enough to produce fairly good results.

We first normalize the input vectors so that they have zero mean and unity variance. This is a standard procedure when using principal components. Those principal components that contribute less than 2% to the total variation in the data set are normally eliminated. A matrix is formed that contains the transformed input vectors. After the network has been trained, this matrix should be used to transform any future inputs that are applied to the network. It effectively becomes a part of the network, just like the network weights and biases.

2.4.4 Multi Layered Perceptron

Perceptrons are especially suited for simple problems in pattern classification. They are fast and reliable networks for the problems they can solve. In addition, an understanding of the operations of the perceptron provides a good basis for understanding more complex networks. It will be easy to discuss first single layer perceptron as depicted in figure 2.13 and then cascade these in series to make multi layered perceptron (MLP) [23].



$$a^1 = \mathrm{hardlim}(IW^{1,1}p^1 + b^1)$$

Where...

R = number of elements in input

S^1 = number of neurons in layer 1

Figure 2.13 Single layered perceptron architecture

The perceptron network consists of a single layer of S perceptron neurons connected to R inputs through a set of weights $W_{i,j}$ as shown in figure 2.13. The network indices i and j indicate that $W_{i,j}$ is the strength of the connection from the jth input to the ith neuron.

The perceptron learning rule is capable of training many cascaded single layer perceptrons. The cascaded form is called multi layered perceptron. Perceptron networks can be created with the function *newp* in MATLAB environment. These networks can be initialized, simulated and trained with the *init*, *sim* and *train* functions respectively.

A perceptron can be created with the function *newp* as

$$\text{net} = newp(\text{PR, S}) \qquad (2.4)$$

Here, input arguments PR is an R-by-2 matrix of minimum and maximum values for input elements. S is the number of neurons. Commonly the ***hardlim*** function is used in perceptrons [23].

2.4.5 Learning Vector Quantization Networks

A Learning Vector Quantization (LVQ) network has a first competitive layer and a second linear layer (figure 2.14). The competitive layer learns to classify input vectors. The linear layer transforms the competitive layer's classes into target classifications defined by the user. The classes learned by the competitive layer are called ***subclasses*** and the classes of the linear layer called ***target classes***. Both the competitive and linear layers have one neuron per (sub or target) class. Thus, the competitive layer can learn up to S_1 subclasses. These, in turn, are combined by the linear layer to form S_2 target classes. (S_1 is always larger than S_2.) For example, suppose neurons *1, 2,* and *3* in the competitive layer all learn subclasses of the input space that belongs to the linear layer target class No. *2*. Then competitive neurons *1, 2,* and *3*, will have $\boldsymbol{LW_{2,1}}$ weights of 1.0 to neuron $\boldsymbol{n_2}$ in the linear layer, and weights of *0* to all other linear neurons. Thus, the linear neuron produces a_1 if any of the three competitive neurons (*1,2,* and *3*) win the competition and output a_1. This is how the subclasses of the competitive layer are combined into target classes in the linear layer [23].

In short, a *1* in the *ith* row of a_1 (the rest of the elements of a_1 will be zero) effectively picks the *ith* column of $LW_{2,1}$ as the network output. Each such column contains a single *1*, corresponding to a specific class. Thus, subclass s_1 From layer *1* get put into various classes, by the $LW_{2,1}\ a_1$ multiplication in layer *2*. It is known ahead of time what fraction of the layer *1* neurons should be classified into the various class outputs of layer *2*, so that it can specify the elements of $LW_{2,1}$ at the start. However, it has to go through a training procedure to get the first layer to produce the correct subclass output for each vector of the training set.

An LVQ network can be created with the function *newlvq* in MATLAB environment as given below:

$$\text{net} = \textit{newlvq}\ (\text{PR,S1,PC,LR,LF}) \tag{2.5}$$

Here PR is an R-by-2 matrix of minimum and maximum values for R input elements, S_1 is the number of first layer hidden neurons, PC is an S_2 element vector of typical class percentages, LR is the learning rate (default 0.01) and LF is the learning function (default is *learnlv1*).

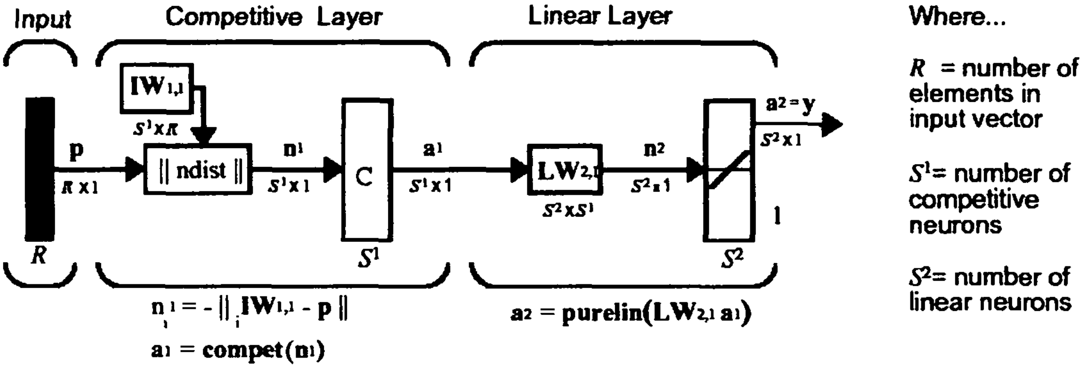


Figure 2.14 The LVQ network architecture

LVQ learning in the competitive layer is based on a set of input and target pairs. Each target vector has a single *1*. The rest of its elements are 0. The output as *1* tells the proper classification of the associated input. To train the network, an input vector **p** is presented, and the distance from **p** to each row of the input weight matrix $\boldsymbol{IW}_{1,1}$ is computed with the function *ndist*. The hidden neurons of layer *1* compete. Suppose that the *i*th element of $\mathbf{n}_1$ is most positive, and neuron *i** wins the competition. Then the competitive transfer function produces a *1* as the *i**th element of $\boldsymbol{a}_1$. All other elements of $\boldsymbol{a}_1$ are *0*. When $\boldsymbol{a}_1$ is multiplied by the layer *2* weights $\boldsymbol{LW}_{2,1}$, the single *1* in $\boldsymbol{a}_1$ selects the class, *k** associated with the input. Thus, the network has assigned the input vector **p** to class *k** and will be *1*. Of course, this assignment may be a good one or a bad one, for may be *1* or *0*, depending on whether the input belonged to class *k** or not.

The learning function that implements these changes in the layer *1* weights in LVQ networks is *learnlv1*. Next the network is trained with *train* to obtain first-layer weights that lead to the correct classification of input vectors.

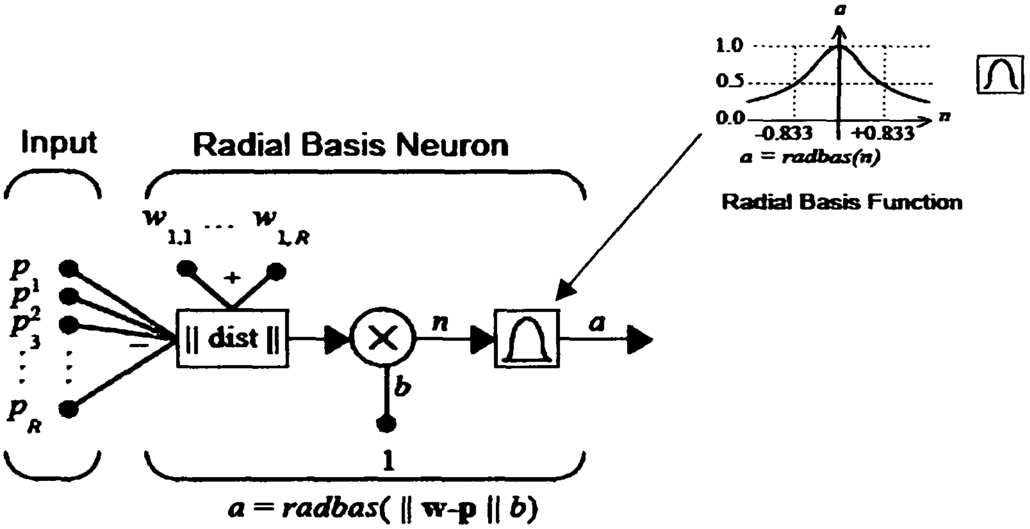


Figure 2.15 A radial basis network with *R* inputs.

2.4.6 Radial Basis Networks

Radial basis networks (figure 2.15) may require more neurons than standard feed-forward backpropagation networks, but often they can be designed in a fraction of the time it takes to train standard feed-forward networks. They work best when many training vectors are available. Radial basis networks can be designed with either *newrbe* or *newrb*. A generalized regression neural network (GRNN) and probabilistic neural networks (PNN) can be designed with *newgrnn* and *newpnn*, respectively [23].

Here the net input to the *radbas* transfer function is the vector distance between its weight vector *w* and the input vector *p*, multiplied by the bias b. The input, as shown in figure 2.15, accepts the input vector *p* and the single row input weight matrix *w*, and produces the dot product of the two.

The transfer function for a radial basis neuron is given by equation 2.6.

$$radbas(n) = e^{-n^2} \tag{2.6}$$

The output of the radial basis function has a maximum of *1* when its input is *0* (*0* distance between *w* and *p*). As the distance between *w* and *p* decreases, the output increases. Thus, a radial basis neuron acts as a detector that produces 1 whenever the input *p* is identical to its weight vector *w* (*0* distance between *w* and *p*). The bias *b* allows the sensitivity of the *radbas* neuron to be adjusted.

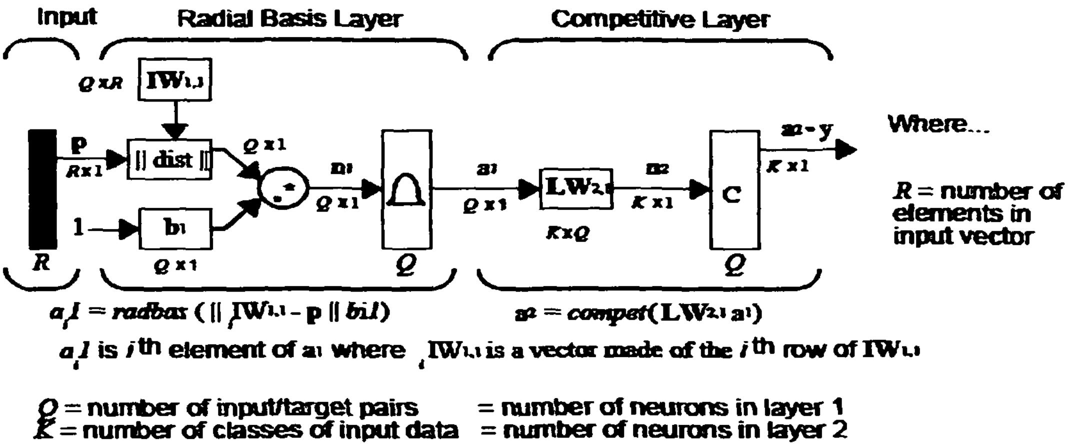

Input
Radial Basis Layer
Competitive Layer
Q x R
IW1,1
p
R x 1
|| dist ||
Q x 1
.*
n1
Q x 1
1
b1
Q x 1
R
Q
a1
Q x 1
LW2,1
K x Q
n2
K x 1
C
Q
a2 = y
K x 1
Where...
R = number of elements in input vector
$a_i1 = radbas(\| {}_i\mathrm{IW}_{1,1} - p \| b_i1)$
$a2 = compet(\mathrm{LW}_{2,1}\, a1)$
a_i1 is ith element of a1 where ${}_i\mathrm{IW}_{1,1}$ is a vector made of the ith row of IW1,1
Q = number of input/target pairs = number of neurons in layer 1
K = number of classes of input data = number of neurons in layer 2

Figure 2.16 The architecture of Probabilistic Neural Networks

2.4.7 Probabilistic Neural Networks

Probabilistic neural networks (PNN) can be used for classification problems [23]. 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 its net output 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 architecture for this network is shown in figure 2.16.

The first-layer input weights, $IW_{1,1}$ (net.IW{1,1}) are set to the transpose of the matrix formed from the Q training pairs and **P'**. When an input is presented the ||dist|| box produces a vector whose elements indicate how close the input is to the vectors of the training set. These elements are multiplied, element by element, by the bias and sent to the *radbas* (bell shaped function) transfer function. An input vector close to a training vector is represented by a number close to 1 in the output vector $\mathbf{a}_l$. If an input is close to several training vectors of a single class, it is represented by several elements of $\mathbf{a}_l$ that are close to 1.

The second-layer weights, $LW_{1,2}$ (net.LW{2,1}), are set to the matrix **T** of target vectors. Each vector has a_l only in the row associated with that particular class of input, and *0*'s elsewhere. The multiplication $\boldsymbol{T.a_l}$ sums the elements of $\boldsymbol{a_l}$ due to each of the *K* input classes. Finally, the second-layer transfer function, compete, produces a *1* corresponding to the largest element of $\boldsymbol{n_2}$, and *0*'s elsewhere. Thus, the network has classified the input vector into a specific one of *K* classes because that class had the maximum probability of being correct [23].

2.4.8 Fuzzy logic

In Boolean logic the function of Boolean operators (gates) AND, OR and INVERT is well known. For instance, by "gating" the value of two variables using an AND, we get 11→ 1, 10→ 0, 01→ 0, or 00→ 0. In fuzzy logic, the values are not crisp, and their fuzziness exhibits a distribution described by the membership function. Hence, if two fuzzy variables are "gated", the output will not be like that of Boolean gates. This theory has been addressed by various fuzzy logics. Here we consider mix-max logic. In simple terms, if we consider "union" (equivalent to OR), the output is equal to the input variable with the greatest value, max $(x_1, x_2, \ldots\ldots, x_n)$. That is, if A = 0.5, B = 0.7 and C = A OR B, then C = max (0.5, 0.7) = 0.7. If we consider "intersection" (equivalent to AND), the outcome is equal to the least value of the input variables, $\min(x_1, x_2, \ldots\ldots\ldots, x_n)$. In this case, if C = A AND B, then C = min (0.5, 0.7) = 0.5. If we consider "complement" (equivalent to NOT), then the outcome is the complement of 1 or $x^- = 1 - x$. If $C = B^-$, then, C = 1 – 0.7 = 0.3.

2.4.8.1 Fuzzy rule generation

In most fuzzy problems the rules are generated based on past experience. Concerning problems that deal with **fuzzy engines** or **fuzzy control**, one should know all possible input-output relationships even in fuzzy terms [2]. The input-output relationships, or **rules**, are then easily expressed with **if......then** statements, such as

If A_1 and/or B_1, then H_{11}, else.
If A_2 and/or B_1, then H_{21}, else.
If A_1 and/or B_2, then H_{12}, else.
If A_2 and/or B_2, then H_{22}.

Here 'and/or' signifies logical union or intersection, the A's and B's are fuzzified inputs and the H's are action for each rule.

The case where rules are expressed by a single input variable 'if A_1 then H_1, if A_2 then H_2,........., if A_n then H_n' represents a simple translation (or transformation) of input variables to the output. However, the most common fuzzy logic problems involve more than one variable. The *if...then* rule becomes more difficult to tabulate if the fuzzy statements are more involved (i.e., have many variables), such as *if* A and B and C or D, *then* H. Tabulation is greatly simplified if statement decomposition process [2] is followed. For example, consider the original problem statements of the form

If A_i and B_j and C_k, *then* H_{ijk}.

This statement is decomposed as

If A_i and B_j, *then* H_{ij}
If H_{ij} and C_k, *then* H_{ijk}

H_{ij} is an intermediate variable.

2.4.8.2 Defuzzification of fuzzy logic

The defuzzification process is an important step. Based on this step, the output action may or may not be successful. It is not uncommon to have, based on the rules and membership functions. In general defuzzification is the process where the membership functions are sampled to find the grade of membership used in the fuzzy logic equations and also an outcome region is defined [2]. From this, the output is deduced.

Several techniques have been developed to produce an output. The three most commonly used are:

Maximizer: by which the *maximum* output is selected.

Weighted average: which averages weighted possible outputs.

Centroid: which finds centre of mass of output.

2.4.9 Neuro Fuzzy PARC

These techniques provide a method for the tuzzy modelmg procedure to *learn* information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. This learning method works similar to that of neural networks. The Fuzzy Logic Toolbox function that accomplishes this membership function parameter adjustment is called *anfis*. The acronym *ANFIS* derives its name from *adaptive neuro-fuzzy inference system*. Using a given input/output data set, the function *anfis* constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a backpropagation algorithm alone, or in combination with a *least squares* type of method. This allows fuzzy systems to learn from the data for modeling.

2.4.9.1 FIS Structure and Parameter Adjustment

A network-type structure similar to that of a neural network can be used to interpret the input/output map. This structure maps inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs. The parameters associated with the membership functions will change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector, which provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters. Once the gradient vector is obtained, any of several optimization routines could be applied in order to adjust the parameters so as to reduce some error measure (usually defined by the sum of the squared difference between actual and desired outputs). The *anfis* uses either back propagation or a combination of least squares estimation and backpropagation for membership function parameter estimation.

2.4.9.2 Constraints of *anfis*

An *anfis* is much more complex than the fuzzy inference systems and is not available for all of the fuzzy inference system options. Specifically, *anfis* only supports Sugeno-type systems, and these must be first or zeroth order Sugeno-type systems, single output, obtained using weighted average defuzzification (linear or constant output membership functions) and of unity weight for each rule.

2.4.10 Linear Regression Methods

The purpose of multiple linear regressions is to establish a quantitative relationship between a group of predictor variables (such as sensors in e-nose), x and a response, y (such as tea flavour terms). This relationship is useful for (i) understanding which predictors has the greatest effect, (ii) knowing the direction of the effect (i.e., increasing x increases/decreases y) and (iii) using the model to predict future values of the response when only the predictors are currently known.

2.4.10.1 Mathematical Foundations of Multiple Linear Regressions

The linear model takes its common form

$$y = X\beta + \varepsilon \tag{2.7}$$

Here y is an n-by-1 vector of observations, X is an n-by-p matrix of regressors, β is a p-by-1 vector of parameters, ε is an n-by-1 vector of random disturbances.

The solution to the problem is a vector, b, which estimates the unknown vector parameters, β. The least squares solution is

$$b = \hat{\beta} = (X^T X)^{-1} X^T y \tag{2.8}$$

The residuals are the difference between the observed and predicted y values.

$$r = y - \hat{y} = (I - H)y \tag{2.9}$$

The residuals are useful for detecting failures in the model assumptions, since they correspond to the errors, ε, in the model equation. By assumption, these errors have independent normal distributions with mean zero and a constant variance.

2.4.10.2 Stepwise Regression

Stepwise regression is a technique for choosing the variables to include in a multiple regression model. Forward stepwise regression starts with no model terms. At each step it adds the most statistically significant term until there are none left. Backward stepwise regression starts with all the terms in the model and removes the least significant terms until all the remaining terms are statistically significant. It is also possible to start with a subset of all the terms and then add significant terms or remove insignificant terms. An important assumption behind the method is that some input variables in a multiple regression do not have an important explanatory effect on the response. If this assumption is true, then it is a convenient simplification to keep only the statistically significant terms in the model.

2.4.10.3 Generalized Linear Models

There are some nonlinear models, known as generalized linear models that can fit into simpler linear methods. In generalized linear models, these characteristics are generalized as follows:

- The response has a distribution that may be normal, binomial, Poisson, gamma, or inverse Gaussian, with parameters including a mean μ.
- A coefficient vector b defines a linear combination $X^{*}b$ of the predictors X.
- A link function $f(\cdot)$ defines the link between the two as $f(\mu) = X^{*}b$.

2.4.11 Comparative Note on ANNs

Perceptrons are useful as classifiers. They can classify linearly separable input vectors very well. Convergence is guaranteed in a finite number of steps providing the perceptron can solve the problem. Perceptrons have a single layer of hard-limit neurons. The number of network inputs and the number of neurons in the layer are constrained by the number of inputs and outputs required by the problem. Training time is sensitive to outliers, but outlier input vectors do not stop the network from finding the solution.

Backpropagation networks can classify nonlinearly separable input vectors. A graphical user interface can be used to create networks and data, train the networks, and export the networks and data.

Radial basis networks can be designed very quickly in two different ways. The first design method, *newrbe*, finds an exact solution. The function *newrbe* creates radial basis networks with as many radial basis neurons as there are input vectors in the training data. The second method, *newrb*, finds the smallest network that can solve the problem within a given error goal. Typically, far fewer neurons are required by *newrb* than are returned *newrbe*. However, because the number of radial basis neurons is proportional to the size of the input space and thus RBFs are larger than backpropagation networks.

A generalized regression neural network (GRNN) is often used for function approximation. It has been shown that, given a sufficient number of hidden neurons, GRNNs can approximate a continuous function to an arbitrary accuracy.

Probabilistic neural networks (PNN) can be used for classification problems. Their design is straightforward and does not depend on training. A PNN is guaranteed to converge to a Bayesian classifier providing it is given enough training data. These networks generalize well. The GRNN and PNN have many advantages, but they both suffer from one major disadvantage that they are slower to operate because they use more computation than other kinds of networks to do their function approximation or classification.

LVQ networks classify input vectors into target classes by using a competitive layer to find subclasses of input vectors, and then combining them into the target classes. Unlike perceptrons, LVQ networks can classify any set of input vectors, not just linearly separable sets of input vectors. The only requirement is that the competitive layer must have enough neurons and each class must be assigned enough competitive neurons. To ensure that each class is assigned an appropriate amount of competitive neurons, it is important that the target vectors used to initialize the LVQ network must have the same distributions of targets as the training data.

2.5 Conclusion

A comprehensive literature review of olfaction and AI is presented in this chapter. Firstly biological systems related to the olfaction and nervous systems are presented in detailed and illustrated form. The recent developments on the subject are discussed at length. This will help in understanding the E-nose technology, its functioning and implementation part. A comparative illustration on biological and artificial neuron is included to make the analogy completely apparent.

Artificial neural networks are covered meticulously selecting relevant and most critical components, which are most essential for understanding the fundamentals of the ANNs. This subject is highly mathematical and vastly diversified which ultimately led to the trade offs at times. It, however, has been endeavored to present in compact and yet complete form. Various ANN paradigms, which have been used in the research, are discussed and compared by merits.

An introductory part on fuzzy systems and artificial intelligence is included since these two fields are closely related with the ANN and PARC. ANNs have been born out AI and these represent modern intelligent instrumentation systems. Finally a brief comparative note on different ANN paradigms is included to compare at a glance.

References

[1]. Ohloff, G., Scent and Fragrances, Springer-Verlag, Berlin Heidelberg, 1994, p. 6.

[2]. Stamations V. Kartalopoulos Understanding Neural Networks and Fuzzy Logic; Basic Concepts and Applications IEEE Press, Prentice Hall of India Pvt Ltd 2000

[3]. Denise Chen & Haviland-Jones, *Physiology and Behaviour* 1999; 68: 241-250)

[4]. Ackerl, Atzmueller & Grammer, *Neuroendocrinol Lett* 2002; 23(2): 79-84)

[5]. Goldberg S., J. Turpin and S. Price, *Anisole binding protein from olfactory epithelium evidence for a role in transduction*, Chem. Senses & Flavour, 4:207 (1979).

[6]. Fesenko E.E., V.I. Nonoselov and L.D. Krapivinskaya, *Molecular mechanisms of olfactory reception*, Biochim. Biophys. Acta, 587:424 (1979).

[7]. Lacazette E, Gachon AM, Pitiot G., *A novel human odourant-binding protein gene family resulting from genomic duplicons at 9q34: differential expression in the oral and genital spheres*, Hum Mol Genet, Jan 22; 9(2): 289-301 (2000)

[8]. Price, S., Willey, A., *Effects of antibodies against odourant binding proteins on electrophysical responses to odourants*, Biochem Biophys Acta, 965: 127ff (1988)

[9]. Langedijk AC, Spinelli S, Anguille C, Hermans P, Nederlof J, Butenandt J, Honegger A, Cambillau C, Pluckthun A., *Insight into odourant perception: the crystal structure and binding characteristics of antibody fragments directed against the musk odourant traseolide*, J Mol Biol 1999 Oct 1; 292(4): 855-69

[10]. LAWRENCE K. ALTMAN Unraveling Enigma of Smell Wins Nobel for 2 Americans Published: October 5, 2004

[11]. Nobel Prize Press Release, *The 2004 Nobel Prize in Physiology or Medicine*, October 4, 2004.

[12]. Linda Buck: *Information Coding in the Olfactory System*, http://www.hhmi.org/science/neurosci/buck.htm

[13]. Richar Axel: *The Molecular Logic of Olfaction*, http://www.hhmi.org/science/neurosci/axel.htm

[14]. Buck, L. and R. Axel, *A novel multigene family may encode odour recognition: a molecular basis for odour recognition*, Cell, 65:175 (1991).

[15]. Sergey Zozulya, Fernando Echeverri & Trieu Nguyen, *The human olfactory receptor repertoire*, Genome Biology 2001 2(6): research0018.1-0018.12

[16]. Lancet, Doron, et.al., *Molecular recognition and evolution in biological repertoires: from olfaction to the origin of life (2000)* at http://www.weizmann.ac.il/Biology/open-day/images/lancet.pdf

[17]. Rice, Ken, *The Evolution and Classification of G-protein-coupled Receptors*, http://www.cis.upenn.edu/~krice/receptor.html (1998)

[18]. Zhao, H., L.Ivic, J.M. Otaki, M. Hashimoto, K. Mikoshiba, S. Firestein, *Functional expression of a mammalian odourant receptor*. Science, Jan 9; 279(5348): 237-242 (1998)

[19]. Travis, John. Making Sense of Scents, Science News, April 10, 1999, Malnic B, Hirono J, Sato T, Buck LB; *Combinatorial receptor codes for odours*. Cell, Mar 5;96(5):713-23 (1999): http://www.hhmi.org/news/buck.htm

[20]. Glusman G., Bahar A., Sharon D., Pilpel Y., White J, Lancet D., *The olfactory receptor gene superfamily: data mining, classification, and nomenclature.*, Mamm. Genome; Nov;11(11):1016-23 (2000).

[21]. Glusman G, Yanai I, Rubin I, Lancet D., *The complete human olfactory subgenome*, Genome Res. 2001 May;11(5):685-702

[22]. Anholt, R.R., *Molecular neurobiology of olfaction*, Crit. Rev. Neurobiol. 7(1): 1-22 (1993)

[23]. Simon Haykin, *Neural Networks: A Comprehensive Foundation*, Pearson Education Asia, 1999.

[24]. Loewi, O., *From the Workshop of Discoveries*, Lawrence: University of Kansas Press, 1953.

[25]. Russell, S.J., and P. Novig, 1995. *Artificial Intelligence: A Modern Approach*, Upper Saddle River, NJ: Prentice-Hall

[26]. Fischler, M.A., and O. Firschein, 1987. *Intelligence: The Eye, The Brain, and The Computer,* Reading, MA: Addison-Wesley.

[27]. Nilsson, N.J., 1980. *Principles of Artificial Intelligence*, New York: Springer-Verlag.

[28]. Pearl, J., 1988. *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA: Morgan Kaufmann. (Revised 2nd printing, 1991).

[29]. Memmi, D., 1989. "Connectionism and artificial intelligence," *Neuro-Nimes'89 International Workshop on Neural Networks and their Applications,* pp. 17-34, Nimes, France.

[30]. Newell, A., and H.A. Simon, 1972. *Human Problem Solving,* Englewood Cliffs, NJ: Prentice-Hall.

§§

CHAPTER 3

DESIGN AND DEVELOPMENT OF ELECTRONIC-NOSE SETUP

Chapter 3

Design and Development of Electronic-Nose Setup

Gas sensors have been used for a long time, however, E-nose technology has recently gone several steps farther. Arrays of sensors that respond to a wide range of gaseous compounds are used, with advanced pattern recognition and artificial intelligence techniques, which enable users to readily extract relevant and reliable information. In the past decade, many papers have appeared in the literature describing the uses of E-nose. These devices are typically array of sensors used to characterize complex odour samples. These arrays of gas sensors are termed as E-noses. Already, commercial systems from several manufactures have targeted applications that range from quality control and assurance of food and drugs to medical diagnosis, environmental monitoring, process operations, safety, security and military use. A brief history of E-nose and principles of sensor design and technology are described in this chapter. This is necessary to fully understand the technology of E-nose subject and to appreciate their impact upon the development of prototype E-nose models. In this research prototype E-nose setup has been designed and used for quality determination of tea and spices. E-nose data handling in MATLAB and GENIE environment are described along with feature selection techniques and User Graphic Interfacing (GUI).

3.1 History of Electronic-Nose

The term, *E-nose* has come into existence as a general term for an array of chemical gas sensors incorporated into an artificial olfaction device, after its introduction in the title of a landmark conference on this subject in Iceland in 1991. The use of this term is reasonable, as there are remarkable analogies between the artificial noses and the "Bio-nose" evolved by nature. Work on artificial olfaction started with the development of electronic instrumentation and ANN techniques. Hartman reported the first work on an experimental instrument in 1954 concerning research on electrochemical sensors. Ten years later, Wilkens and Hartman introduced the concept of odour sensors [1]. They developed a system comprising of an array of eight different electromechanical cells using various combinations of metal electrodes, electrolytes and applied potentials. At about same time in 1961, Moncrieff reported on another approach to odour sensors using temperature-sensitive resistors coated with different materials. He postulated that when built into an array they would be able to discriminate between a large numbers of different gaseous smells. The idea of using metal and semiconductor gas sensors for odour sensing was published a few years later while working on modulation of contact potential.

However, the concept of an E-nose as a chemical sensor array for odour classification did not emerge until nearly 20 years later. Zaromb and Stetter proposed the use of an array of gas sensors with partially overlapping sensitivities to detect complex odours. Meanwhile, a group of scientists led by K. Persaud in the UK were approaching the sensor array problem as a means to understand the biological process of olfaction. They created a mimic of the array of biological gas sensors in the human nose with an array of three electrochemical metal oxide (MOX) sensors. They were able to discriminate 20 odorous pure compounds and complex essential oils from one another using this array. Following work in the mid 1980s in the USA included an array of MOX sensors at Carnegie Mellon University and an array of mechanical surface acoustic wave (SAW) sensors at the Naval Research Laboratories, both designed to detect hydrocarbons. In Japan, fish freshness testing was investigated with MOX sensor arrays. Advances in the technology have been made ever since early 1980s when researchers at the University of Warwick in Coventry, UK, developed sensor arrays for odour detection. Focused primarily on the sensor aspect of the problem, the initial research explored the use of metal oxide devices. Later work at Warwick University explored the use of conducting polymers. Those early efforts have attracted several commercial enterprises. The development of this sensor array technology continued in the 1980s. However, the E-nose received a renewed expansion of interest after the North Atlantic Treaty Organization (NATO) conference, 1991, on this subject, a series of annual meetings on E-nose technology and the new website and school for the E-nose sponsored by the European community. In 1995 the study of sensors based on porphyrins started. These studied resulted in Libra-Noses. Libra-Nose is based on an array of Thickness Shear Mode Resonators (TSMR), also known in literature as Quartz Microbalance (QMB) sensors. The chemical sensitivity is given by molecular film of pyrrolic macrocycles. Porphyrin-based sensors have been demonstrated to work with different basic transducers such as TSMR. Surface acoustic wave, conductivity, work function, and optical properties have been used for E-nose sensors.

As a part of this research work the experiments were conducted with a Warwick University MOS setup to predict the quality of overlapping flavours of tea [2, 3]. Another setup was developed at Tezpur University, which was successfully used for non-overlapping tea flavour quality determination and spice aroma discrimination with drift compensations [4].

3.2 Principles of Electronic-Nose Technology

The terms *Electronic-nose* is popular and descriptive but sometimes it is technically not correct. The chemical sensor array is more generally appropriate term for this field because of the fact that these systems are typically not sensing the same chemicals in the same way as human nose does. The human perceptions may be due to an entirely different chemical presence in the samples than the chemicals that are detected by the sensors in a sensor array. This is obvious when one considers that the E-nose often contains a sensor that is sensitive to carbon monoxide and the human nose cannot respond to this compound at all. The reverse also applies. However, chemical sensor arrays can be *nose like* in certain respects. For example, the rancidity of olive oil has been traced to the presence of one or two specific aldehydes that are formed during spoiling. These are readily detected by human olfactory systems as well as by chemical sensors. The E-nose and the human nose, both effectively discriminate bad olive oil from good. The E-nose can also create different patterns depending upon the concentration of aldehydes in the oil. This is an example of the analytical capability of the E-nose being similar to the human nose and is a simple example that can be explained fully on a molecular basis. The E-nose has the interesting ability to address analytical problems that have been noncompliant to traditional analytical approaches. The volatiles from coffee, for example, contain at least 640 different compounds. These interact in such a way that no single compound or group of compounds is associated with the subjective assessment of flavour or odour. The flavour of *good* coffee cannot be traced to a specific molecular origin or simple list of chemicals by traditional analytical techniques like gas chromatography or mass spectroscopy. One analyzes *good* coffee and *bad* coffee as described by coffee experts and finds differences in concentrations of hundreds of the compounds in the samples. Today, it is still not possible to add or subtract these known compounds according to a rational system, and make the *bad* coffee *good* or vice versa. We do not have a sufficient compositional understanding at the molecular level of the gustatory terms *good* and *bad* coffee. Yet the E-nose can learn the fingerprints of coffees and easily tell these different coffees apart. A statistical relationship between the array patterns for good and *bad* can be easily found. Something within the complex response of the sensors in the array is encoded with the difference between *good* and *bad* in the chemical fingerprint, just as it is in the complex response of the millions of receptors in our nose. We still do not know exactly the molecular basis for this difference but we can definitely say that array response vectors are different and statistically related to the quality *good* or *bad* coffee. Frequently no such chemical markers or specific molecular cause can be found for the quality of many perfumes or the off-odour of a group of recycled plastic parts. In such cases, an olfactory panel, that is, a panel of skilled and trained human noses, performs determination of flavour or odour. The response of a chemical gas sensor array or electronic nose can be statistically related to such properties as odour, flavour, explosive, or even bacterial content. Thus, the E-nose offers an opportunity to develop an instrumental approach for the human analytical terms like odour, flavour, hazardous, contaminated, spoiled, and the like.

The signals for the sensors, when taken together, form a pattern or fingerprint that is easily recognized by the nose as different for each chemical. A simple computer program, that treats each set of sensor responses as a vector in n-dimensional space (where *n* is the number of sensor channels) can compare the vectors and tell the difference. In the simple demonstration of the E-nose approach, the vectors are normalized to take out the concentration information, simple normalization requires only setting the largest signal to unity and scaling the other signals accordingly. The normalized vectors are stored as a collection of reference patterns in the memory of the computer.

To identify an unknown odourous gas, the vector for the gas is measured and normalized. Each element in its vector is subtracted from the corresponding element in the reference vector. The numerical length of the difference vector will be small if the patterns are very similar and large if they are different. Each vector is compared to the unknown, the closest one may be the identity of the unknown. If none of the vectors are close to each other, the unknown is not in the library of reference vectors and is unknown to the E-nose. The E-nose can only identify those compounds and mixtures that have been taught to it, just like our own nose. But it can learn new odours simply, which is often called calibration when it is done for ordinary instruments but training for E-nose. This is a simple description of the so-called nearest-neighborhood approach, which was implemented in first real E-nose and is still a powerful pattern classification method. Later, neural networks for the identification of grain odours were adopted. This was also one of the first examples in which the E-nose was used for sniffing out biologically caused contamination. The power of the E-nose, to extend the sensors' ability to perform both qualitative and quantitative analysis in a single step, is unique.

The analytical power of the E-nose is due to the large dimensionality of sensor arrays in what is called their *feature space*. This gives the array the power to encode such information as the *odour/flavour* of coffee, using the relative signals of an array of differently selective chemical sensors. The feature space is created by the diversity of the chemical responses of the sensors. The dimensionality of feature space is large for even small sensor arrays and large arrays can represent a very large number of different situations (10^{21} or more). Even for all its power, the E-nose approach is not exempted from the fundamental principle of analytical chemistry.

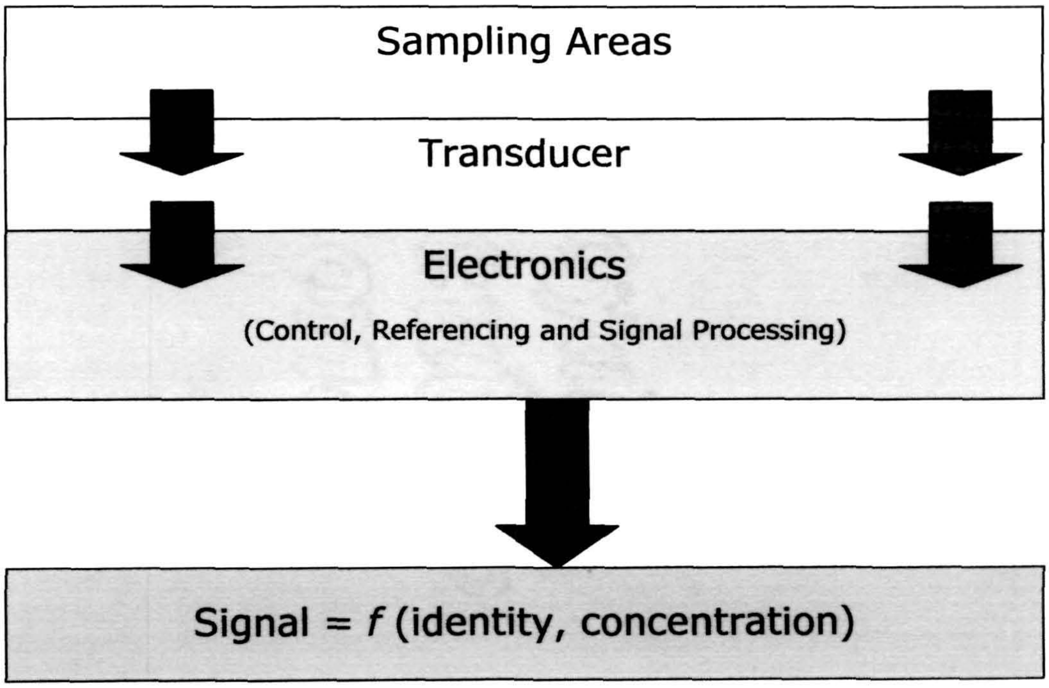

Sampling Areas
Transducer
Electronics
(Control, Referencing and Signal Processing)
Signal = *f* (identity, concentration)

Figure 3.1: The main parts of a typical sensor

3.3 Sensor Technology of Electronic-nose

A chemical sensor is a device, which responds to a particular odourant in a selective way by means of a reversible chemical interaction and can be used for the quantitative or qualitative determination of the odourants [5]. All sensors are composed of two main regions: the first is where the selective chemistry occurs and the second is the transducer. The transducer allows the conversion of one form of energy to another. The chemical reaction produces a signal such as a conductivity or colour change, fluorescence, production of heat or a change in the oscillator frequency of a crystal [5]. Other parts of a sensor include the signal processing electronics and a signal display unit. The major regions of a typical sensor are shown in Figure 3.1. Several categories of transducers are available and these include Electrochemical, such as ion-selective electrodes (ISE), ion-selective field effect transistors (FET), solid electrolyte gas sensors and semiconductor-based gas sensors. Piezoelectric transducers are surface acoustic wave (SAW) sensors. Piezoelectric materials are sensitive to changes in mass, density or viscosity and, therefore, frequency can be used as a sensitive transduction parameter [6]. Quartz is the most widely used piezoelectric material because it can act as a mass-to-frequency transducer. Optical transducers include optical fibers, as well as the more traditional absorbance, reflectance, luminescence and surface plasmon resonance (SPR) techniques. In thermal systems the heat of a chemical reaction involving the odourants is monitored with a transducer such as a thermistor. A subdivision of the sensor grouping is the biosensors. These incorporate a biological sensing element positioned close to the transducer. Polymer conductor based transducers are used extensively for E-nose applications.

3.3.1 Semiconductor Metal Oxide Sensors

A commercially available Taguchi Gas Sensor (TGS) can be and is widely used as the core-sensing element in array-based odour detectors. This consists of an electrically heated ceramic pellet upon which a thin film of tin oxide doped with precious metals is deposited [7]. Tin oxide is an *n*-type semiconductor and when oxygen adsorbs on the surface, one of the negatively charged oxygen species is generated depending on the temperature. This results in the surface potential becoming increasingly negative and the electron donors within the material become positively charged. When an oxidisable material comes into contact with the sensor surfaces the adsorbed oxygen is consumed in the resulting chemical reaction. This reduces the surface potential and increases the conductivity of the film. Several recent developments with tin oxide detectors have led to further advantages over the Taguchi sensor, which generally requires high power consumption and high temperatures. These include the fabrication of thin-film tin (II) oxide arrays using planar microelectronic technology leading to reduced size and lower power use, the production of thin-film sensors by chemical vapour deposition and the use of screen printing to make thick-film sensors [7]. Typical functioning of the MOS sensor is shown in figure 3.2. The transducer keeps the sensing material at an elevated temperature. Changes in the composition of the ambient atmosphere create a corresponding change in the resistance of the sensing layer, allowing the sensor to detect a wide range of toxic and explosive gases even at very low concentrations. The sensing layer is a porous thick film of polycrystalline tin oxide (SnO_2). In normal ambient air, oxygen and water vapor-related gases are absorbed at the surface of the SnO_2 grains. For reducing gases such as CO, a reaction takes place with the pre-absorbed oxygen and water vapor-related gases that decreases sensor resistance Conversely, oxidizing gases such as NO_2 and O_3 increase the resistance. The magnitude of change depends on the microstructure and composition or doping of the base material on the morphology and geometrical characteristics of the sensing layers and substrate, as well as on the temperature at which the sensing takes place. To detect a wide variety of different gases or classes of gases, the sensor can be tuned to detect alterations in any of these parameters. In practice, the relationship between sensor resistance and concentration of the target gas usually follows a power law that can be described by,

$$R = K\;X\;C^{+n} \tag{3.1}$$

Where C = concentration of target gas, K= a measurement constant and n has values between 0.3 and 0.8; the positive sign is used for oxidizing gases, while the negative sign is used for reducing gases

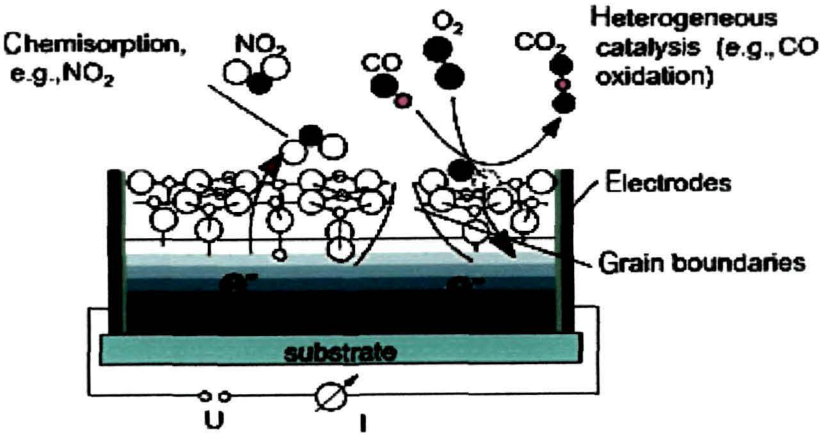

Chemisorption, e.g.,NO_2
NO_2
CO
O_2
CO_2
Heterogeneous catalysis (e.g.,CO oxidation)
Electrodes
Grain boundaries
substrate
U
I

Figure 3.2: The metal oxide semiconductor (MOS) sensor consists of a sensing material and a transducer (substrate).

Figure 3.3 shows a simple electrical circuit that can be used to measure sensor resistance, R_S. The heating voltage, V_H, is applied between 1 and 5 V, with typical values for both types of sensor ranging between 2 and 5 V. The measuring voltage, V_S, is applied between 2 and 4 V, with the recommended value not to exceed 5 V. To determine R_S, V_{out} is measured and R_L is known. The relationship between R_S and V_{out} can be given by equation 3.2.

$$R_s = R_L\left(\frac{V_S}{V_{out}} - 1\right) \tag{3.2}$$

For example, the sensor resistance drops very quickly immediately after CO exposure. After the CO is removed, resistance quickly reverts to its original value. Response and recovery speed are determined by the operating temperature, the sensor layer type, and the gases involved. Quick recovery is essential for detection of multiple gases, as sensing instruments are blind during the recovery period.

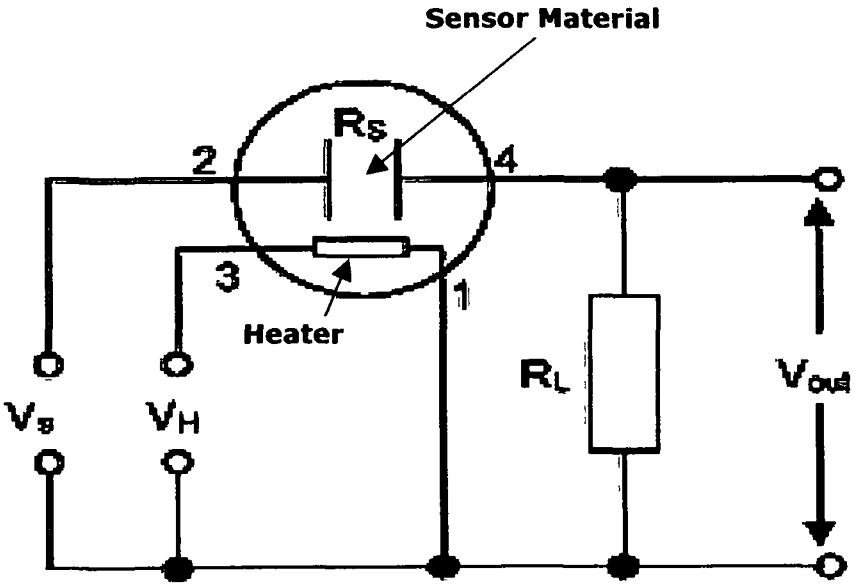


Figure 3.3: Schematic of a MOS sensor's basic circuit used to determine sensor resistance in E-nose sensor array.

3.3.2 Conducting Polymer Sensors

Many polymer materials are conducting (or semi-conducting) and show a variation in conductivity with absorption of different gases and vapours. Conducting polymers are very popular in the development of gas and liquid-phase sensors with *polypyrrole* and *polyaniline* being the favoured choices. Materials used to make conducting polymers tend to have some common features, including the ability to form them through either chemical or electrochemical polymerization and the ability to change their conductivity through oxidation or reduction. Conducting polymers are widely used as odour-sensing devices, the major reasons for this are the sensors show rapid adsorption and de-absorption phenomena at room temperature, power consumption is low, specificity can be achieved by modifying the structure of the polymer; they are not easily inactivated by contaminants and they are very sensitive to humidity [7].

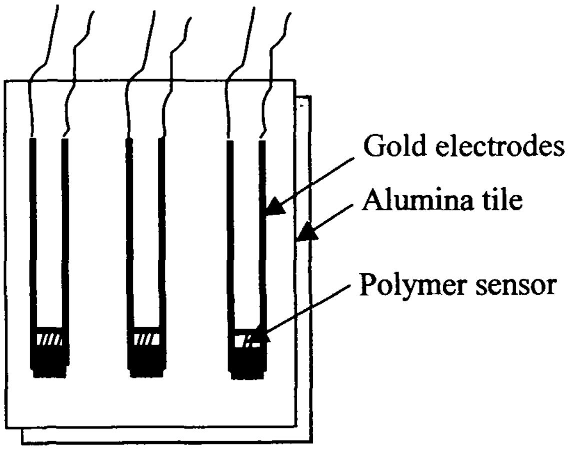


Figure 3.4: Conducting polymers electrode micro-array, electrode length 3 mm, electrode width 10 μm and gap width 10 μm approximately.

The nature of polypyrrole polymerization has been used to make resistometric sensors with interdigitated arrays (figure 3.4). The tracks are connected in parallel as the working electrode and a suitably oxidizing potential is applied in the presence of the pyrrole monomer [8]. The electrode grows until the gap between the adjacent lines is bridged, and the resistance across this gap is then used for sensing various analytes. For organic solvent-soluble conducting polymers (such as certain polyanilines), the polymer can be cast onto the interdigitated array. For resistance measurements, it is necessary to deposit conducting polymer along an insulating surface between the microelectrodes. This usually involves the use of lithographically coated microelectrode arrays in which the gap between the electrodes can be made as small as 5-10 μm reproducibly. Lateral growth is promoted by both silanization of the electrode surface and the addition of surfactants to the monomer solution during growth [9].

When a polymer film is exposed to a gaseous vapour, some of the vapour partitions into the film and causes the film to swell. The vapour-induced film swelling produces an increase in the electrical resistance of the film because the swelling decreases the number of connected pathways of the conducting component of the composite material. The detector films can be formed from conducting polymer composites, in which the electronically conductive phase is a conducting organic polymer and the insulating phase is an organic polymer. The films can also be from polymer-conductor composites in which the conductive phase is an inorganic conductor such as carbon black, Au, Ag, etc. The insulating phase is a swellable organic material. Any individual sensor film responds to a variety of vapors, because numerous chemicals will partition into the polymer and cause it to swell to varying degrees. However, an array of sensors, containing different polymers, yields a distinct fingerprint for each odour. The pattern of resistance changes on the array indicates type of the vapour, while the amplitude of the patterns indicates the concentration of the vapour.

Currently, with films on the order of 1 micron in thickness, the swelling (and therefore resistance) response times range from 0.1 sec to 100 sec. Characteristic patterns are produced even at times shorter than this, so the time-dependent swelling properties also provide diagnostic pattern information on the vapour of interest. More rapid responses to equilibrium can be obtained through reduction in the film thickness.

At small swellings, the film returns fully to its initial unswollen state after the vapour source is removed, and the film resistance on each array element returns back to its original value. The polymer-based electronic nose technology affords an inherent sensitivity advantage towards very low vapour pressure compounds in the presence of high vapour pressure materials. The commercial, off-the-shelf, organic polymers provide the basic sensor components, based on the differences in polarity, molecular size, and other properties of the vapour.

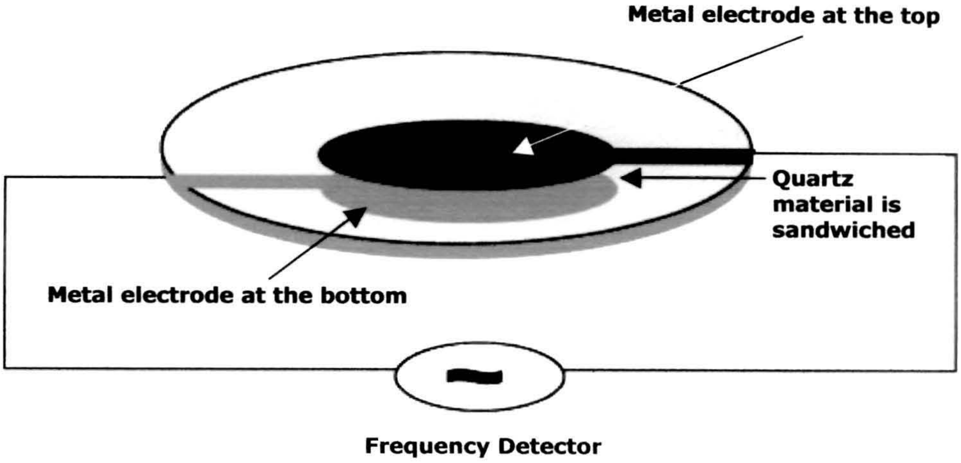

Metal electrode at the top
Quartz material is sandwiched
Metal electrode at the bottom
Frequency Detector

Figure 3.5: Typical thickness-shear mode (TSM) sensor with electrode connections.

3.3.3 Quartz Resonator Sensors

These sensors are comprised of a quartz crystal and a sensitive coating that is deposited on the surface of the crystal. Quartz crystals are typically operated in a resonance circuit to establish frequency, as in a quartz watch. In a sensor application, crystal's coating absorbs molecules from the gas phase, and the crystals change their fundamental frequency according to the mass increase of absorbed molecules. The absorption of molecules on the quartz depends on the partition coefficient of the molecule in the polymer coating deposited on the quartz and the concentration of analyte in the gas phase. If the partition is not concentration dependent, which is true for many polymers and analytes, the frequency change is proportional to the analyte concentration. These sensors offer excellent reproducibility and long-term stability, and, in addition, they do not need oxygen in the carrier gas. However, the sensors do exhibit a certain cross-sensitivity towards humidity.

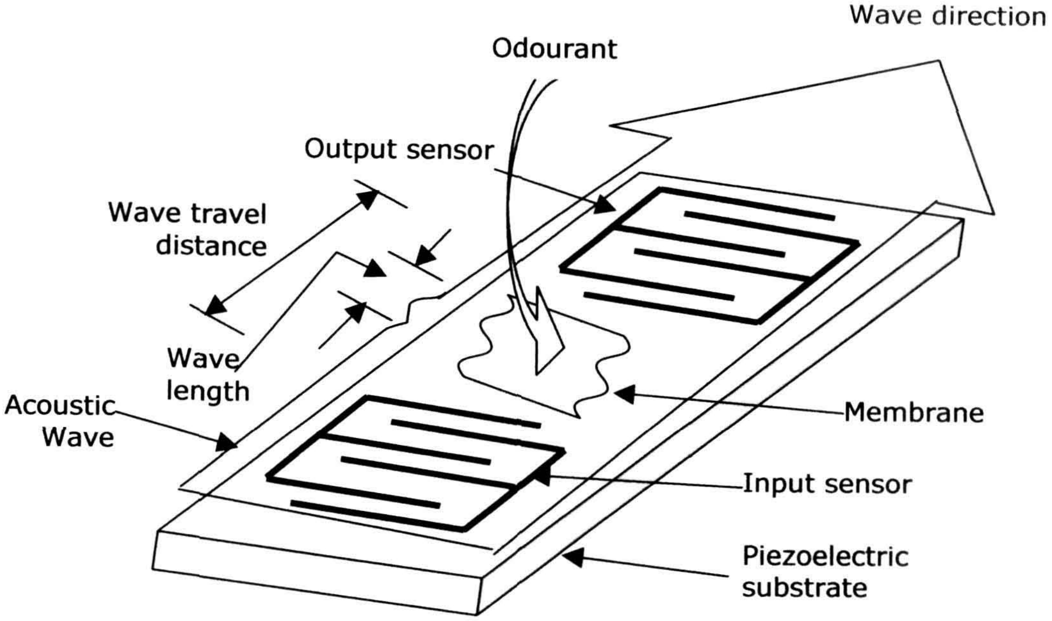


Figure 3.6: Surface acoustic wave (SAW) sensor with interdigitated electrodes.

AT-cut quartz crystals (+35°15' orientation of the plate with respect to the crystal plane) are favoured as piezoelectric sensors because of their excellent temperature coefficients. The crystal cut determines the type of acoustic wave generated in piezoelectric materials, thickness of the material used and by the geometry and configuration of the metal electrodes employed to produce the electric field [10]. One of the first sensors to be introduced was the thickness-shear mode (TSM) sensor, which, if the substrate is quartz, may commonly be termed the quartz crystal microbalance (QCM) or bulk acoustic wave (BAW) sensor.

A typical TSM sensor is shown in figure 3.5. The sensor consists of overlapping metal electrodes at the top and bottom and the device is normally 1.56 mm thick and 12.5 mm in diameter. This type can be used with up to 10 MHz fundamental resonance frequency with a standing resonant wave being generated where the wavelengths are related to the thickness. As the thickness increases (for example, due to added mass by deposition on the surface), the wavelength increases and the frequency decreases. Thus, the TSM can act as a mass-sensitive device. A major advance on the TSM sensor was the SAW version consisting of interdigitated electrodes fabricated on to quartz containing a thin film of material as shown in figure 3.6. When a potential is applied across the two halves of the interdigital transducer (IDT), a surface Rayleigh wave is launched in both directions across the surface from the IDT. Adsorption of odours to the coating membrane results in a change in mass and the acoustic wave is perturbed leading to a frequency shift [11]. This property can be used for pattern recognition since the shift in the frequency is different for the different odours. SAW sensors can be operated at higher frequencies than QCM sensors thereby leading to better sensitivities. Acoustic wave sensors can also be operated in the liquid-phase.

3.3.4 MOSFET Sensors

In the 1970s, improvements in semiconductor technology led to the development of a Field Effect Transistor (FET). This is a very high impedance transistor and the most sensitive measurements of small potentials requiring very low current flows are made using this technology. In the FET, current flows along a semiconductor path called the channel, at one end of which is a source electrode. At the opposite end is the drain electrode. The effective electrical diameter of the channel can be varied by application of a voltage to a control or gate electrode. The conductivity of the FET depends on the electrical diameter of the channel. A small change in gate voltage leads to a large variation in current from the source to the drain. This allows the signal to be amplified. For the Metal Oxide Semiconductor FET (MOSFET), the thermal oxidation process used to form the silicon dioxide layer on the silicon surface of the device also forms a double layer, which can induce a conducting channel in the silicon substrate. In the MOSFET, the conducting channel is insulated from the gate terminal by a layer of oxide. Thus, there is no conduction even if a reverse voltage is applied to the gate.

MOSFET sensors can be operated both with and without a reference electrode. These sensors employ a thin catalytic metal layer on top of a MOSFET. The sensors are small, measuring less than 1 mm x 1 mm, and have low power consumption. Manufacturing methods for these sensors are similar to those used to make integrated circuits, so the technology makes it possible to make these sensors in high volumes at low cost. The sensitivity pattern can be manipulated by changing the type, structure, and thickness of the catalytic layer, as well as the operating temperature of the device, which is 100°C to 180°C. Due to their operation principle, these sensors are relatively insensitive to external disturbances related to humidity. They are sensitive to a broad range of hydrogen-containing or polar compounds such as hydrogen, amines, aldehydes, esters, ketones, aromates, and alcohols. The sensors are robust and can provide a broad spectrum of information from highly complex gases and odors.

3.3.5 Other Sensors

Ion mobility spectrometry (IMS) has the ability to separate ionic species at atmospheric pressure. However, there is also a research underway to develop low-pressure IMS systems. This latter technique can be used to detect and characterize organic vapours in air. This involves the ionization of molecules and their subsequent drift through an electric field. Analysis is based on analyte separations resulting from ionic mobility rather than ionic masses. A major advantage of operation at atmospheric pressure is that it is possible to have smaller analytical units, lower power requirements, lighter weight and easier use [11].

Metal Silicon Carbide (MISiC) Field Effect Sensors represent a relatively new development. These field-effect sensors are similar in construction to the MOSFET, but are based on silicon carbide instead of silicon. This makes it possible to operate the MISiC sensors at much higher temperatures (up to 1,000°C) and thus increase the range of detectable compounds. This will enable a number of novel applications for chemical sensor arrays. In addition, silicon carbide is an extremely inert material, so the sensors can be used in rough, high-temperature environments. They also exhibit a fast response time of 1 millisecond.

Two recent developments in MS are atomic pressure ionization (API) and proton transfer reaction (PTR). Both are rapid, sensitive and specific and allow measurements in real-time. Additionally, they do not suffer the drift or calibration problems currently experienced by electronic noses. In API-MS ionization takes place at atmospheric pressure. Sample molecules pass through the region to be ionized by the transfer of a proton to produce ions [12]. The impact of volatile organic compounds (VOCs) on the environment has led to a growing demand for devices to detect these compounds. One of the most promising uses the proton transfer reaction

(PTR)-MS, with the first instruments just appearing on the market [9]. As a result of the advantages listed above, these may play an increased role in the future development of electronic noses. However, a limitation is the use of quadrupole MS, which has a modest mass resolution and can only monitor a single mass channel at any moment [13].

3.4 E-nose Applications

E-nose technologies have been developed for a wide range of military and civilian applications, but in general opinion, current e-nose applications have only scratched the surface of the potential impact of chemical sensing technology. Military applications have focused on the early detection of chemical warfare agents and explosives, while commercial applications include detection of food spoilage, stages of wine and beer fermentation, banana and egg freshness determination, reproducible formulation of fragrances and the detection and identification of bacterial infections. E-Nose technology has the ability to send a signal to an environmental control system where a central computer decides how to handle the problem, without human interaction. Additionally, it monitors and analyzes temperature, relative humidity, carbon dioxide, carbon monoxide, airborne particles, total volatile organic compounds, mold and pollen, ozone and radon in an environment.

3.4.1 Food Processing and Quality Determination

Aroma production during wine fermentation has been monitored during bioconversion. In this process, the profile formed as a result of yeast metabolism is complex, being composed of many compounds. These differ from each other in concentration, chemical and organoleptic properties and contribute to the overall muscatel aroma. A commercially available electronic nose (A32S Aroma Scan, Osmetech, UK) consisting of 32 organic conducting polymer-based sensors was used [14]. Data analysis was by PCA. The electronic nose was able to discriminate samples based on aroma content.

Electronic noses have also been used for the quality determination of fruit and vegetables. An electronic nose based on arrays of differently coated quartz microbalances (QMB) has been used to discriminate between VOCs formed during the post-harvest ripening of apples. Both qualitative (type of apple) and quantitative identification were possible [15]. Warmed-over flavour (WOF) in beef was evaluated by 32 conducting polymer sensors. The meat was processed by vacuum cook-in-bag/tray technology (VCT) and stored in a refrigerator [16]. The VCT process involves treatment at 50 °C for 390 min. The study indicated that an electronic nose can be used for WOF odour identification in beef and may be a complementary approach for current sensory analysis. The ham-drying process involves several time-temperature interactions and these lead to variation in flavour. Use of an Aroma Scan A32/50S (AromaScan plc, Crewe, UK) electronic nose (32 sensors) allowed the differentiation of Spanish Serrano dry-cured hams processed for 12-month periods [17].

French researchers have coupled gas chromatography (GC) with an electronic nose to identify alcoholic beverages [18]. High concentrations of ethanol tend to affect the detection of aroma compounds by an electronic nose. Tequila, whisky, vodka and red wine were analysed and four compounds responsible for off-flavour in red wine were detected [19]. PCA allowed discrimination between the four types of beverages and wines from different regions.

Electronic noses have been used in the discrimination of extra virgin olive oils by identifying the geographical origin of these products. Similarly, vegetable oils can be classified by chemo-metric treatment of the data obtained from a gas sensor array. Products derived from Iberian breed pigs and different types of Italian cheeses have been classified by this method

3.4.2 Other Applications

The sense of smell is an important sense to the physician thus an electronic nose has applicability as a tool for diagnosis. An electronic nose can examine odours from the body (e.g., breath, wounds, body fluids, etc.) and identify possible infections and diseases. Odours in the breath can be indicative of gastrointestinal infections, sinus infections, diabetes, and liver problems. Infected wounds and tissues emit distinctive odours that can be detected by an electronic nose. The Odours of body fluids can indicate liver and bladder problems. E-nose for examining wound infections has been tested at South Manchester University Hospital [20].

A more advanced application of electronic noses has been proposed for tele-surgery [21]. While the inclusion of visual, aural, and tactile senses into tele-present systems is widespread, the sense of smell has been largely ignored in the past. The electronic nose is being potentially used as a key component in an olfactory input to tele-present virtual reality systems of tele-surgery. The electronic nose can identify odours in the remote surgical environment. These identified odours would then be electronically transmitted to another site where an odour generation system would recreate them.

In chemical industries, at present, the leak detection and monitoring techniques used are resource intensive and cumbersome. Leading chemical manufacturing firms are evaluating the E-nose technology for use in the development of products to detect leaks in pipelines and storage containers. E-nose detectors may also provide in-line quality control and quality assessment in industrial, packaging, automotive, and petrochemical processing.

Environmental applications of electronic noses include analysis of fuel mixtures [22], detection of oil leaks [23], testing of ground water contaminations, and identification of household odours [24]. E-noses can identify air pollution and toxic spills. E-noses can spot explosives and drugs.

The effect of different colouring agents ('taints') on the sensory properties of packaging materials has been determined by using E-nose [25]. The electronic nose has efficiently grouped these materials depending on colouring agents or lacquering. E-nose has been used for monitoring Space Shuttle Air for possible contamination [26].

As a safety device, the E-nose has a lot to offer to check for gas buildups in offshore oilrigs. Sanitation workers would benefit by knowing if any poisonous gases have collected down in the sewers.

Engineering students at Yale University have developed, Robots equipped with smart E-noses, called robotic dogs, to sniff out toxins. They sniff, fetch and run in packs. Here at Yale, E-nose technology has a social factor too. 'These dogs are programmed into instruments for social activism and services. Robotic technology is increasingly being applied to repetitive factory tasks or dangerous work such as defusing bombs or finding victims in collapsed buildings.

3.5 Electronic-nose as an Intelligent Instrumentation System

Intelligence, in psychology, is the general mental ability involved in calculating, reasoning, perceiving relationships and analogies, learning quickly, storing and retrieving information, using language fluently, classifying, generalizing, and adjusting to new situations. Alfred Binet, the French psychologist, defined intelligence as the totality of mental processes involved in adapting to the environment. Although there remains a strong tendency to view intelligence as a purely intellectual or cognitive function, considerable evidence suggests that intelligence has many facets.

Some of the instruments are called intelligent based on the fact that there exists strong analogy between psychological intelligence and the way instruments function. These instruments are endowed with logical power and self-initiated follow up actions under given conditions similar to humans. E-nose sensor is one of the intelligent instrumentation systems.

Nano-technology has yielded tiny, low cost, low power sensors. Tiny is important because they can be scattered around unobtrusively to measure just about everything that one can imagine. Low power is important because these don't need to carry a large battery and even are solar-powered. Low-cost is utter requirement because the numbers required are enormous. Several new companies are already producing ultra low-power, postage stamp-sized smart E-nose sensors, which yield good results in a variety of applications. But the thumbnail size is still only an interim stage – soon integrated E-nose sensors will yield microscopic components, which can be scattered around like "smart dust". The vast intelligent arrays can yield individual hot spots instantly for real-time analysis. In addition, it provides information to facilitate overall pattern recognition analysis in ways that were previously unimaginable.

Tiny, low power E-nose sensors make the measurements which are communicated and coordinated through peer-to-peer links, the technology moves to whole new software arenas of pattern-recognition, heuristic analysis, self-organizing systems, and complexity science. Soon it will be possible to communicate smells and fragrances to distant targets. E-nose has the considerable intelligence matching with humans.

3.6 Design and development of Prototype E-nose Setup

This research work was primarily aimed at designing and developing prototype E-nose setup for the tea and spice quality determination and flavour discrimination. The prototype E-nose system has been used successfully for number of experiments on tea quality determination. It has also been used for various hydrocarbon classification and spice aroma discrimination.

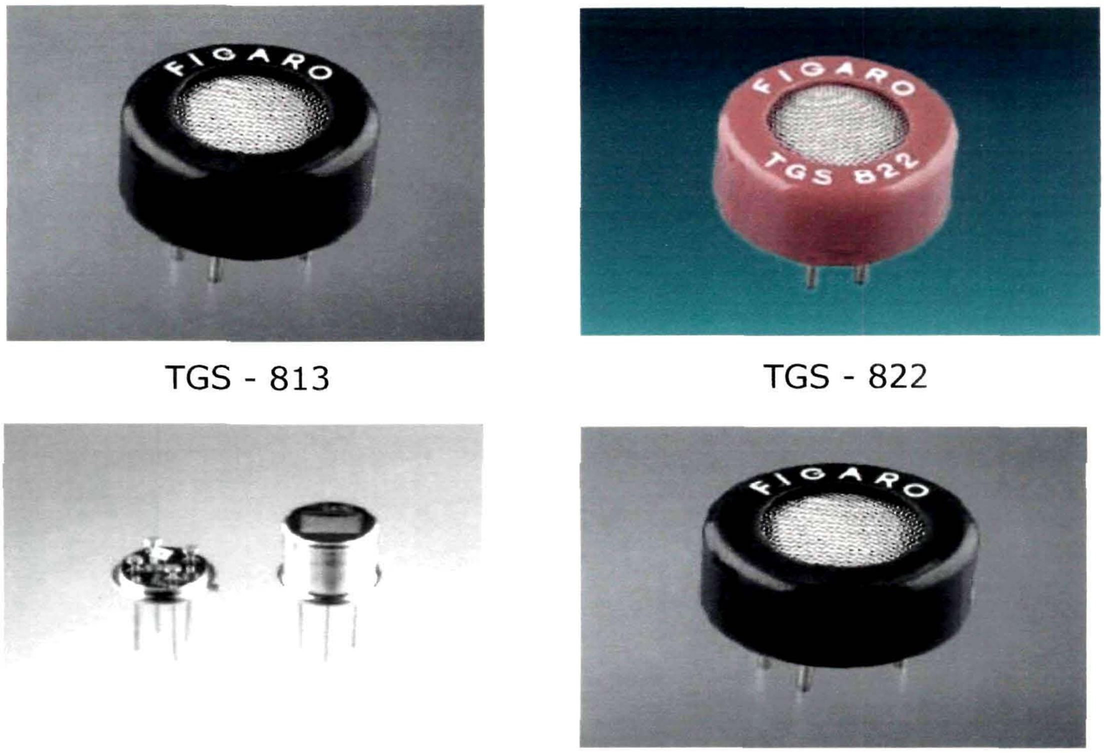
Initially, experiments were done at the University of Warwick [27] on overlapping tea flavour discrimination, and the result was conforming to tea samples based on intentional manufacturing defects. In the next part of the experiments tea samples were chosen from the standard tea favour notes resulting in non-overlapping and high accuracy classification rates [28].

3.6.1 Materials and Components for E-nose Setup

Main components of the E-nose design were sensors. Four MOS based gas sensors, TGS-2611, TGS-842, TGS-822, and TGS-813-J01 from FIGARO INC, Japan (Table 3.1) are used for E-nose setup to carry out experiments for tea and spice flavour determination. These sensors are shown in figure 3.7. The experimental setup consists of these sensors as E-noses and a diaphragm pump system with flow control devices with suitable PC control for switching between sample headspace and base room environment. E-noses are housed in a large metal chamber to shield it from electromagnetic effects as shown in the figure 3.8.

The sensors were selected on the basis of sensitivity to different odours such as cooking vapors, alcohol, volatile organic compounds (VOCs) etc [29]. Four sensor chambers are connected through plastic pipelines to the flow path junction that in turn is connected to the sample and base room environments through flow control devices. Flow control devices provide constant flow of sample vapour and air to the sensors through connected pipelines.

Switching between reference vessel and sample vessel is by PC with controlled data acquisition software and GENIE (American Advantech® Corporation) based GUI [30]. Ideally, the reference and sample vessels are placed in room environment at about 25^0 C temperature and 65% relative humidity with deviation less than ±2%.



TGS - 813

TGS - 822

TGS - 2611

TGS - 842

Figure 3. 7: Four Taguchi Gas Sensors from Figaro Inc. Japan, used in this Research Work.

Table 3.1: Details of the Sensors used in Experiments.

Sensors	Manufacturer	Specific Use
TGS 2611	Figaro Engineering Inc.	Methane, Combustible gas detection
TGS 842	Figaro Engineering Inc	Methane, Combustible gas detection
TGS 822	Figaro Engineering Inc	Organic solvents such as alcohol, toluene.
TGS 813	Figaro Engineering Inc	Domestic and Portable gases

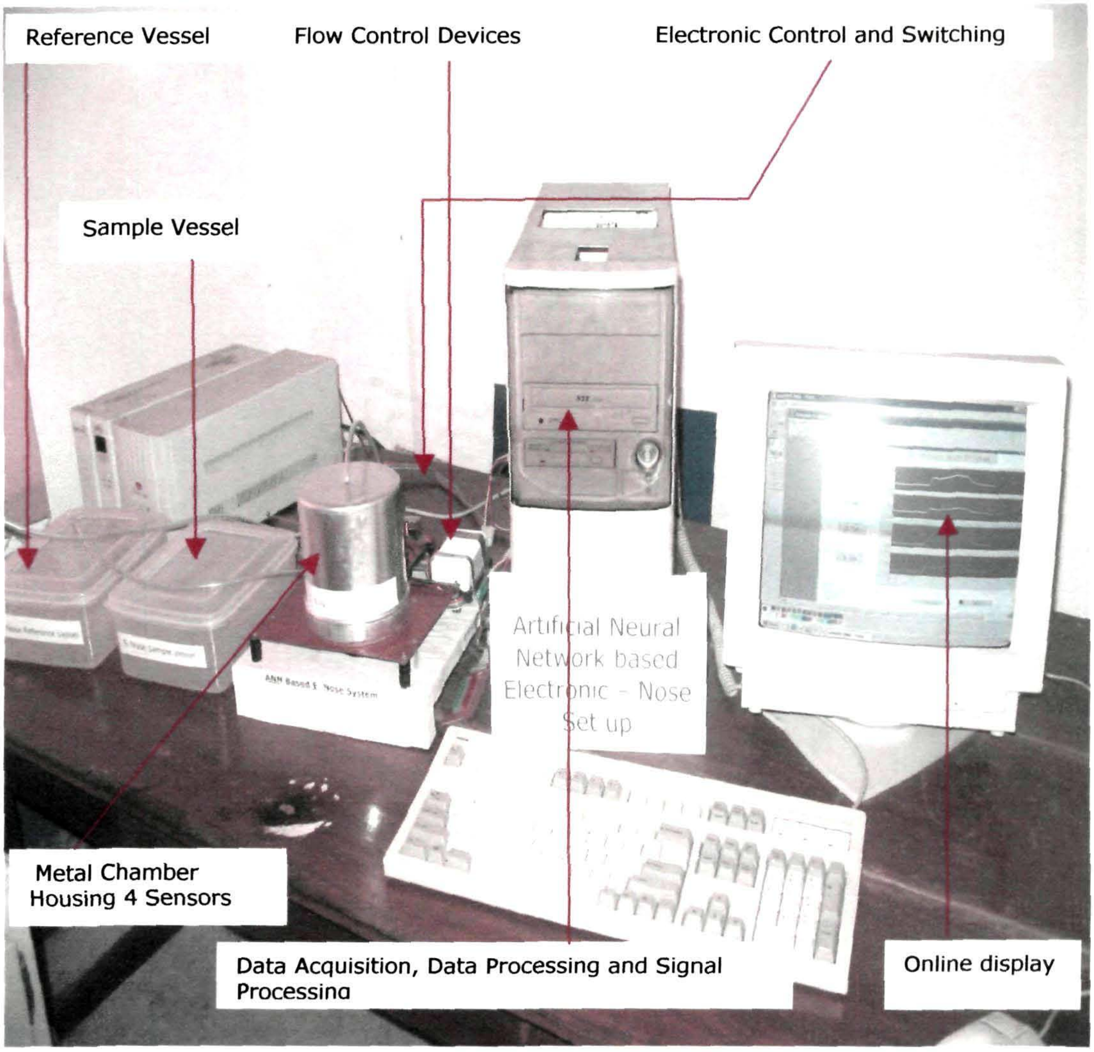

Reference Vessel
Flow Control Devices
Electronic Control and Switching
Sample Vessel
Artificial Neural Network based Electronic - Nose Set up
Metal Chamber Housing 4 Sensors
Data Acquisition, Data Processing and Signal Processina
Online display

Figure 3.8: E-nose System developed in the Department of Electronics Laboratory, University of Tezpur, India

The switching circuits are designed and converted into hardware using 12 VDC 150 Ω No. 9744B relay for the alternately switching between sample headspace and reference room air. Switching time was controlled by GENIE (American Advantech® Corporation) based GUI design which was programmed as user customizable interface. The ICs used for power switching are BC 558, BK 042 and power transistors. Two diaphragm pumps (SO-FINE-AQUA-1) are used for supplying a constant flow of sample vapour and room air as desired by the programmed options. Data acquisition and online logging of the signals from E-nose sensors is achieved by PCL-208 High Performance Data Acquisition Card from Dynalog (India), Ltd, Mumbai- 400086; India [31]. PCL-208 card has also been used to condition the data in required format and convert into digital form to make compatible with computer processing. A logical process indicator was designed to indicate whether the E-noses are responding to room air or to sample headspace. The photographic view of the E-nose setup is depicted in the figure 3.8.

3.6.2 Headspace Generation

Headspace refers to the technique in which sample vapour is generated and subsequently it is blown with certain flow rate on to the sensor surface to produce E-nose response. Stainless steel containers are used to place the tea and spice samples in the temperature and humidity controlled chambers. The steel containers are specially modified to suit flow pipeline connections in the experiments. High quality plastic pipelines and junctions (3-ways and 4-ways) are used from reference and sample environments to the sensor array. Thermometers and hygrometers were used for monitoring temperature and humidity of the ambient air at the time of conducting experiments.

3.6.3 Diaphragm Pump Flow System

The flow rate of the tea and spice vapour is controlled with the help of two diaphragm pumps (SO-FINE- AQUA-1). These pumps are electrically operated and switching time for the sample vapour and the airflow is controlled with the help of PC based program and intractable GUI. The flow rates are assumed to be constant throughout the experiment. A typical flow control diagram is shown in figure 3.9. A circuit diagram for the flow control and switching between sample and reference vessels is shown in figure 3.10.

In case of overlapping tea samples, the sample vapour flow rate is maintained at about 2 liters per minute. Sample vapour is transported for 5 minute to the sensor surface followed by 5 minute fresh air blow to refresh the sensor surface or to make sensor surface free of any residue of tea vapour. The sample vapour removed from the sample vessel was replaced by room air, which is assumed to be contaminant free. The plastic reference vessel is filled with pure water. This ensures that water vapour contents are almost equal in the tea sample vapour (tea is prepared in boiling water) and room air. The room air was pumped into the sensor chamber through this reference vessel. The sensors were allowed to return to their baseline level over a period of 20 minutes, after sampling the headspace of the tea vessel. This ensured that the E-nose system had responded only to the tea flavour rather than to any residual smell of the plastic vessel or only to the different environment in the reference plastic vessels. Figure 3.9 shows the experimental set-up.

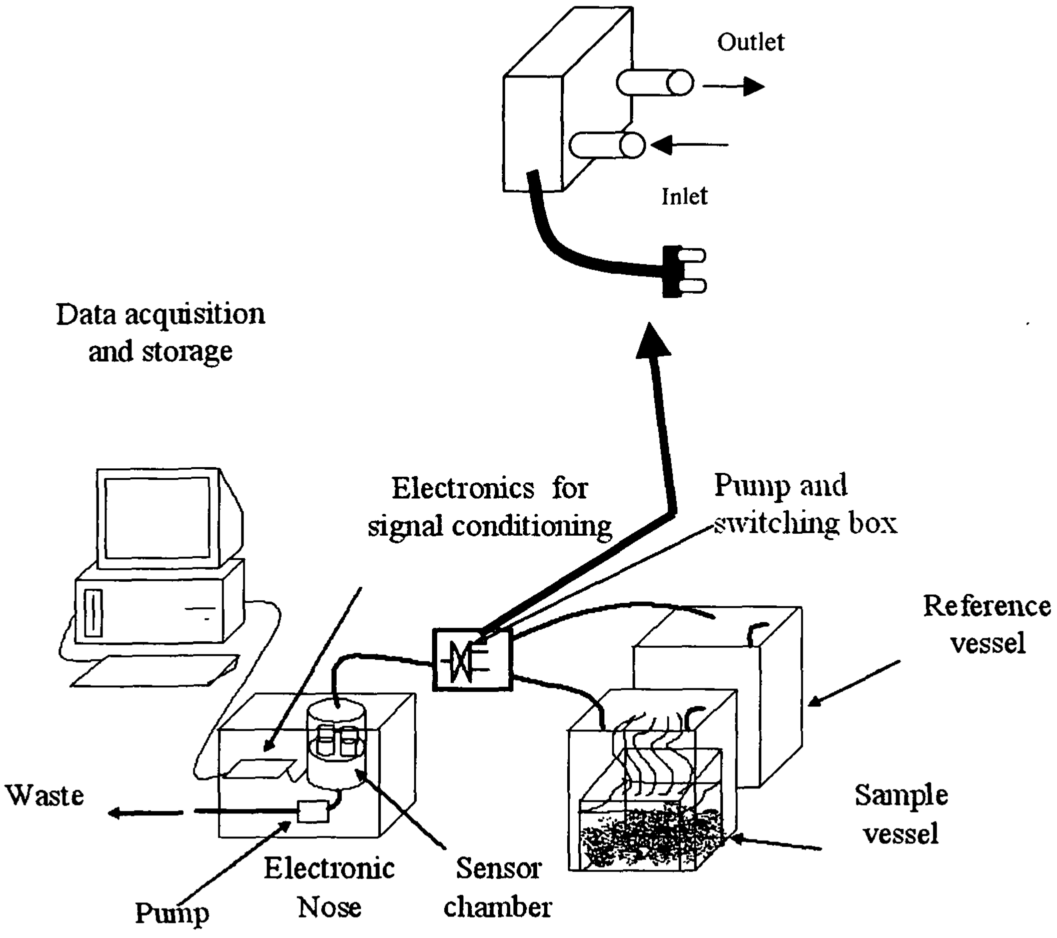


Figure 3.9: Flow control valve setup for the MOS based E-nose system for tea experiments

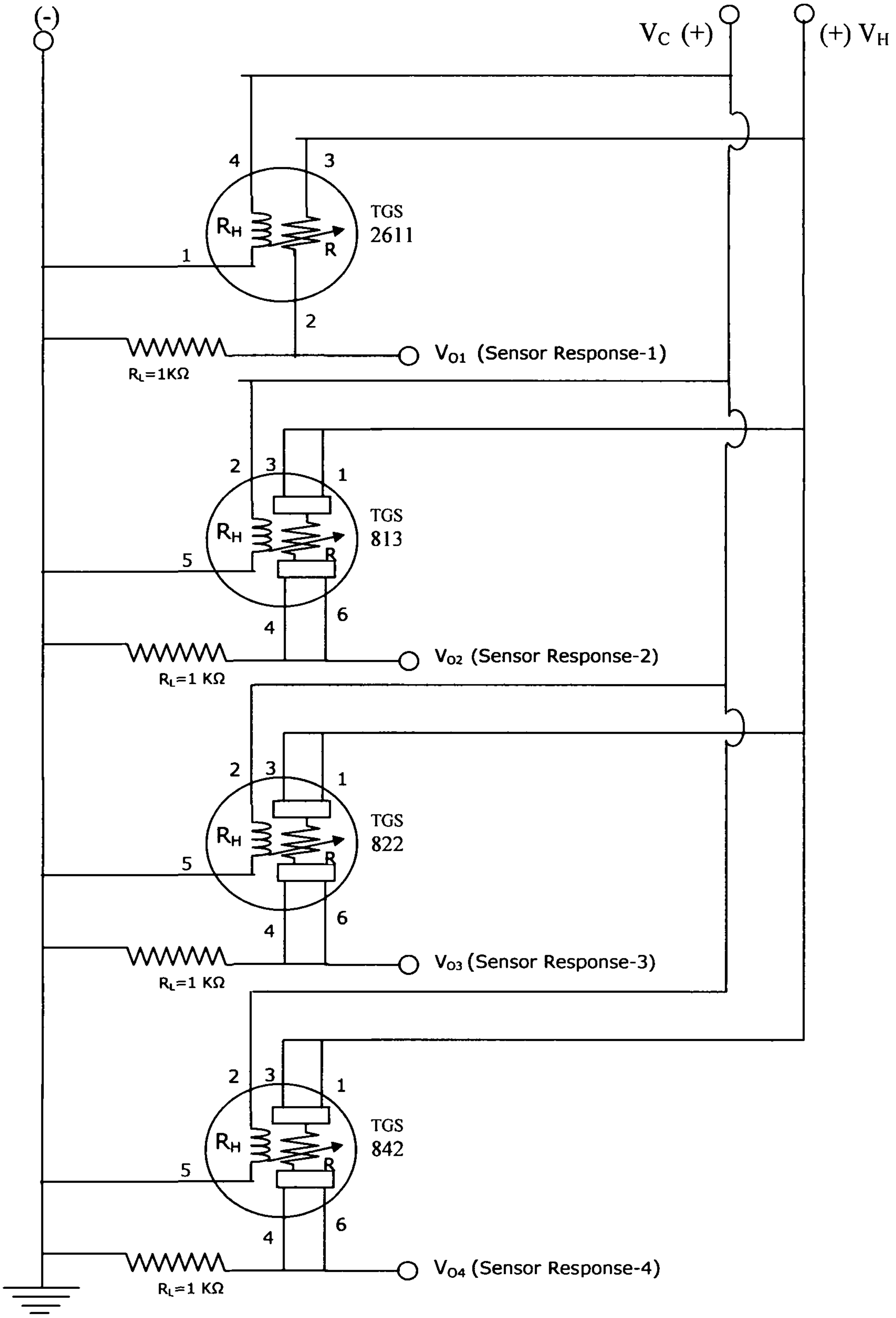


Figure 3.10: Measurement Circuit for the E-nose sensor responses

The temperature and humidity values in the laboratory were typically 25° C ± 2 and 65 % ± 2 RH, over the period of the experiments. One measurement comprised of sampling, alternatively, a headspace sample of the tea vessel followed by the sampling of headspace of reference vessel. However, the temperature and humidity values in the laboratory were varied according to the experimental requirements while determining the drift coefficients. Sensor response measurement rate was maintained at 1 Hz. During the measurements, resistance of each sensor was also measured after every 5 second and stored in a data file for subsequent processing of data response for classification. It can be recollected that output of the sensors' response is voltage signal, however, resistance is the physical quantity which changes in the presence of sample vapour (tea VOC's in this case). This process was repeated for each of the 5 tea samples of overlapping nature.

3.6.4 MOS Sensor Response Measuring Circuit

The application circuit diagram for measuring the MOS sensors response is shown in the figure 3.10. The following resistances and power supplies, which are indicated in the figure 3.10, are used in the application circuit.

Sensor Resistance (R)

This is characteristic resistance of the sensor surface. It is normally defined by the manufacture in a range under standard ppm of certain sample and other circuit conditions of load resistance and power supply (V_C). It's value changes on exposure to the sample vapour. The typical values are 1 - 5 k Ω (TGS-2611), 5 – 15 k Ω (TGS-813), 1 - 10 k Ω (TGS-822) and 3 - 15 k Ω (TGS-842). The change in resistance of the sensor surface generates output response. Other specifications defined by the manufacture for these 4 MOS sensors are given in appendix 6.

Load Resistance (R_L)

These resistances are chosen by user, subjected to the limitations that power dissipation does not exceed the limit specified by the manufacturer for the sensors. The resistance values are calculated for the 4 sensors based on the power dissipation limitation under worst condition of the circuit parameters i.e. when a maximum of current flow takes place under minimum sensor resistance (R_S). The values are indicated in the circuit diagram of figure 3.10 and listed in table 3.2.

Circuit Power Supply (V_C)

Circuit power supply is specified by manufacturer to a maximum of 24 Volt for TGS-813, TGS-822, TGS-842 and 5 Volt for TGS-2611 (appendix-6). In this circuit, V_C is 5 volt normally, but for some of experiments it is 10 volt (except for TGS 2611).

Heater Circuit Power Supply (V_H)

Heater power supply is uniform value of 5 volt for all the sensors. This power supply keeps sensors at elevated temperature to have a better response.

Output Voltage (V_O)

Output voltage is measured as sensor response across load resistance R_L. This voltage varies from a fraction of a volt to few volts (maximum of 5 volt).

Power Dissipation (P_S)

Maximum power dissipation takes place when sensor resistance (R) is minimum for chosen load resistance and power supply. The worst condition (maximum power dissipation) is listed in table 3.2 along with maximum power allowed in accordance with manufacturers specifications. The power dissipation is given by equation 3.3.

$$P_S = (V_C^2 \text{ x } R_S) / (R_S + R_L)^2 \qquad (3.3)$$

Table 3.2: Sensor parameters of experiments.

Sensor	V_H (volt)	V_C (volt)	R (k ohm) (Range)	R_L (k ohm)	Max. Power Dissipation (P_S), for R (min) (Worst condition)	Max. Power Dissipation allowed
TGS-2611	5	5	1 - 5	1	12.5 mW	15 mW
TGS-813	5	5 or 10	5 - 15	1	3.47 mW	15 mW
TGS-822	5	5 or 10	1 - 10	1	12.5 mW	15 mW
TGS-842	5	5 or 10	3 - 15	1	4.68 mW	15 mW

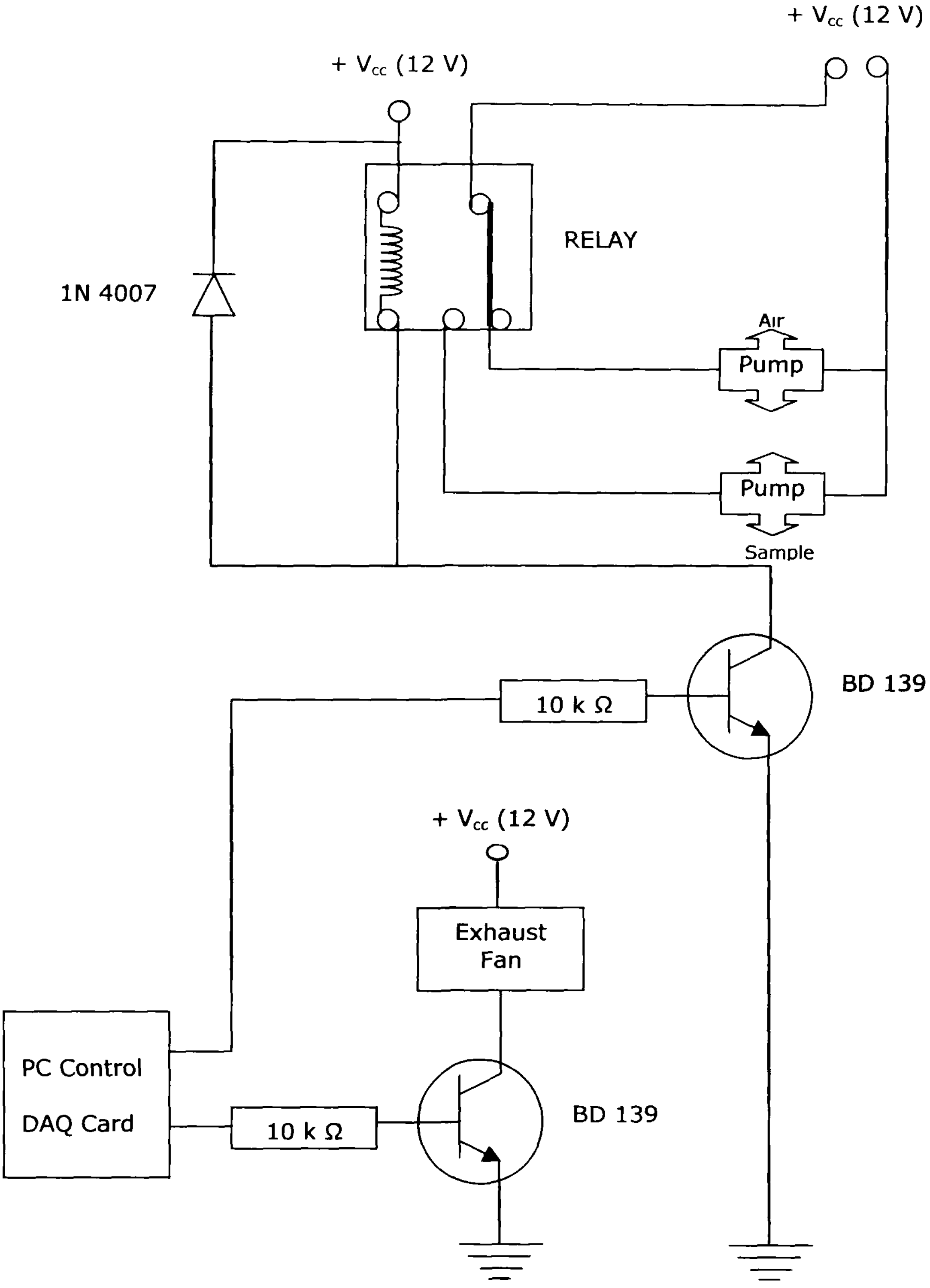


Figure 3.11: Diaphragm Pump control driver circuit for sample vapours and room airflow

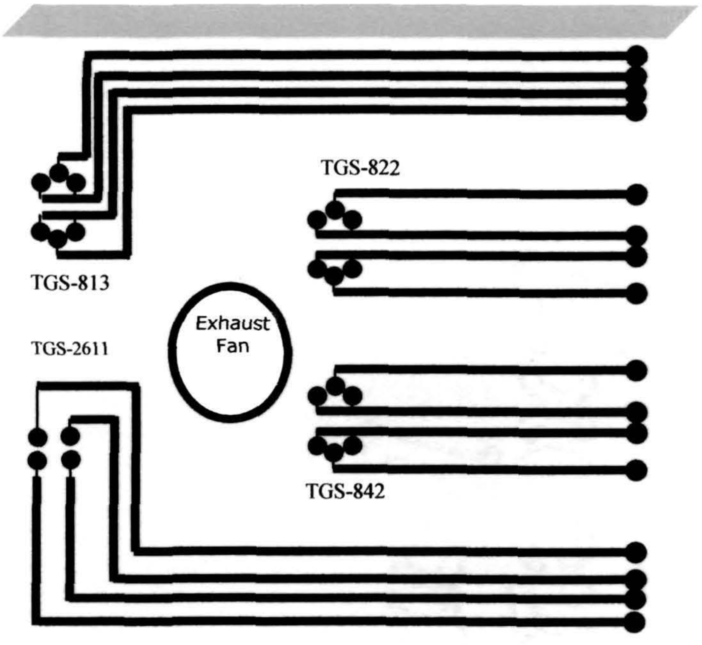


Figure 3.12: PCB Design for the Sensor layout.

3.6.5 PCB Layout

A Printed Circuit Board (PCB) layout (7" x 9") is shown in figure 3.12. The exhaust fan draws out used sample vapours of tea and spice, from sensor chambers. The PCB is prepared by masking and chemical itching method. Four sensors are laid on the board in small plastic chambers with inlet for sample vapour and air, and outlet for exhaust. Circuit diagrams for the sensor response measurement and switching control circuits are shown in figures 3.10 and 3.11 respectively.

3.6.6 Complete E-nose system

All E-nose components are laid out on a single board. Two high quality plastic chambers are used as sample and room air vessels. Layout is shown in the figure 3.13 and 3.14. are used for the flow of air and vapour. The sample vapour and air flow is directed through two diaphragm pumps, which are connected through plastic pipelines. A small separate PCB is prepared as a driver circuit for diaphragm pumps. The switching between sample vessel and reference vessel at pre-defined time intervals is achieved with the help of the PC. A process indicator, as a visual aid, is prepared to indicate the status of the diaphragm pump. Exhaust fan is laid out on the board to regulate the vapour flow by removing used air and sample vapours from it.

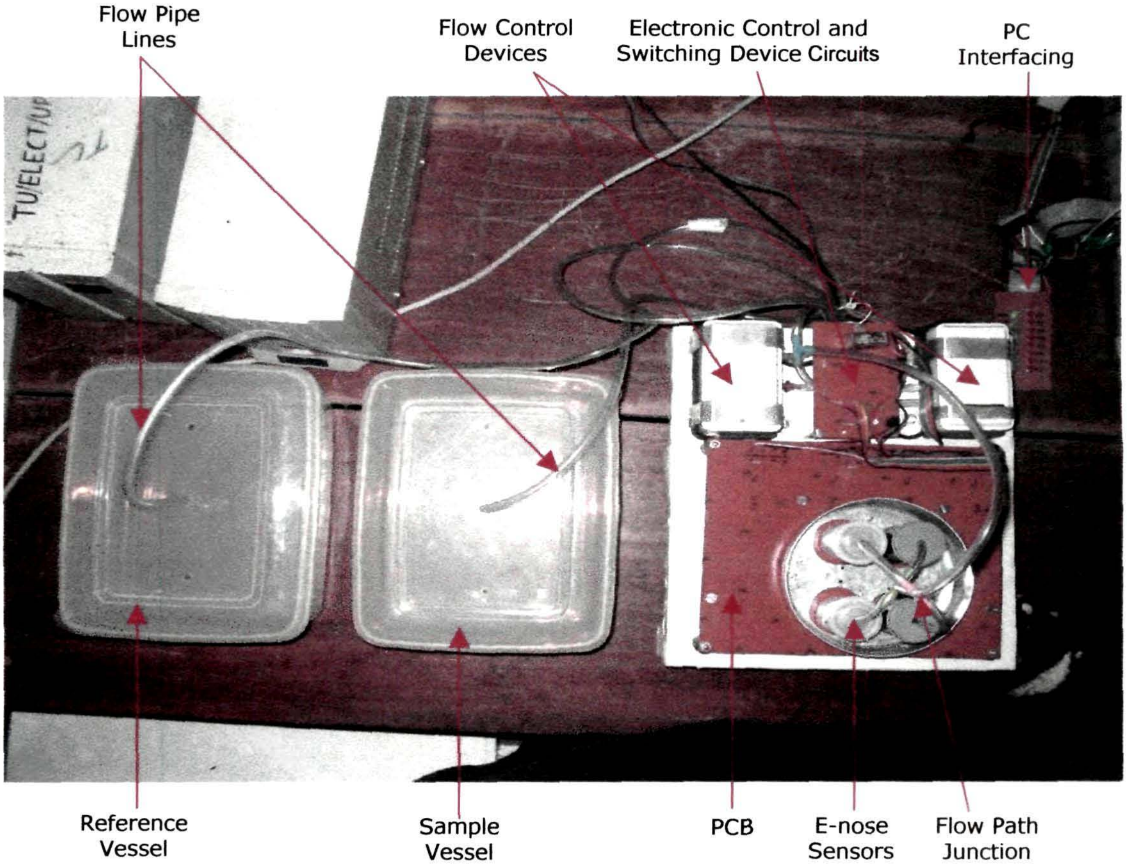


Figure 3.13: Onboard Integration of Electronic-nose Components (top view)

3.6.7 Interfacing E-nose system with the PC

The figure 3.14 is shown with a PENTIUM based PC interfacing with the E-nose system, including controlling and application circuits. The main idea behind interfacing is the online acquisition of E-nose sensor response in real time and providing control signals to the E-nose electronic circuits for various operations such as switching and time setting. The interfacing is done with the help of a data acquisition card, PCL - 208, Dynalog (India), Pvt. Ltd. Bombay (Appendix – 5).

The PCL-208 is a low cost AD/DA card, which is highly versatile in the field of the industrial process automation and for the low-end data acquisition needs.

Although limited in overall speed, it has a very good accuracy and a reasonably good signal conversion (AD to DA and vice versa) speed, making it cost effective alternative to the costlier AD/DA cards. It has input voltage range as bipolar ± 5 V with accuracy 0.015 % of reading and ± 1 bit. Data transfer takes place by program control. IC AD7541 performs DAC, which is monolithic multiplying type. PCL 208 is compatible with GENIE (American Advantech® Corporation). It has provided powerful and easy to use software driver routine of GENIE configuration.

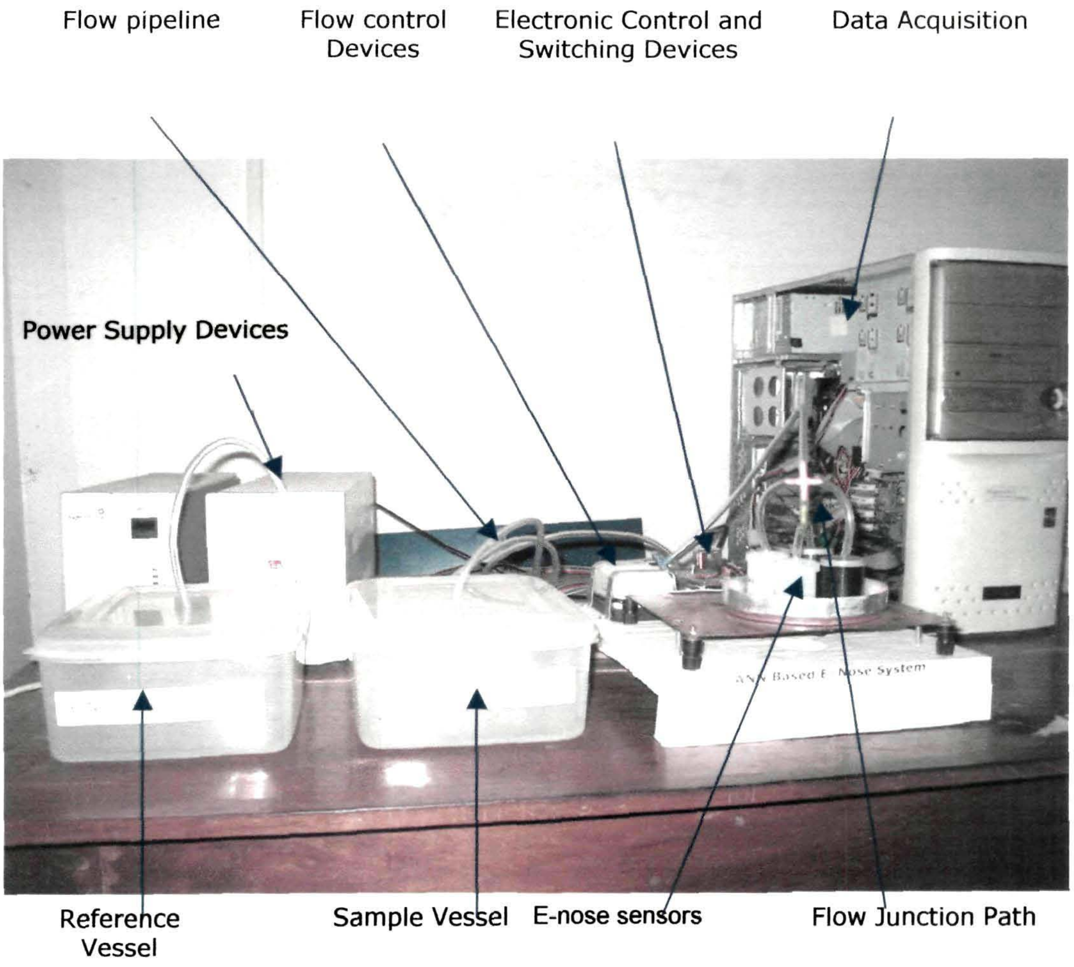


Figure 3.14: Electronic-nose setup Interfacing with PC.

The PCL-208 uses successive approximation method of analog to digital conversion and offers 12-bit resolution with 25-microsecond conversion time. 8 single ended input channels are available for use. In addition to the 8 channels of ADC, the card also provides a single channel 12-bit DAC output.

The output range is 0 to +5 volt and 0 to +10 volt. First 4 channels, 1 to 4, are used for on readings from 4 E-nose sensors. These 4 channels are configured to display the E-nose sensor reading status in GUI display window graphically and numerically. Next 2 channels from 5 to 6 are used to indicate the onboard status of pumping system, which switches alternately for configured time durations for sample and room air.

The analog output from PCL 208 is used for switching operations. It has been demonstrated in figure 3.10 that exhaust fan is also configured to run only during refreshing period. Using IC BD 139 and 1N 4007 supporting circuits are designed to function with the help of the interfacing card. A process indicator is interfaced to indicate whether E-nose is sampling VOC's or it is logged on to fresh air for refreshing E-nose. Power supplies of different polarities (5 Volt, ±10 Volt and 230 volt normal domestic supply) are used in accordance with requirements.

3.7 E-nose Data Acquisition

In the first part of the experiments, the data acquisition and storage system was controlled using LabVIEW software [32] (National Instruments Corporation, 1998). A PC-LPM-16 PnP Data acquisition card was used for online data gathering [32] (National Instruments Corporation, 1998).

In the second part of the experiments, GENIE software is used for online logging and displaying results. Data acquisition was continuously performed for long timescales to get widely spaced response from the E-nose sensors. A long series of experiments were conducted to gather data for processing. Temperature was maintained at room environment at about 25^0 C and 65% relative humidity with deviation less than ±2%. Each data set was acquired for 200 seconds and 300 seconds was allowed for refreshing to air.

3.7.1 Introduction to GENIE Environment

GENIE is a comprehensive application development tool for data acquisition and control. It supports functions and utilities to develop automation applications for the use in Windows 3.x and MS Windows environments. System Monitor Displays and Dynamic Operator Displays are configured in GENIE.

GENIE provides advanced programming features and tools in maintaining ease of use. A library of Icon Blocks representing data acquisition and control, mathematical and control functions are provided through ask designer. Display Designer provides a variety of graphic object to design monitoring and control displays. Report Designer features a configurable format design utility and scheduler to generate reports automatically. In addition to the feature listed above, GENIE's built-in VBA compatible programming tools strengthen its ability to perform calculation or logical analysis.

The GENIE packages consist of two major software modules and several utility programs. The two major executables are GENIE Strategy Editor / Runner (GENIE.EXE) and GENIE Runtime only program (GWRUN.EXE). GENIE.EXE is for designing and testing Strategies. GWRUN.EXE is for running strategies in a live environment. GWRUN.EXE uses fewer resources and achieves better runtime performance because it does not perform validity checks on objects and links between objects, it does not allow any changes to be made to the strategy being run.

3.7.1.1 GENIE Architecture

GENIE 3.0 has a modular-oriented, open integrated architecture, as shown in the figure 3.15. The open plateform allows easy integrating GENIE with other applications to share real-time control data. The performance and number of I/O blocks GENIE can support are increased significantly through this new architecture. The new architecture is depicted in the figure 3.15.

Data center is responsible for data acquisition and control data. It manages all GENIE real-time data and provides three sets of interfaces to the outside world, DDE & OLE Automation and C API. Through these three interfaces, other applications can retrieve or input data to GENIE.

I/O Driver is accessing real-time data from hardware equipment. GENIE I/O drivers cover all Advantech Industrial automation hardware, including plug-in DA &

C cards, PC-based modular controller MIC-2000, ADAM-4000 remote modules and ADAM-5000 distributed modules.

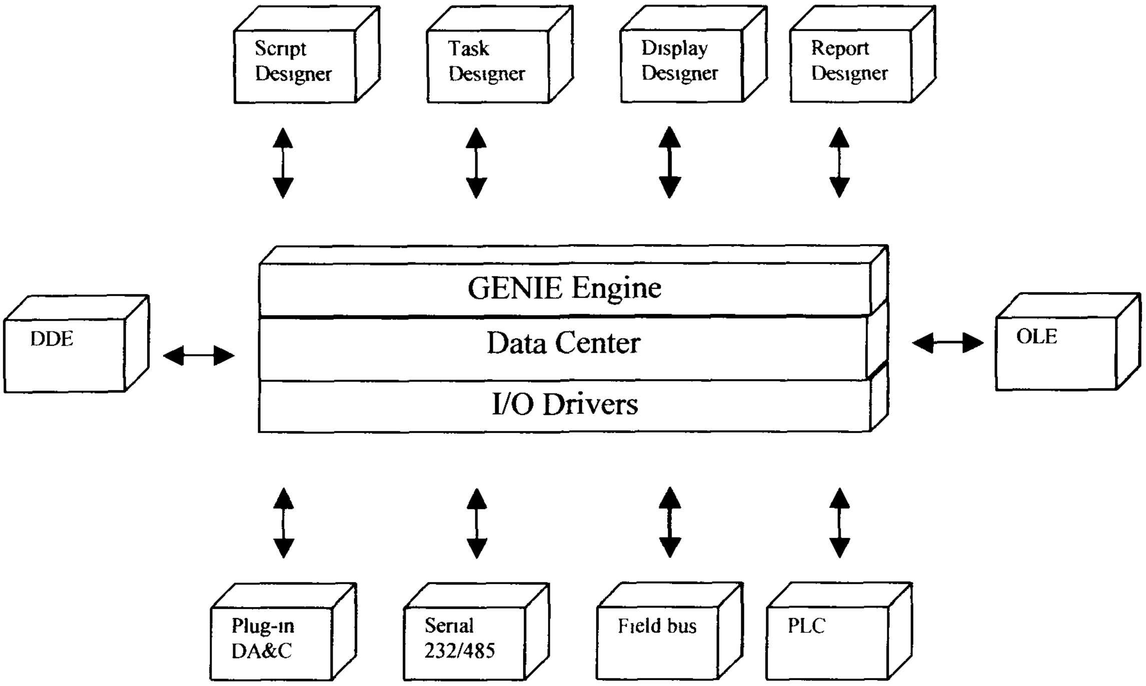
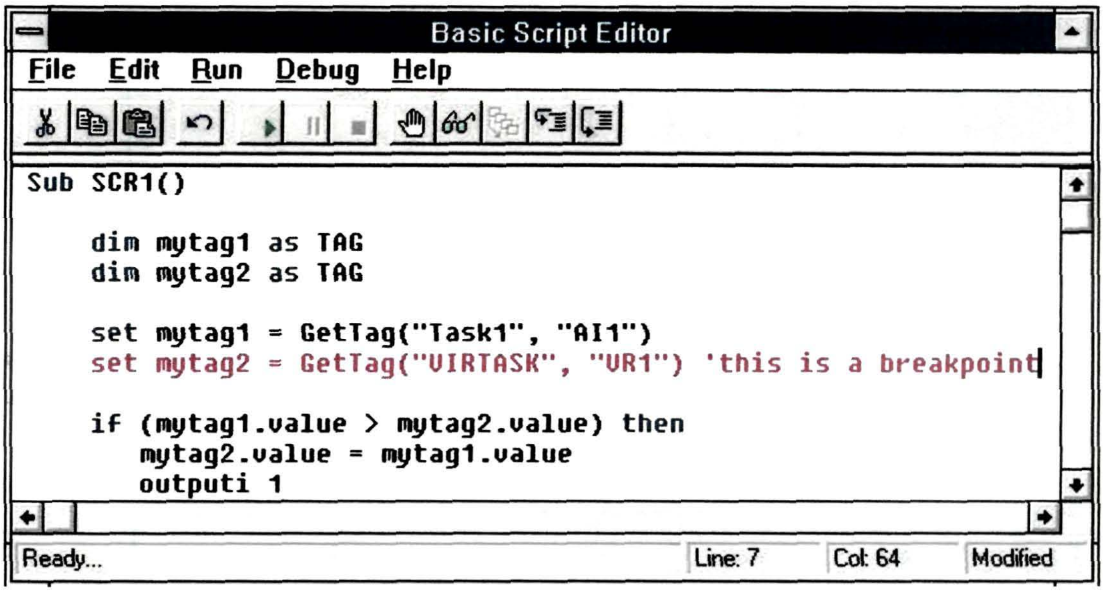


Figure 3.15: GENIE 3.0 System Architecture

3.7.1.2 GENIE Basic- Script Designer

GENIE Script Designer is a VBA-compatible basic script engine. Script Designer not only features a robust Basic programming engine, but also includes many tasks and real time data access functions. Through this script engine, users can call DDE, OLE and ODBC (SQL) functions to integrate with other applications.

The script designer is basically a text editor with some convenient features for editing script code. The script source will be compiled into p-code after editing so it won't need to be compiled again at run-time. The syntax of the Basic-Script is compatible with Microsoft VBA (Visual Basic for Application in Excel, Word, Access, etc.) and Microsoft Visual Basic. It is possible to take a Visual Basic source code and compile and execute it under Basic-Script without changing a word if only common functions are used. The two development environments are very similar. In the source code design stage it features cut, copy and paste functions. Figure 3.16 is a typical script editor where programming codes are written.



```
Sub SCR1()

    dim mytag1 as TAG
    dim mytag2 as TAG

    set mytag1 = GetTag("Task1", "AI1")
    set mytag2 = GetTag("VIRTASK", "VR1") 'this is a breakpoint

    if (mytag1.value > mytag2.value) then
       mytag2.value = mytag1.value
       outputi 1
```

Figure 3.16: GENIE 3.0 Basic-Script Writer

The GENIE editing program can run strategies in a debug mode, in which one can work through a program line by line and review a scan task block by block. This major improvement allows users to design and debug complicated strategies in the editor before using the runtime only program. The runtime program now uses even less memory and offers improved performance.

The Script Designer is used for editing the Main Script and scripts inside a task, including Pre-Task, Post-Task, and Basic-Script. The main script controls the entire run-time, including starting a task and/or stopping a task. GENIE provides a variety of commands to process I/O data.

The main script is used to control and manage tasks. The Load/Unload statement is used to load/unload the whole strategy. The initialize statement initializes all the data related to a task. The Set-Task-Time statement is used to change the start time, duration or period of the scan task. The Run statement is used to start and run a task to completion. Scan statement will do a one-time scan of the task specified. The pre-task script is used to define task properties and initialize all data related to a task. The post-task script is used to clear-out task related data. Basic-Script is used to get I/O data and to set I/O data.

3.7.1.3 GENIE Task Designer

It uses a dataflow-programming model that makes one free from the linear architecture of text-based languages. To create a process monitoring and control application, one can construct the block diagram without worrying about the many syntactical details of conventional programming. Objects (icons) from the Functions menu can be selected and can be connected with wires to pass data from one block to the next.

GENIE 3.0 Task Designer allows editing of multiple tasks at the same time. Each task is contained in a task window and has it's own properties such as scan rate, start/stop method, etc. One strategy file is used to store all scan tasks that are related to a control strategy. For simple strategy with only one task, it runs the same way as before. But for strategies that have more than one scan task, a top-level main script is required to manage the execution of all scan tasks. A large complicated task can be broken into several smaller, simpler tasks. This not only simplifies the editing job, but also increases the performance at run-time, as fewer blocks need to be processed at each scan.

GENIE Task Designer features the block sequence arrangement functions that show the order of execution on all blocks. With displayed order number, users can arrange the order of execution of the blocks (icons) based on the priority of operations to meet the requirements of system needs.

The Virtual Tag is a powerful feature that provides the ability to let developer to create customized tag in Task Designer without using User Defined DLL. The virtual tag is created by Task Designer and stored in data center as other built-in blocks. The virtual tags are globally available to all tasks, one can use virtual tags to share data among multiple tasks.

3.7.1.4 GENIE Display Designer

If one has man machine interface (MMI) requirements, GENIE 3.0 screen designer will help to quickly create intuitive standard graphical displays by providing graphical wizards. And, one can further customize MMI with drawing tools and user-defined display tool.

GENIE enhances the man-machine interface by providing graphic tools to draw pumps, valves, rectangles, circles, segments, and polygons in the screen designer's toolbox. In addition, it allows the user to configure the colors and sizes of these figures. These drawing tools include oval, rectangle, round rectangle, polygon and line. In addition to drawing tools, GENIE also provides *Make Object* and *Break Object* commands to integrate drawing components into a meaningful picture for data acquisition and control.

3.7.1.5 GENIE Report Designer

GENIE Report Designer provides a configurable environment in which users can define the contents of a given report. It collects the required data at specific time intervals and these reports are printed automatically at a user-defined time. The interfaces provided in Report Designer may also be used to select and print reports manually.

Data collection function creates database files for each defined TAG point at a user-specified time. Data collection function is designed for report generation. The shortest time interval for data collection from a given TAG point is 10 minutes. High-speed data collection is accomplished through other trend data collection functions.

Report format configuration function provides user interface dialog boxes, which allow users to set up the report format and report print time. Report format entries are organized in a table form and users enter text or specified keywords to define each table column. Information from each report format is saved to a format file, and extracted during report generation.

Report generation function combines the format file and data collection database file to produce a printed report. At the moment, reports are limited to tabular format report. Report scheduler sets the time at which reports will be generated. At a user-defined time, report scheduler calls the report generation function to generate the report. Report Scheduler also informs users of report printing status.

Alarm report generation function produces equipment fault reports. These reports provide information on time of fault occurrence, operator acknowledgement and recovering records.

3.7.2 Data Pre-processing

Data analysis should include calibration and modeling prior to the Pattern Recognition (PARC). Many of these procedures are based on multivariate numerical data processing and before the methods can be successfully applied, it is usual to perform pre-processing of the data [33] (Adams, 1995). The main aims of this stage are: (1) to reduce the amount of data which are irrelevant to the study; (2) to enhance sufficient information within the data to achieve the desired goal; (3) to extract the information in, or transform the data to, a form suitable for further analysis, (4) reduction in complexity of the sensor coating selection. Pre-processing simplifies the main task of characterizing complex mixtures without the need to identify and quantify individual components and that it can be exploited for structure activity relationship studies.

The data generated by each sensor are pre-processed by analytical methods and the results are then analysed. Probably the most common method of preprocessing spectral data is normalization. This may involve either scaling each spectrum in a collection so that the most intense band in each spectrum is a constant value or the spectra may be normalized to constant area under the absorption or emission curve. The techniques used for preprocessing include baseline manipulation and suitably weighting of E-nose sensors. Baseline manipulation transforms the time-dependant sensor response to its baseline and removes the time dependence. The three techniques are commonly used for baseline manipulation:

Deference:

$$X_{ij} = |Y_{ij}^{\text{odour}} - Y_{ij}^{\text{reference}}| \tag{3.4}$$

Relative:

$$X_{ij} = |(Y_{ij}{}^{odour}) \ / \ (Y_{ij}{}^{reference})| \qquad (3.5)$$

Fractional:

$$X_{ij} = |(Y_{ij}{}^{odour} - Y_{ij}{}^{reference})| \ / \ (Y_{ij}{}^{reference}) \qquad (3.6)$$

Where X's are the manipulated or calculated (preprocessed) values and Y's are the observed or measured values for sample odour and reference room air. Most widely used normalization technique is vector normalization where each feature vector is divided by its mean so that it is transformed to lie on a hyper-sphere of unit value. Normalization removes sample-to-sample absolute variability and transforms vector length to be one. Normalization assumes that the extracted features linearly correlate with signal intensity. The following normalization techniques are normally used.

Liberalization:

$$X_{ij} = \log |(Y_{ij}{}^{max} - Y_{ij}{}^{min})| \qquad (3.7)$$

$$X_{ij} = \sqrt{(|(Y_{ij}{}^{max} - Y_{ij}{}^{min})|)} \qquad (3.8)$$

Vector Normalization:

$$R_{ij} = X_{ij} \ / \ \sum (X_{ij})^2 \qquad (3.9)$$

Sensor Normalization:

$$R_{ij} = (X_{ij} - X_{ij}{}^{min}) \ / \ (X_i^{max} - X_i{}^{min}) \qquad (3.10)$$

Sensor Auto Scaling:

$$R_{ij} = |(X_{ij} - X_{ij}{}^{mean})| \sigma_i \qquad (3.11)$$

Where σ is Standard deviation.

More complex approaches involve developing a covariance matrix between variables and extracting the eigenvectors and eigenvalues. Eigen analysis provides a set of variables, which are linear combinations of the original variables. This has the effect of reducing the dimensionality of the data and making the analysis simpler [33]. An important consideration is the statistical design of experiments (DOE) to ensure that data obtained are valid. This is largely because of the use of too few samples with very different characteristics. One-at-a-time experiments, i.e. those where only one variable is changed at a time, are inefficient. Thus, the statistical DOE is of paramount importance, especially when many variables are involved. DOE techniques allow the experimenter to identify the input variables, which affect the output of any process [34]. DOE can speed up process and product development as well as improve existing processes and products. Furthermore, DOE can also be used to reduce the amount of effort needed in an experiment by eliminating redundant observations. Thus, the main advantages of DOE are: (1) fewer experiments are needed because several factors are varied simultaneously; (2) the judicious choice of factor settings reduces the number of experimental runs; (3) more information per experiment is generated.

3.7.3 Feature Selection

In the past few years, the number of papers related to feature extraction, including feature construction, space dimensionality reduction, sparse representations, and feature selection have appeared in the literature. The applications studied cover a wide range of domains, including bio-informatics, chemistry, text processing, pattern recognition, speech processing, vision perception, flavour discrimination and quality prediction. The feature selection includes experimental design, algorithms, and theoretical analysis. Resultant data sets from experiments for flavour discrimination of tea and spice are used for feature selection. The different data sets or different data splits have been proved to be useful. This makes them easy to compare. A number of datasets were formatted for the purpose of benchmarking feature selection algorithms described in section, 7.3.2. The data sets were chosen to span a wide variety of domains. We chose data sets that had sufficiently large experimental measurements to create a large enough test set to obtain statistically significant results. The input variables are continuous or binary, sparse or dense. The similarity of the tasks allows entering results on all data sets to test the genericity of the algorithms. The feature selection process is to find suitable feature selection algorithms that significantly outperform methods using all features at a time or using as a benchmark. All data responses from E-nose on five flavours of overlapping tea, ten flavours of non-overlapping tea and five flavours of spice were formatted for this purpose. To facilitate entering results for all datasets, the tasks of classification are of two-class problems. Each dataset is split into training, validation, and test set. Only the training labels are provided.

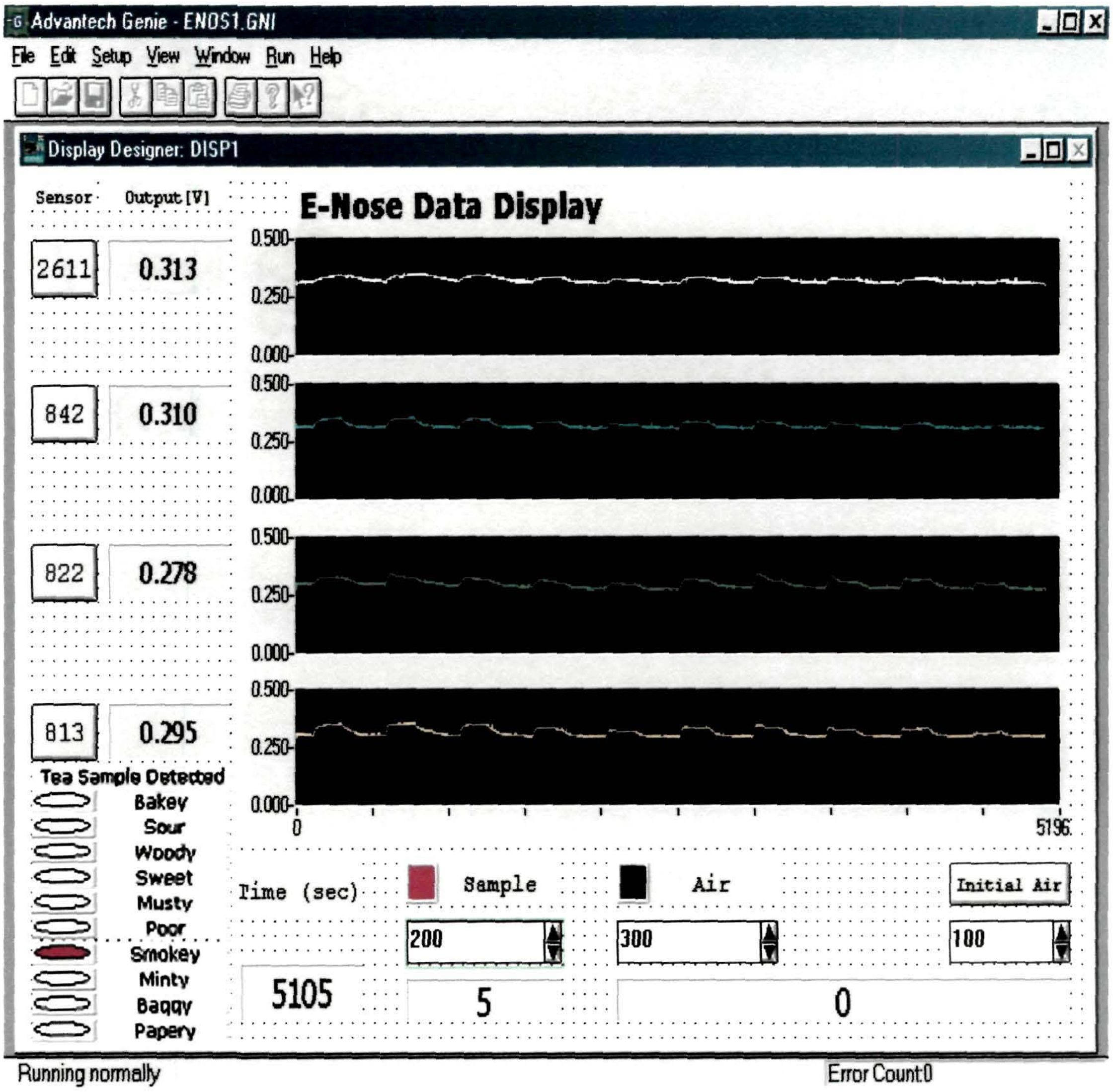
After checking some simple feature statistics, a subset is selected and trained it using Neural Network tools and program modules (Appendix 7). It can further select a feature subset, which is then trained by MLP, LVQ, RBF and PNN with full parameter selection. Drift compensations are also considered at this stage. Results vary on data sets. For example, one or two data sets, after simple feature statistics, have not matched but most of the results for tea and spice flavours have performed 90 % and above correct classifications.

A combined data clustering approach was implemented for tea data by combining the 3D-Scatter plot, Fuzzy Cluster Mean (FCM) and Self Organizing Map (SOM) network. In multisensor space, normalized data-sets were represented using 3D-Scatter plots. From the FCM approach, a cluster center is found for each group by minimizing a dissimilarity function. These cluster centers were plotted in multisensor space. So combining the 3D-Scatters plot and FCM, cluster centers were properly located in multisensor space and also within the data. Thereafter a [5x1] SOM network was trained, with the whole data sets. After 500 epochs it was clear that the five nodes had approached to the five cluster centers (estimated using FCM). So, using these three data clustering algorithms simultaneously, better *classification* of data into different clusters was achieved.

3.7.4 Graphical User Interface

The graphical user interface (GUI) is designed to make calculations and analytical processing of data simple and user friendly. A GENIE based GUI is shown in the figures 3.17 and LabVIEW (National Instruments Inc.) based GUI is shown in figure

3.18. Let us take a simple example to understand the basic idea of the GUI. Figures 3.17 and 3.18 are presented with a GUI data manager and display windows, which were designed for the experimental work. This window has its own work area, separate from the more familiar command line workspace. Thus, when using the GUI, the GUI results may be *exported* to the other analysis workspace and windows. Similarly the results can be imported from the command line workspace to the GUI.



Sensor	Output [V]
2611	0.313
842	0.310
822	0.278
813	0.295

Time (sec)	Sample	Air	Initial Air
	200	300	100
5105	5	0	

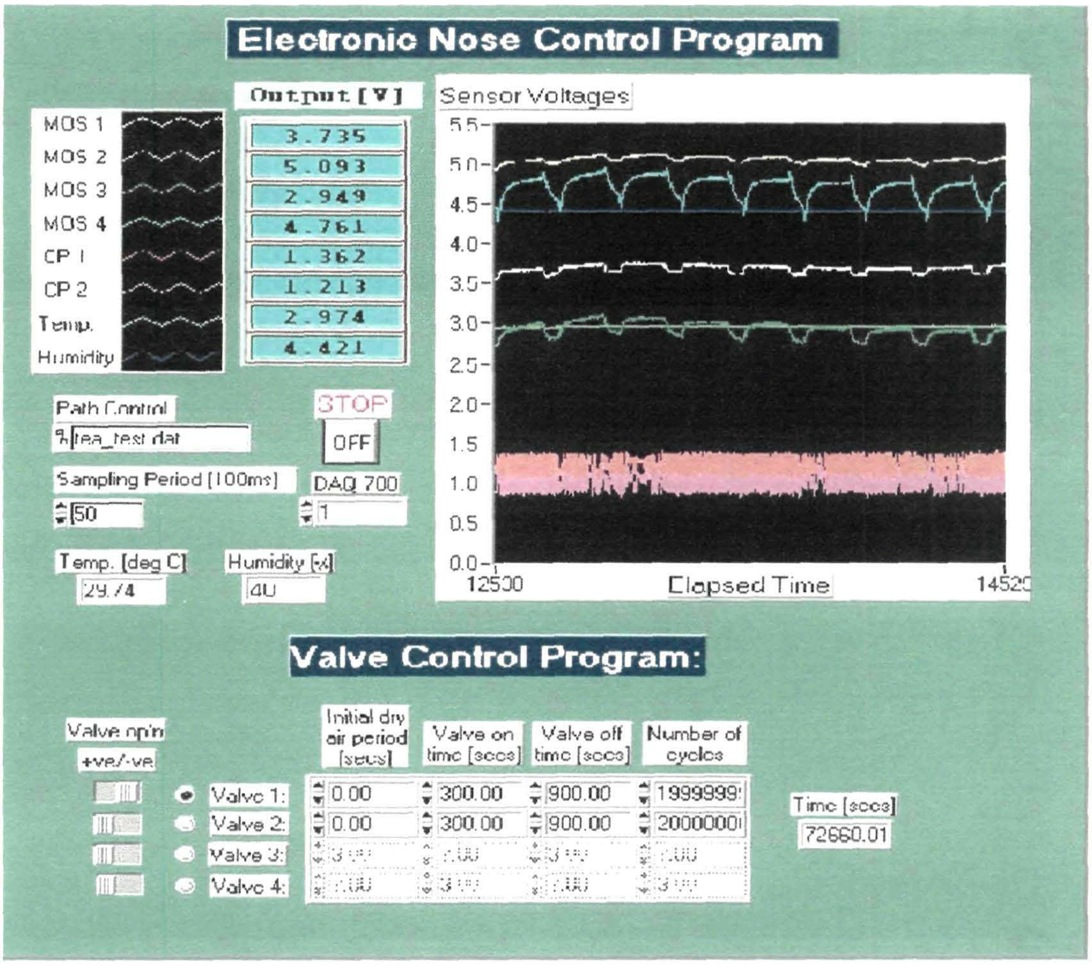
Running normally

Error Count:0

Figure 3.17: Typical Graphical Interface Designed in GENIE Environment

This interface is user friendly and flexible to configure many parameters, such as time settings, pump selection, online odour classifications display, graphical display of the response of all four sensors and experimental settings according to ambient conditions of temperature and humidity.

Once the interface window is ready and running, a network can be created, viewed, trained, simulated or exported with the final results to the required target point. The data can also be imported from the other applications for use in the GUI. Results and responses can be redisplayed as in graphical display window. As it is very clear from the figures 3.17 and 3.18 that the E-nose process can always be monitored by many useful visual information available from the GUI designed display window.



	Output [V]
MOS 1	3.735
MOS 2	5.093
MOS 3	2.949
MOS 4	4.761
CP 1	1.362
CP 2	1.213
Temp.	2.974
Humidity	4.421

Path Control: tea_test.dat

Sampling Period [100ms]: 50

DAQ 700: 1

Temp. [deg C]: 29.74

Humidity [%]: 40

	Initial dry air period [secs]	Valve on time [secs]	Valve off time [secs]	Number of cycles
● Valve 1:	0.00	300.00	900.00	1999999
○ Valve 2:	0.00	300.00	900.00	2000000
○ Valve 3:	3.00	0.00	3.00	0.00
○ Valve 4:	0.00	3.00	0.00	3.00

Time [secs]: 72660.01

Figure 3.18: Typical Graphical Interface Designed in LabView Environment.

3.8 Conclusion

The concept of E-nose was evolved as far as 40 years back, however, in the form of gas sensors. The present E-nose has made remarkable progress in last decade and reached to the level of smart and intelligent instruments such as robotics. It has found huge applications in diversified fields including industrial process, food quality, medical diagnosis and space applications in space shuttles and remote sensing. E-nose future seems very bright for diagnosis of most fatal diseases at low cost and in shortest possible time to save the lives.

E-nose technology uses wide variety of sensors such as Conducting Polymers, MOS based Sensors and Piezoelectric Sensors depending upon suitability and applicability. Figaro gas sensors are used for the research work in this thesis. These sensors make perfect choice for VOC's and other vapours.

E-nose has all characteristic features, functional capacity and applications areas to be classified as intelligent instrumentation system.

A prototype E-nose is designed and developed using TGS 813, TGS 822, TGS 842 and TGS 2611 gas sensors and other control Electronics, in the Department Of Electronic, School of Science and Technology, Tezpur University (A Central University), India. The system is successfully applied for classification and discrimination of tea and spice flavours.

Data acquisition is performed in GENIE and LabVIEW environments. Classification and prediction of flavours is achieved by the application of Neural Network algorithms. GUI is programmed in GENIE and LabVIEW to manage data acquisition, preprocessing, classification and visual displays.

Detailed E-nose technology and its applications are given elaborate space in this chapter. This will make clear and complete conceptual understanding of the subject. A prototype system, which is developed in this research, is treated with design techniques, methods and procedures.

References

[1] Wilkens, W.F. and D. Hartman. 1964. *An electronic analog for the olfactory process*. Journal Food Sci. 29:372-378.

[2] Ritaban Dutta, K. R. Kashwan, M. Bhuyan, E. L. Hines, J. W. Gardner *Electronic Nose based tea quality standardization,* Neural Networks Volume 16, Issue 5-6 (June 2003) 2003 Special issue: Advances in neural networks research IJCNN'03, Elsevier Science Ltd, UK

[3] Ritaban Dutta, E. L. Hines, J. W. Gardner K. R. Kashwan, M. Bhuyan, *Determination of tea quality by using a Neural Network based Electronic Nose,* International Joint Conference on Neural Network (IJCNN 2003), Portland, Oregon, USA. Conference sponsored by IEEE Neural Network Society and International Neural network Society, July 20-24 Vol. 1, pp. 404-409, 2003.

[4] K. R. Kashwan, M.Bhuyan, *Determination of Drift due to Temperature, Humidity and Pressure in the Samples for MOS based Electronic-Nose Sensors,* BIOMED-2005, National Conference on Emerging Trends in Biomedical Instrumentation, Birla Institute of Technology, Mesra, Ranchi, Apr 29-30 2005.

[5] Cattrall, R.W. (1997). *Chemical Sensors.* Pp. 1-2. Oxford: Oxford University Press.

[6] Hall, E.A.H. (1990). *Biosensors.* Pp. 1-10. Milton Keynes: Open University Press.

[7] Persaud, K.C. & Travers, P.J. (1997). Arrays of broad specificity films for sensing volatile chemicals. In: *Handbook of Biosensors and Electronic Noses: Medicine, Food and Environment* (edited by E. Kress-Rogers). Pp. 563-589. Boca Raton: CRC Press.

[8] Nishizawa, M.; Shibuya, M.; Sawaguchi, T.; Matsue, T.; Uchida, I. *J. Phys. Chem.* 1991, *95*, 9042.

[9] Nishizawa, M. *J. Electroanal. Chem.* 1994, *371*, 273-75.

[10] Thompson, M. & Stone, D.C. (1997). *Surface-launched Acoustic Wave Sensors.* Pp. 1-10. New York: Wiley-Interscience.

[11] Graseby Ionics (2002). Available from: http://www.graseby.

[12] Ashcroft, A.E. (1997). *Ionization Methods in Organic Mass Spectrometry.* Pp. 44-46. Cambridge: Royal Society of Chemistry.

[13] Ellis, A. (2003). From: http://www.le.ac.uk/chemistry/research/chreame2.html.

[14] Pinheiro, C., Rodrigues, C.M., Schafer, T. & Crespo, J.G. (2002). Monitoring the aroma production during wine-must fermentation with an electronic nose. *Biotechnology and Bioengineering, 77, 632-640*

[15] Herrmann, H., Jonischkeit, T., Bargon, J. *et al.* (2002). Monitoring apple flavour by use of quartz microbalances. *Analytical and Bioanalytical Chemistry*, 372, 611-614.

[16] Grigioni, G.M., Margaria, C.A., Pensel, N.A., Sanchez, G. & Vaudagma, S.R. (2000). Warmed-over flavour analysis in low temperature-long time processed meat by an electronic nose. *Meat Science*, 56, 221-228.

[17] Spanier, A.M., Flores, M. & Toldra, F. (1997). Flavour differences due to processing in dry-cured and other ham products using conducting polymers (electronic nose). In: *Flavour and Chemistry of Ethnic Foods*. Pp. 169-183. New York: Kluwer/Plenum.

[18] Ragazzo, J.A., Chalier, P., Crouzet, J. & Ghommidh, C. (2001). Identification of alcoholic beverages by coupling gas chromatography and electronic nose. In: *Food Flavours and Chemistry*. Pp. 404-411. Cambridge: Royal Society of Chemistry

[19] Walte, A. & Munchmeyer, W. (2000). Novel electronic nose for the analysis of alcoholic beverages. In: *Frontiers of Flavour Science* (edited by P. Schieberle & E. K.-H. Garching). Pp. 144-147. Germany.

[20] K. Pope, "*Technology Improves on the Nose As Science Tries to Imitate Smell*," Wall Street Journal, pp. B1-2, 1 March 1995.

[21] P.E. Keller, R.T. Kouzes, L.J. Kangas, and S. Hashem, "*Transmission of Olfactory Information for Telemedicine*," Interactive Technology and the New Paradigm for Healthcare, R.M. Satava, K. Morgan, H.B. Sieburg, R. Mattheus, and J.P. Christensen *(ed.s), IOS Press, Amsterdam, 1995, pp. 168-172.*

[22] R.J. Lauf and B.S. Hoffheins, "*Analysis of Liquid Fuels Using a Gas Sensor Array*," Fuel , vol. 70, pp. 935-940, 1991.

[23] H.V. Shurmur, "*The fifth sense: on the scent of the electronic nose*," IEE Review, pp. 95-58, March 1990.

[24] P.E. Keller, R.T. Kouzes, and L.J. Kangas, "*Three Neural Network Based Sensor Systems for Environmental Monitoring*," IEEE Electro 94 Conference Proceedings, Boston, MA, 1994, pp. 377-382.

[25] Heino, R.L. & Ahvenainen, R. (2002). *Monitoring of taints related to printed solid boards with an electronic nose*. Food Additives and Contaminants, **19**, 209-220.

[26] M. A. Rayan, M. L. Homer, M.G. Buehler, K. S. Manatt, B. Lau, D. Karmon and S. Jackson, *Monitoring Space Shuttle Air for Selected Contaminants Using an Electronic Nose*, International Conference on Environmental System, Danvers MA, July 12 – 16 1988

[27] Ritában Dutta, E. L. Hines, J. W. Gardner K. R. Kashwan, M. Bhuyan, *Tea quality prediction using a tin oxide based electronic nose*: Elsevier Science Journal:

Sensors and Actuators B 94(2003) Volume 94, Issue 2 (1 September 2003) Pages: 228-237. Elsevier Science Ltd. Oxford, UK.

[28] K. R. Kashwan, M.Bhuyan, *Tea Flavour Discrimination by using Electronic-Nose Sensors and Artificial Neural Network Pattern Recognition Techniques*, BIOMED-2005, National Conference on Emerging Trends in Biomedical Instrumentation, Birla Institute of Technology, Mesra, Ranchi, Apr 29-30, 2005

[29] Prasanna Chandrasekhar, *Conducting Polymers, fundamentals and Applications: A Practical Approach,* Kluwer Academic Publishers © 1999.

[30] Advantech Industrial Automation with PC'S *User's Guide GENIE: Application builder for data acquisition and control version 3.0* Advantech Genie 1st Edition.

[31] PCL-708 High Performance Data acquisition card. *User's Manual* Dynalog (India) Ltd. MATLAB Hand Book for Reference, www.mathworks.com

[32] National Instruments Corporation (1998). LabVIEW version 5 Data Manual. Austin, TX: National Instruments.

[33] Adams, M.J. (1995). Chemometrics in Analytical Spectroscopy. Pp. 70–79. Cambridge: Royal Society of Chemistry.

[34] Connecticut Quality Council (2003). Design of experiments (DOE): The Statistical Tool of Improvement. Available from: http://ctqualitycouncil.org/classes/doe.htm (accessed 16 April 2003)

§§

CHAPTER 4

TEA AND SPICE FLAVOUR DISCRIMINATION

Chapter 4

Tea and Spice Flavour Discrimination

Tea and Spice flavour discrimination using E-nose system is the main focus of this research. The experimental procedures, sample collection methods, data acquisition and analysis, discussions and results are the main topics described in this chapter. The experiments were performed for three different classes of samples, which covered key applications of E-nose in the areas of flavour discrimination. These three experiments for different samples are (i) flavour discrimination of overlapping samples of tea (ii) flavour discrimination of non-overlapping samples of tea and (iii) spice aroma classification. The first experiment reports on the results obtained from E-nose system on the tests carried out with five overlapping samples of the tea quality. The overlapping in the flavour terms of tea quality is due to faulty techniques adopted during tea manufacturing process. The second experiment is performed on the non-overlapping tea quality samples. The non-overlapping characteristics are not due to manufacturing defects but inherent flavours found in the tea quality. These flavour terms represent agronomical different qualities of tea due to either different tea species or due to deliberately infused flavours, such as mint, smoked flavour etc. The third experiment focuses on the classification techniques for five common spices, which are used for marinating and infusing aroma in the foodstuff.

Each sub-section focuses on the individual results together with the investigative ANN paradigms used. PCA technique is used as a basic data clustering and dimensional reduction method, FCM as a non-linear fuzzy classifier and ANNs, namely MLP, LVQ, RBF and PNN, as non-parametric PARC. A program, implemented using MATLAB® code presented in Appendix-7, is used to process the data-vectors and perform pattern analysis. A pre-processed data file of the correct format is created for each data set containing a data matrix, each row is a vector representing a sample for a given class and each column is a dimension representing a feature (sensor). For each sample a target class, represented by an integer value, is loaded at the same time for supervised classifiers.

4.1 Analysis of Overlapping Tea Samples

Basically, in the first E-nose experiment on the tea samples, faulty manufactured samples are used for flavour discrimination. This class of five tea samples had different flavour notes due to deviation from ideal manufacturing process or defective manufacturing and processing of tea. The deviations are non-compliance of temperature, humidity and timing requirements at the time of tea processing. This class of tea samples shows overlapping flavours [1] in terms of the classification results since the clustering techniques have produced partially overlapping graphical cluster displays [2].

4.1.1 Sample

The five tea samples, manufactured deliberately under non-compliance of standard manufacturing and processing conditions, are gathered from Tea Research Center Tocklai, Jorhat, Assam, India, for E-nose experiments. These tea samples are as follows:

(i) Drier mouth

(ii) Drier mouth again over-fired

(iii) Well fermented normal fired in oven

(iv) Well fermented over-fired in oven

(v) Under fermented normal fired in oven

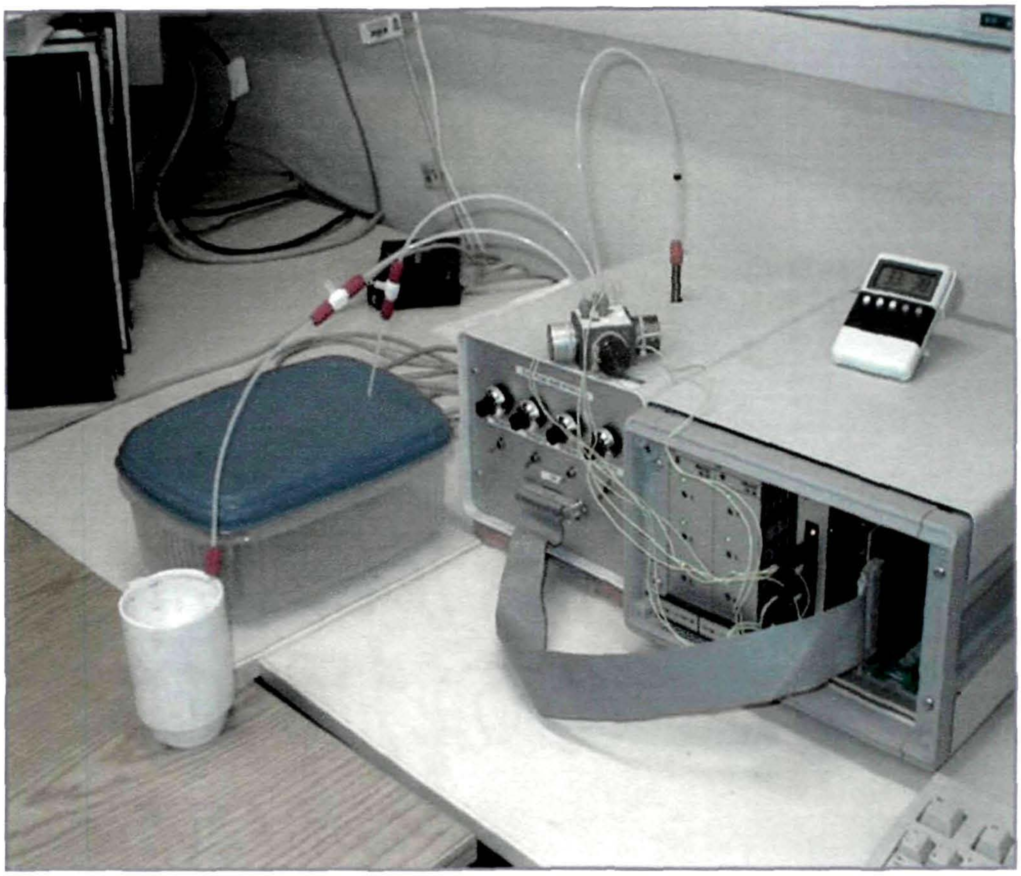


Figure 4.1: E-Nose System for testinq five overlapping tea samples.

The deliberately defective (non-compliance of temperature, humidity and timing requirements) manufacturing of tea samples was done by the Tea Research Center, Tocklai, on special request for E-nose experiments. The deliberate non-compliance of standard conditions for manufacturing tea was chosen based on the fact that these faults are practically quite common to happen inadvertently or due to limitations in the manufacturing and processing of the tea. Normally, one or other fault, to a less or more degree, can happen in the manufacturing. The five tea samples, therefore, truly represent broad categories of the tea flavour qualities in the market. The tea samples were hermetically packed just after manufacturing and then brought to the laboratory for experimentation. Each sample, without any additional alteration, was prepared for E-nose experiments.

4.1.2 Experimental Procedure

Photographic view of the tea testing setup is shown in figure 4.1. The five test samples of overlapping tea quality flavour terms were prepared and tested one at a time for experimentation. Conventionally, the tea beverage is prepared by adding tea grains to the boiling water and subsequently filtering out infused tea grains from liquor. Milk (optional) and sugar are added to the tea liquor for better taste. For the E-nose experiments, 10 gm of dry tea sample was added to 200 ml of boiling water to prepare liquor tea sample. The prepared tea sample was placed in the sample vessel. Other samples were similarly prepared for the experiments. The boiling water was added frequently to the tea samples to increase evaporation of VOCs.

The two plastic vessels (sample and reference) are connected through plastic tubes to the inputs to the sensor chamber. A diaphragm pump was used to sample the headspace of the vessels (Figure 3.9, chapter-3). The headspace of the sample vessel containing the tea samples and the reference vessel containing fresh room air were sampled in a sequence for the predefined time duration.

A sample measurement was performed for 5 minutes to complete first cycle of sample reading followed by 20 minutes of refreshing time (25 minutes for one data gathering cycle). The flow rate was kept at 2 l/minute. The sampling duration was chosen to optimize the stability of the sensor response to the flavour evaporated by the tea sample. The tea flavour vapour removed from the sample vessel by the diaphragm pump was replaced by fresh air from the room (Figure 3.9, chapter-3)

The reference vessel is half filled with pure water. Air from the room was pumped into the sensor chamber through reference vessel. The sample is prepared in boiling water and thus, there is a good amount of humidity present in tea vapour. The reason for keeping water in reference vessel to maintain the same humidity levels while sampling tea sample and fresh room air. It was expected that water vapour constituted same level in both the sampling instances. The sensors were allowed to return to their baseline level over a period of about 20 minutes after sampling the headspace of the tea sample vessel. The sensors' response to the pure water in the reference vessel was used as a baseline response for the experiments. This ensured that the E-nose system was responding to the tea VOCs rather than to water vapour. The room air from laboratory was used as a carrier gas to keep the experiment simple. Any variations of the 'base line' (which may be due to different volatiles present in room air) were monitored. The results from different data processing algorithms suggest that there were no significant variations in the base line during the time period of the experiments. One data gathering cycle comprised of taking alternately, a headspace sample reading from the tea sample vessel (5 minutes) and then from the reference vessel (20 minutes). During the process of the measurements, a sample of each sensor's response was taken every second and stored in a data file for subsequent processing (Figure. 3.18, chapter-3).

4.1.3 Data Acquisition

Data acquisition in case of an E-nose system refers to the process of recording E-nose sensor response in a prescribed format at predefined rate. In this research good amount of data sets were acquired in the long series of experiments. The data set acquisition during experiments were graphically displayed in LabVIEW© (National Instruments Inc.) and GENIE (ADVANTECH®, American Advantech Corporation) environments to facilitate visual and statistical analysis.

The data for analysis were extracted from the stable part of the sensors' responses curves under constant temperature and humidity conditions. This was done to rule out any fluctuation in sensors' response at the time of addition of hot water.

The five different tea sample response data sets were gathered for 62.5 hours for each sample. Thus a total time of 312.5 hours required for data gathering. For each sample of tea quality, 150 data vectors each of dimensions of 4 × 150 for 4 sensors were gathered from 150 data gathering cycles over a continuous period of consecutive days. The number of data vectors and samples were limited by practical circumstances although data collection continued for days using online E-nose data logger system.

4.1.4 Data Preprocessing

For one data gathering cycle, five data matrices (each of dimensions of 4 × 150 for 4 sensors) for five-tea samples were concatenated (merged) into a single matrix of dimensions of 4 × 750. This resulted in a single data set made of the five data sets from five tea samples. A total of 150 data sets (each of dimensions 4 × 750) were formed from 150 data gathering cycles. The response of the E-nose sensors is displayed in figure 3.18 in chapter 3, using LabVIEW environment.

In this part of research experiments, the model used for the data processing is static change in sensor resistance, given by,

$$dR_i = R_{\text{air, i}} - R_{\text{odour, i}} \tag{4.1}$$

Where dRi = Change in the resistance of the sensor when exposed to the tea vapour

$R_{\text{air, i}}$ = The resistance of the sensor when exposed to the room air

$R_{\text{odour , i}}$ = The resistance of the sensor when exposed to the tea vapour

i = 1,2...4, Number of sensors used in experiments

All tea data sets were then normalized by dividing each d*R* by the maximum value of resistance R of the sensor to set the range of resistance between *0* and *1*. This normalization was applied to the data, prior to principal component analysis (PCA), self-organizing map (SOM), fuzzy C means (FCM) algorithm analysis and the analysis using the MLP, LVQ, PNN and RBF networks simulated on the MATLAB 6.1 environment. For example, a sample of the normalization of sensor response data is shown tea in Table 4.1 for a baggy tea sample. The first 4 columns show the actual response (in volts) of 4 sensors used in E-nose experiments. The data set (4 × 100) is extracted from most stable part of the E-nose response. Table 4.2 shows the statistical parameters of each sensor's response. Data vector for each sensor is divided by maximum response value for corresponding sensor. The new vector is called normalized vector for the particular sensor. Columns 5 through 8 in Table 4.1 show the normalized vectors for corresponding 4 sensors' response given in columns 1 through 4. The response (output) is recorded in volts, however it can be converted into resistance, as both are correlated quantities. The actual response in the example of Table 4.1 is in the range of *0* to *1* volt, however, the response may be in any range from *0* and *5* volt. The range of response after normalization is always from *0* to *1*. If the normalization is done by using equation 4.1, the normalized sensors' response will be in resistance units.

Table 4.1: Sensors' response before and after normalizations for baggy tea sample (contd.)

Sensors' actual response (volts)				Sensors' normalized response				Sensors' actual response (volts)				Sensors' normalized response			
Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4
0 3080	0 3760	0 3220	0 3150	1 0000	0 9330	1 0000	0 8607	0 3050	0 3980	0 2470	0 3640	0 9903	0 9876	0 7671	0 9945
0 2980	0 3660	0 3130	0 3030	0 9675	0 9082	0 9720	0 8279	0 3050	0 3960	0 2470	0 3660	0 9903	0 9826	0 7671	1 0000
0 3050	0 3780	0 3220	0 3170	0 9903	0 9380	1 0000	0 8661	0 3050	0 3910	0 2470	0 3660	0 9903	0 9702	0 7671	1 0000
0 3080	0 3810	0 3200	0 3250	1 0000	0 9454	0 9938	0 8880	0 3050	0 4000	0 2470	0 3640	0 9903	0 9926	0 7671	0 9945
0 3080	0 3830	0 3170	0 3320	1 0000	0 9504	0 9845	0 9071	0 3030	0 4030	0 2440	0 3640	0 9838	1 0000	0 7578	0 9945
0 3080	0 3880	0 3130	0 3320	1 0000	0 9628	0 9720	0 9071	0 3050	0 4000	0 2470	0 3640	0 9903	0 9926	0 7671	0 9945
0 3080	0 3880	0 3080	0 3340	1 0000	0 9628	0 9565	0 9126	0 3050	0 4000	0 2470	0 3640	0 9903	0 9926	0 7671	0 9945
0 3080	0 3910	0 3000	0 3420	1 0000	0 9702	0 9317	0 9344	0 3050	0 3980	0 2440	0 3640	0 9903	0 9876	0 7578	0 9945
0 3050	0 3910	0 3000	0 3420	0 9903	0 9702	0 9317	0 9344	0 3030	0 4000	0 2440	0 3640	0 9838	0 9926	0 7578	0 9945
0 3050	0 3910	0 2950	0 3440	0 9903	0 9702	0 9161	0 9399	0 3050	0 4000	0 2440	0 3640	0 9903	0 9926	0 7578	0 9945
0 3030	0 3960	0 2910	0 3470	0 9838	0 9826	0 9037	0 9481	0 3050	0 4000	0 2440	0 3660	0 9903	0 9926	0 7578	1 0000
0 3080	0 3960	0 2910	0 3490	1 0000	0 9826	0 9037	0 9536	0 3050	0 4000	0 2440	0 3640	0 9903	0 9926	0 7578	0 9945
0 3050	0 3960	0 2860	0 3520	0 9903	0 9826	0 8882	0 9617	0 3050	0 3980	0 2440	0 3640	0 9903	0 9876	0 7578	0 9945
0 3050	0 3980	0 2860	0 3490	0 9903	0 9876	0 8882	0 9536	0 3050	0 3980	0 2420	0 3640	0 9903	0 9876	0 7516	0 9945
0 3080	0 3760	0 3220	0 3151	1 0000	0 9330	1 0000	0 8608	0 3030	0 4000	0 2420	0 3640	0 9838	0 9926	0 7516	0 9945
0 2980	0 3660	0 3130	0 3031	0 9675	0 9082	0 9720	0 8280	0 3030	0 4000	0 2420	0 3640	0 9838	0 9926	0 7516	0 9945
0 3050	0 3780	0 3220	0 3171	0 9903	0 9380	1 0000	0 8662	0 3050	0 3980	0 2440	0 3640	0 9903	0 9876	0 7578	0 9945
0 3080	0 3980	0 2780	0 3540	1 0000	0 9876	0 8634	0 9672	0 3030	0 4000	0 2420	0 3660	0 9838	0 9926	0 7516	1 0000
0 3050	0 3980	0 2730	0 3560	0 9903	0 9876	0 8478	0 9727	0 3080	0 4000	0 2440	0 3640	1 0000	0 9926	0 7578	0 9945
0 3050	0 3980	0 2730	0 3560	0 9903	0 9876	0 8478	0 9727	0 3030	0 3980	0 2440	0 3640	0 9838	0 9876	0 7578	0 9945
0 3050	0 3980	0 2710	0 3590	0 9903	0 9876	0 8416	0 9809	0 3050	0 4030	0 2420	0 3660	0 9903	1 0000	0 7516	1 0000
0 3080	0 3980	0 2710	0 3590	1 0000	0 9876	0 8416	0 9809	0 3050	0 4000	0 2420	0 3640	0 9903	0 9926	0 7516	0 9945
0 3050	0 4000	0 2690	0 3590	0 9903	0 9926	0 8354	0 9809	0 3050	0 3980	0 2420	0 3640	0 9903	0 9876	0 7516	0 9945
0 3050	0 3980	0 2660	0 3590	0 9903	0 9876	0 8261	0 9809	0 3030	0 3980	0 2420	0 3640	0 9838	0 9876	0 7516	0 9945
0 3050	0 3960	0 2660	0 3590	0 9903	0 9826	0 8261	0 9809	0 3050	0 4000	0 2420	0 3640	0 9903	0 9926	0 7516	0 9945
0 3050	0 3960	0 2610	0 3590	0 9903	0 9826	0 8106	0 9809	0 3050	0 4000	0 2420	0 3660	0 9903	0 9926	0 7516	1 0000
0 3050	0 3980	0 2640	0 3590	0 9903	0 9876	0 8199	0 9809	0 3050	0 3980	0 2440	0 3640	0 9903	0 9876	0 7578	0 9945
0 3050	0 3980	0 2640	0 3610	0 9903	0 9876	0 8199	0 9863	0 3050	0 3980	0 2420	0 3640	0 9903	0 9876	0 7516	0 9945
0 3050	0 4030	0 2590	0 3610	0 9903	1 0000	0 8043	0 9863	0 3030	0 4000	0 2420	0 3640	0 9838	0 9926	0 7516	0 9945
0 3050	0 4000	0 2590	0 3610	0 9903	0 9926	0 8043	0 9863	0 3050	0 4000	0 2420	0 3640	0 9903	0 9926	0 7516	0 9945
0 3050	0 3980	0 2590	0 3610	0 9903	0 9876	0 8043	0 9863	0 3050	0 3980	0 2390	0 3640	0 9903	0 9876	0 7422	0 9945
0 3050	0 3980	0 2560	0 3640	0 9903	0 9876	0 7950	0 9945	0 3030	0 4000	0 2390	0 3640	0 9838	0 9926	0 7422	0 9945
0 3030	0 4000	0 2560	0 3640	0 9838	0 9926	0 7950	0 9945	0 3030	0 4000	0 2420	0 3640	0 9838	0 9926	0 7516	0 9945
0 3030	0 3980	0 2560	0 3610	0 9838	0 9876	0 7950	0 9863	0 3050	0 3980	0 2420	0 3640	0 9903	0 9876	0 7516	0 9945
0 3050	0 3980	0 2540	0 3640	0 9903	0 9876	0 7888	0 9945	0 3050	0 4000	0 2420	0 3640	0 9903	0 9926	0 7516	0 9945
0 3030	0 4000	0 2560	0 3640	0 9838	0 9926	0 7950	0 9945	0 3050	0 4000	0 2420	0 3640	0 9903	0 9926	0 7516	0 9945
0 3050	0 4000	0 2540	0 3640	0 9903	0 9926	0 7888	0 9945	0 3050	0 4000	0 2390	0 3640	0 9903	0 9926	0 7422	0 9945
0 3030	*0 3960*	*0 2540*	*0 3610*	*0 9838*	*0 9826*	*0 7888*	*0 9863*	*0 3050*	*0 4030*	0 2370	0 3640	0 9903	1 0000	0 7360	*0 9945*
0 3050	0 4000	0 2540	0 3640	0 9903	0 9926	0 7888	0 9945	0 3050	0 4000	0 2370	0 3610	0 9903	0 9926	0 7360	0 9863
0 3050	0 3980	0 2510	0 3640	0 9903	0 9876	0 7795	0 9945	0 3050	0 4000	0 2340	0 3640	0 9903	0 9926	0 7267	0 9945
0 3050	0 4000	0 2510	0 3660	0 9903	0 9926	0 7795	1 0000	0 3050	0 4000	0 2340	0 3610	0 9903	0 9926	0 7267	0 9863
0 3080	0 4000	0 2510	0 3640	1 0000	0 9926	0 7795	0 9945	0 3050	0 4000	0 2320	0 3640	0 9903	0 9926	0 7205	0 9945
0 3030	0 4000	0 2490	0 3640	0 9838	0 9926	0 7733	0 9945	0 3080	0 4000	0 2370	0 3640	1 0000	0 9926	0 7360	0 9945
0 3050	*0 4000*	*0 2510*	*0 3640*	*0 9903*	*0 9926*	*0 7795*	*0 9945*	*0 3050*	*0 3960*	*0 2340*	*0 3640*	*0 9903*	*0 9826*	*0 7267*	*0 9945*
0 3050	0 4000	0 2490	0 3640	0 9903	0 9926	0 7733	0 9945	0 3050	0 4000	0 2370	0 3640	0 9903	0 9926	0 7360	0 9945
0 3030	0 4030	0 2490	0 3660	0 9838	1 0000	0 7733	1 0000	0 3050	0 4000	0 2320	0 3660	0 9903	0 9926	0 7205	1 0000
) 3050	0 3980	0 2490	0 3660	0 9903	0 9876	0 7733	1 0000	0 3050	0 4000	0 2340	0 3640	0 9903	0 9926	0 7267	0 9945

0 3030	0 4000	0 2490	0 3660	0 9838	0 9926	0 7733	1 0000	0 3080	0 4000	0 2340	0 3660	1 0000	0 9926	0 7267	1 0000
0 3050	0 4000	0 2470	0 3640	0 9903	0 9926	0 7671	0 9945	0 3050	0 4000	0 2340	0 3640	0 9903	0 9926	0 7267	0 9945
0 3050	0 3980	0 2470	0 3640	0 9903	0 9876	0 7671	0 9945	0 3050	0 3980	0 2290	0 3640	0 9903	0 9876	0 7112	0 9945

Table 4.1: concluded

4.1.5 Data Cluster Analysis

Four different cluster classification methods were used for data clustering in case of five overlapping samples of tea flavour quality. These techniques are PCA, SOM, FCM and Combination of FCM, SOM and 3D Scatter diagrams. The 'cluster classification' methods were applied to explore the existence of expected tea clusters in feature space within the datasets. It was desirable to check the result to confirm that the categories or clusters identified by using each of these methods were not arbitrary. The data cluster analysis is explained in the following sub-sections for five overlapping tea flavour samples.

Table 4.2: Sensors' response statistics before and after normalization of data

Statistical property	Sensors' actual response (volts)				Sensors' normalized response			
	Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4
Max value	0.3080	0.4030	0.3220	0.3660	1.0000	1.0000	1.0000	1.0000
Min Value	0.2980	0.3660	0.2290	0.3030	0.9675	0.9082	0.7112	0.8279
Mean	0.3049	0.3974	0.2566	0.3590	0.9901	0.9862	0.7968	0.9808
Std. Dev.	0.0016	0.0056	0.0229	0.0116	0.0051	0.0139	0.0711	0.0317

4.1.5.1 Principal Component Analysis

The PCA is a linear method that has been shown to be effective for discriminating the response of the E-nose to complex tea quality flavours [4]. This method consists of expressing the sensor response vectors in terms of a linear combination of orthogonal vectors. Each orthogonal (principal) vector accounts for a certain amount of variance in the data, with a decreasing degree of importance.

The PCA was used to investigate how the response data vectors from the E-nose sensor array cluster in a multi-sensor space. The objective of this analysis was to establish the extent to which simple categories for the five different qualities of the tea samples exist. The results of PCA, using the normalized data vectors as described in section 4.1.4, are shown in figure. 4.2. The first three principal components were considered for analysis because they accounted for 99.8556% of the variance in the data set. The PC # 4 and higher were accounted only for 0.1444% of variance, thus neglected for analysis purpose. Five distinct tea flavour quality features appear to be evident. It is also clearly evident that the sensors are linearly correlated. The variance and load values for each of 3 principal components (figure 4.2) are shown in Table 4.3, which indicates that sensor 1, 3 and 4 were less correlated.

Table 4.3: The results of PCA analysis for overlapping tea flavour samples

PC	**% Variance**	**Load Values**			
		Sensor$_1$	**Sensor$_2$**	**Sensor$_3$**	**Sensor$_4$**
PC$_1$	97.9693	-0.4830	-0.5258	-0.4474	-0.5386
PC$_2$	1.0708	0.4535	0.3301	-0.8268	-0.0421
PC$_3$	0.8155	-0.6652	0.2167	-0.3110	0.6433

The correlation coefficients were calculated using the standard MATLAB function *'corrcoef'* [5], which has also indicated that sensors 1, 3 and 4 are less correlated. A reasonable correlation exists between five tea flavours, thus it can be concluded that the tea flavours established by PCA are consistent with the five different qualities of the tea samples [6, 7]. However, the inter-flavour boundaries are complex in shape and appear overlapping in figure 4.2, therefore, some patterns that belong to different tea flavours apparently appear to have very similar scores. This effect might have been due to experimental limitations, which has focused on practicality and simplicity of implementation.

The multivariate data analysis suggests that there is a considerable spread in the data. This spread might be due to drift in the sensors' responses. The drift problem has been dealt separately in chapter five. The occurrence of complex boundaries implies that a non-linear classification method is needed to obtain a good performance in terms of pattern recognition, rather than a linear PCA technique.

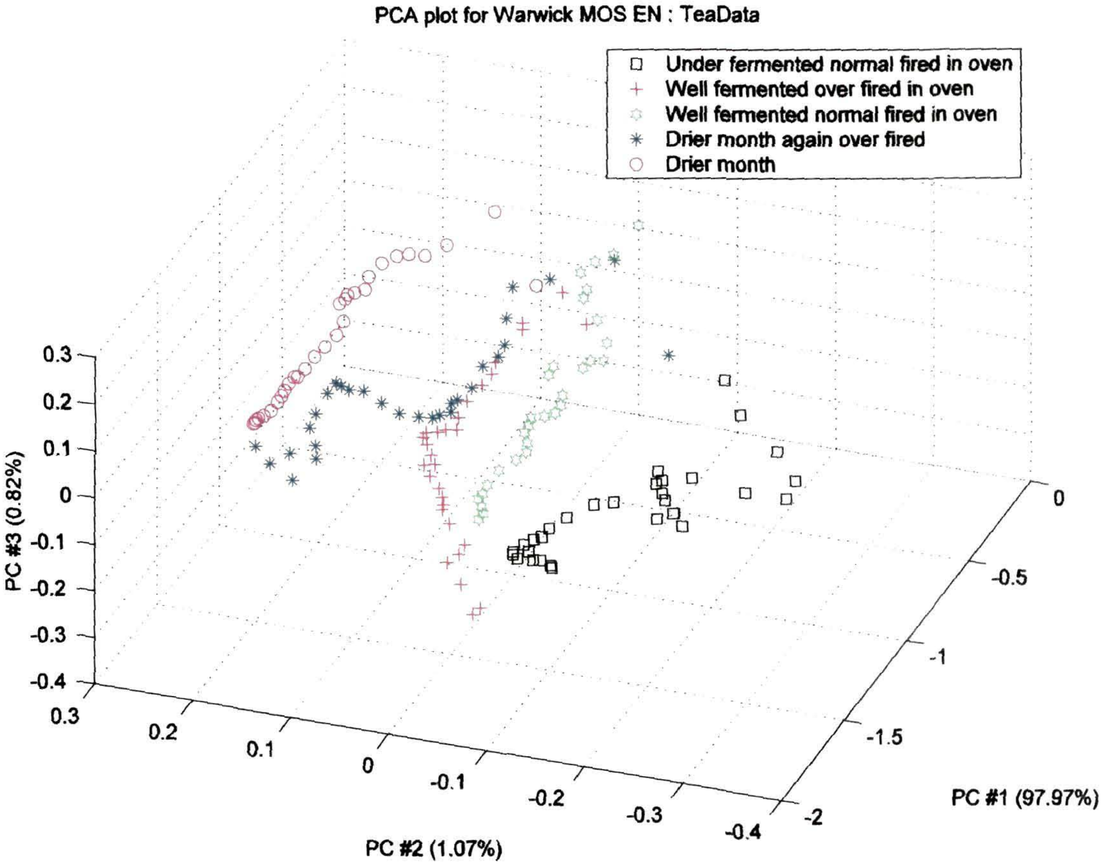


Figure 4.2: PCA plot for the overlapping tea sample data cluster analysis.

4.1.5.2 Fuzzy Cluster Mean Analysis

Fuzzy Cluster Mean (FCM) is a fuzzy data clustering and partitioning algorithm in which each data vector of particular tea flavour quality belongs to a cluster according to its degree of membership [8]. Fuzzy clustering essentially deals with the task of partitioning a set of patterns into a number of more-or-less homogeneous classes (clusters) with respect to a suitable similarity measure such that the patterns belonging to any one of the clusters are similar and the patterns of different clusters are as dissimilar as possible.

With FCM, an initial estimate of the number of clusters is needed so that the data sets are divided into C fuzzy groups (five clusters in this case). A cluster center is found for each tea flavour quality by minimizing a dissimilarity function [6, 7]. The similarity measure used has an important effect on the clustering results since it indicates that which mathematical properties of the data set should be used in order to identify the clusters. Fuzzy clustering provides partitioning results with additional information supplied by the cluster membership values indicating different degrees of belongingness. By applying FCM algorithm, the five cluster centers for five different tea flavour classes were obtained as shown in the figure 4.3.

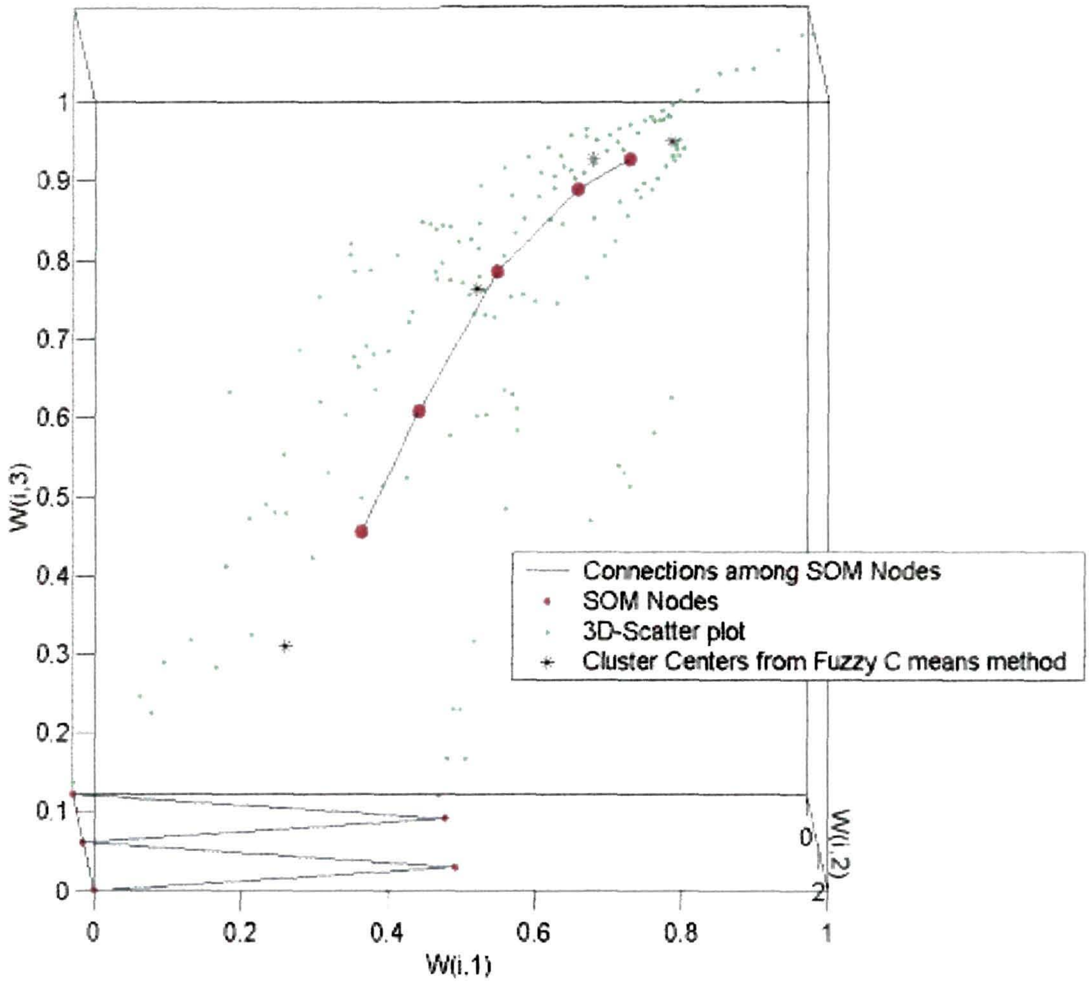


Figure 4.3: Combined 3D scatter plot for FCM and SOM for E-nose response data of five overlapping tea flavour quality samples.

4.1.5.3 Self Organizing Map Analysis

Self Organizing Map (SOM) was applied to the E-nose response data sets obtained from five tea flavour quality samples (overlapping nature) in order to investigate clustering using the response data from the four sensors. A SOM network is a non-linear ANN paradigm, which is able to accumulate statistical information about data with no other supplementary information than that provided by the sensors [9]. An SOM network was programmed and trained with the entire data set. Subsequently, samples are associated with one of the output neurons and neurons are grouped together to form categories of each identified class of tea sample. In the figure 4.4, there are five neurons that indicate the initial weights of the SOM before training.

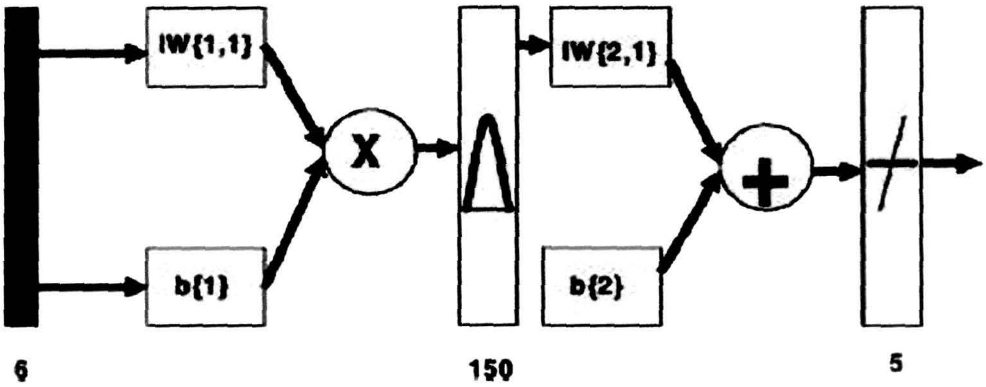


Figure 4.4: The architecture of the selected RBF neural network for SOM clustering.

4.1.5.4 Combined Analysis of SOM, FCM and 3D Scatter Diagram

An innovative data clustering approach was investigated for the E-nose response data of five tea samples by combining the 3D scatter plot, FCM and SOM network [10]. This is shown in figure. 4.3. For multi-sensor space, normalized data sets were represented to the 3D scatter plots. From the FCM approach, a cluster center is found for each group by minimizing a dissimilarity function [4, 9]. These cluster centers were plotted in multi-sensor space. By combining the 3D scatter plots and FCM, cluster centers were located in multi-sensor space and within the data set, and then a 5X1 SOM network was trained with the data sets.

As shown in figure 4.4, there are five neurons, which indicate the initial weights of the SOM before training. After 500 epochs the five nodes were approached to the five cluster centers estimated by FCM, as shown in the figure 4.3. By using these three data clustering algorithms simultaneously, a better representation of data clustering was achieved. E-nose data are very complex, noisy and non-linear. For this kind of complex data-set, FCM had located the most probable cluster centers within the data points whose centers were represented by a cluster for a particular tea sample flavour. Using 3D scatter plots (figure 4.5), the most probable positions of the probable clusters were estimated. SOM network had learned about the data set and on the basis of learning stages, SOM nodes took positions of the probable clusters' centers. These three un-supervised data processing algorithms, independently and effectively produced the same results about probable cluster positions within data sets.

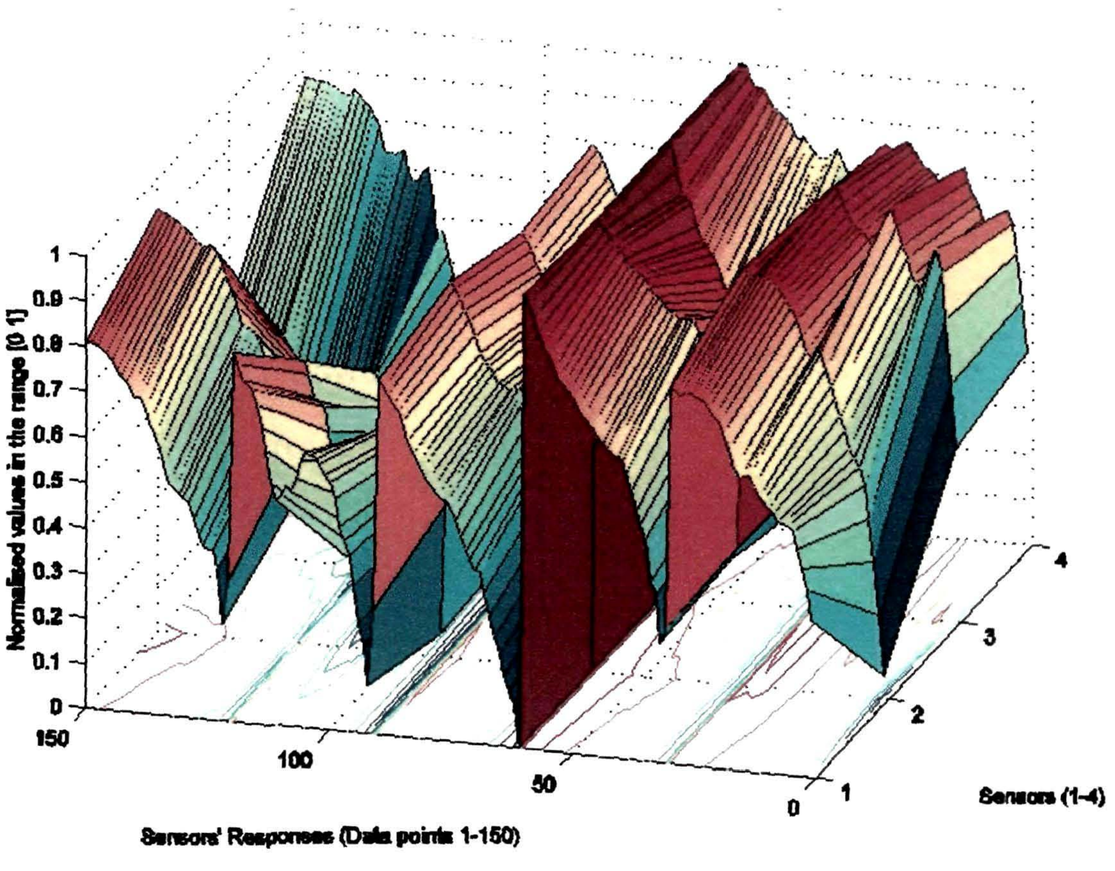


Figure 4.5: 3-D Scatter of normalised data of overlapping tea samples.

4.1.6 Artificial Neural Network Analysis

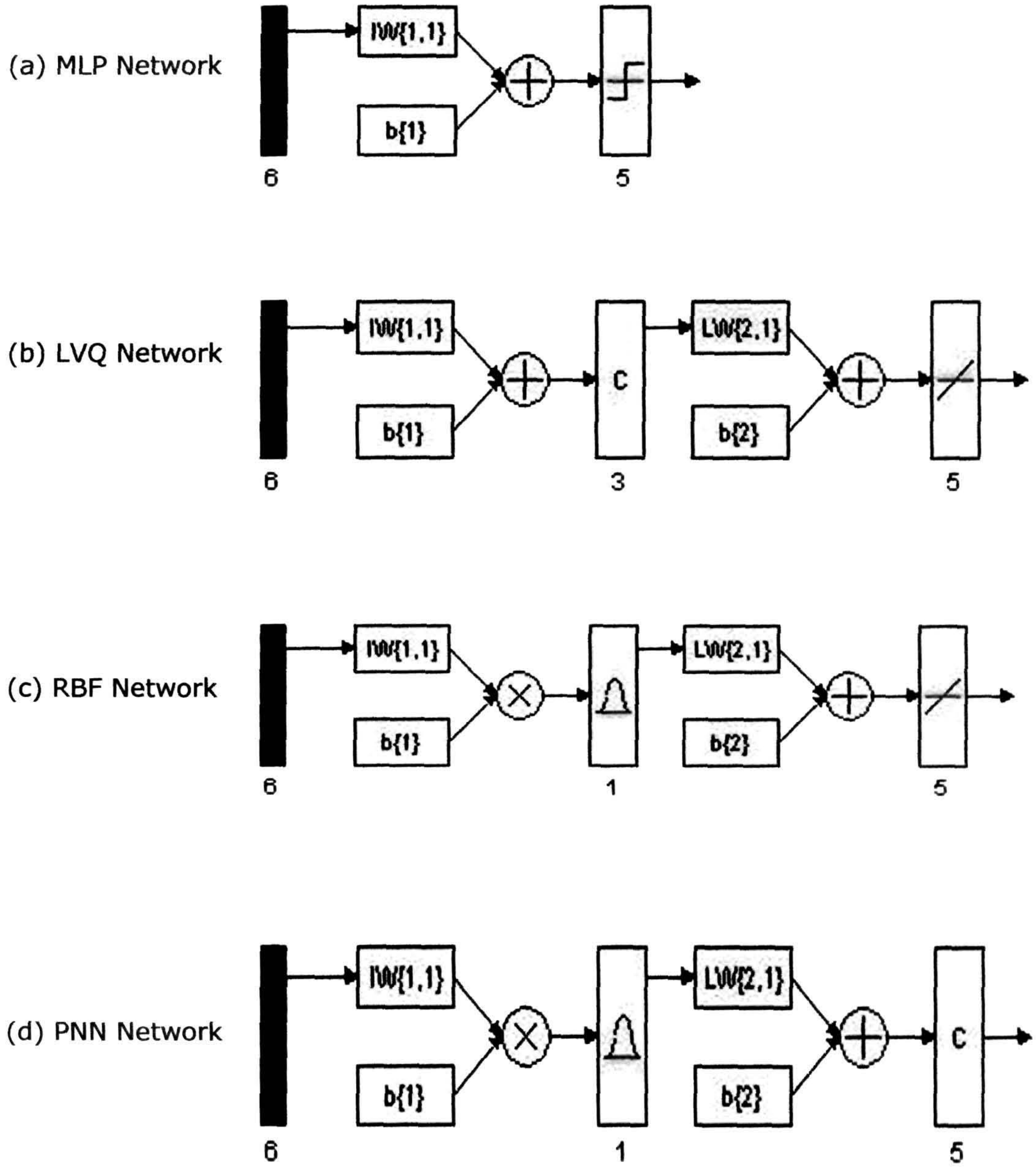
The data sets were analyzed using four supervised ANN classifiers, namely the MLP, LVQ, PNN and RBF paradigms [5, 11]. Training of the neural networks was performed with 50% of the whole data sets and 50% of the data sets were used for testing the neural network paradigms. As described in the section 4.1.4, a total of 150 concatenated data matrices (each of dimensions 4 × 750) were formed, thus 75 data sets were used for training and remaining 75 data sets were used to test the performance of the ANN paradigm. The four ANN paradigms were used to analyse the comparative results from E-nose on tea quality prediction. The aim of this comparative study was to identify the most appropriate ANN paradigm, which can be trained with the high accuracy, to predict the quality of tea samples. The figure 4.6 is shown with the structures of the four ANN paradigms and Table 4.4 summarizes the architectures and the results of correct classification rates of ANN paradigms, which were used for experimental training and discrimination of overlapping tea flavour quality samples, and thus it's quality determination.

Table: 4.4: Architecture of different artificial neural networks and correct classification results.

Neural networks	Architecture	Classification (%)
Multi-layer perceptron (MLP)	For this network the transfer function was HARDLIM and learning function was LEARNP	88
Learning vector quantization (LVQ)	Three hidden neurons were used for this network. Output class percentage was [0.2, 0.2, 0.2, 0.2, 0.2] and learning rate was 0.0125	89
Probabilistic neural network (PNN)	It was very similar to RBF network with a competitive output layer and SPREAD, the constant was set as 1.0	94
Radial basis function (RBF) network	Radial basis networks consist of two layers: a hidden radial basis layer and an output linear layer. For this network SPREAD, the constant was set as 1.0	100

4.1.6.1 Multi Layer Perceptron

An MLP network was programmed in MATLAB environment with learning rate equal to 0.3 and a momentum term equal to 0.4. Its architecture is shown in the figure 4.6 (a). It had six inputs, four from sensors response and two from temperature and humidity changes. The hidden neurons were kept five and five output neurons were chosen for five overlapping tea flavour samples. The network was able to reach a classification rate of 88%. MLP network completed its training after approximately


(a) MLP Network

(b) LVQ Network

(c) RBF Network

(d) PNN Network

Figure 4.6: The architectures of four selected ANN paradigms for overlapping tea flavour quality prediction

20,000 epochs taking about 7 hours of training time. Since MLP is poor in adapting the uneven distribution of samples and thus the classifications rate was less compared to the other paradigms.

4.1.6.2 Linear Vector Quantization

The networks had six input and five output neurons and a variable number of nodes in the competitive layer, as shown in figure 4.6 (b). In this case, the training was performed in two stages. Initially the network was trained with a learning rate equal to 0.01 and the conscience factor was set equal to 1. With this option, the class winner is always moved towards the input vector if it is in the right class and moved away from the input vector if it is in the wrong class. In the next stage, once a relatively good solution had been found, the boundaries between zones were modified where misclassifications occurred and thus the solution was further refined. The learning rate was set to 0.0135. LVQ was able to correctly classify 89% of the tea sample response vectors. LVQ network completed it's training after approximately 3000 epochs taking about 2 hours of training time. LVQ can adapt better to the uneven distribution of samples and thus the classifications rate was higher compared to the MLP paradigm.

4.1.6.3 RBF and PNN

These networks' architectures are shown in figure 4.6 (c) and (d). Neurons are added to the network until the sum-squared error (SSE) reduced to an error goal of 0.000001, or a maximum number of internal neurons were reached to 150. Initially, the spread parameter was kept large enough so that the radial basis neurons could respond only to overlapping regions of the input space, but not so large that all the neurons could respond in essentially the same manner [5, 11, 12]. For both the networks the spread parameter was set to 1.0. The PNN had classified 94% of the response vectors correctly whereas the RBF network's classification rate was 100%. RBF and PNN can easily adjust their weight matrices to match the morphological variability of the patterns. They performed better than MLP network in the discrimination rates for overlapping tea flavour samples.

Table: 4.5: The training performance of Artificial Neural Network paradigms

Network	Training Time	No. of epochs
MLP	7 Hours	20,000
LVQ	2 Hours	3000
RBF	57 Minutes	170
PNN	61 Minutes	180

4.1.6.4 Neural network training performance

The training performances of the four ANN paradigms are shown in the Table 4.5. The MLP network required typically 20,000 training iterations, LVQ required 3000 training iterations, RBF required only 170 training iterations and PNN needed only 180 training iterations. Thus, the time required to train MLP is longer than that of RBF, PNN and LVQ. MLP got trained in approximately 7 hours duration with tea data vectors whereas LVQ took 2 hours, PNN and RBF took 61 and 57 minutes, respectively to complete the training. These figures are based on computations using a PC with processor speed of 866 MHz and Pentium III processor. A *t*-test was performed to check whether the RBF, PNN and LVQ algorithms were performing better than the MLP in terms of the total number of patterns correctly classified. The test demonstrated that there was no significant difference among the mean number of patterns misclassified by the RBF, PNN and LVQ and the MLP.

4.1.7 Results and Discussions

In this part of research experiments the flavours of five overlapping tea samples were discriminated by using an ANN based E-nose system. The odour patterns (response data) from five tea samples were gathered with the help of E-nose sensors and associated electronic circuitry. Five tea flavours were identified as flavour clusters with the help of PCA, FCM and SOM of the sensor responses. The E-nose system has successfully discriminated the flavours of tea samples manufactured under nonstandard processing conditions, viz. over-fermented, over-fired, under-fermented with the help of linear data processing technique of PCA. Then MLP, LVQ, RBF and PNN network algorithms were applied to the response data for classification of the five overlapping tea samples. An accuracy of 100% was reached in the classification rate using RBF network and 94% by using PNN. It was found that these performances were comparably better than that of trained MLP (88%) and LVQ (89%). Finally, the training time of RBF and PNN were found to be faster than MLP and LVQ as explained in Table 4.5. It was also found that the networks had a good performance reaching 100% accuracy on the classification rates of patterns belonging to previously trained samples. If a new and untrained tea sample were offered during testing, RBF, PNN and LVQ associated this pattern to a tea sample which was the nearest to the actual tea sample already known. These characteristics termed the RBF and PNN network superior in the pattern classification in the context of intelligent instruments.

In these experiments the two major drift effects due to environmental temperature and humidity variations were considered. Variations of temperature and humidity were acquired online as two additional parameters for the drift analysis. To consider temperature and humidity drift effects on the sensors' responses data, two extra input nodes were added to the ANN paradigms. The temperature and humidity parameters were forced as two additional inputs to the added nodes. The ANNs trained with temperature and humidity, as additional inputs were able to adjust the weights according to the changes in temperature, humidity and sensors' responses. This resulted in faster and more accurate training of the ANNs, which increased classification rate, e.g. 100% with an RBF network.

RBF, PNN and LVQ algorithms can adapt to the distribution in the databases where the number of patterns are complex and uneven. Thus, while RBF, PNN and LVQ were able to classify most of the patterns corresponding the tea samples, MLP was less able to adapt to the uneven distribution of samples. RBF and PNN were able to adjust their network weights of generalization to match the morphological variability of the patterns. They produced better results and training performances than MLP in the discrimination of the tea sample flavours. In case of the LVQ algorithm, when a relatively good solution had been found, modifying the boundaries between zones where misclassification occurred further refined the classification results. It can be easily concluded that an ANN and MOS based E-nose system had provided high discrimination rate for the five flavours of the overlapping tea samples.

4.2 Analysis of Non-overlapping Tea Flavour Samples

For the second part of the experiments, a group of tea samples was chosen from standard tea flavour terms. These flavour terms are used to indicate the tea quality, which may be due to variable agronomical and agricultural practices in growing the tea plantations apart from geographical and climatic effects. The E-nose system has successfully discriminated standard tea flavour terms, which form a part of the tea flavour wheel [3] containing normally used tea terms in tea trade.

The term 'non-overlapping' is used to indicate that these flavours are taken from different tea species and thus the samples are very distinct and appeared quite separate clusters when analysed by PCA cluster technique. The better experimental setup and more efficient analysis had also contributed to produce improved results. It can be remembered that in case of overlapping samples, the species of the tea flavours were not taken into account but the same class of tea sample was used to manufacture different tea flavours by varying environmental conditions (temperature and humidity) and timing requirements at the time of tea processing. Also it was noted that the nonstandard manufacturing conditions produced five different tea flavour samples, however, when analysed, they had shown quite overlapping characteristics. This was expected, as the basic tea sample was the same. However, in case of non-overlapping, the samples were naturally different due to different tea species and additionally, other samples were prepared at laboratory by infusing external flavours of mint, paper, bag etc. as these would often be present in the market.

4.2.1 Sample Preparation

In this part of experiments on the non-overlapping tea samples, the sample size comprised of 60 tea samples of 10 different flavour notes of non-overlapping nature. Thus, each of 10 flavours has 6 samples from 6 sources. The samples were gathered from independent sources of Tea Gardens of Noorbari and Sonabhil, Tezpur (Assam), Golden Tea Company, Tezpur, (Assam), Tea garden of Science and Technology Entrepreneurs' Park, Indian Institute of Technology, Kharagpur and Tea Tasters, Green Gold Pvt. Ltd, (Guwahati) and J. Thomas Pvt. Ltd. at Guwahati, India. The samples are as following:

(i) Bakey
(ii) Sour
(iii) Woody
(iv) Sweet
(v) Musty
(vi) Poor
(vii) Smokey
(viii) Minty
(ix) Baggy
(x) Papery

These tea flavours are defined in appendix 1. These flavour terms are a part of the standard tea terminology mentioned in the tea flavour wheel, which is explained in Appendix 2.

The last four tea flavour samples i.e. smokey, minty, baggy and papery are prepared at the laboratory. However, the basic tea samples for these laboratory prepared flavour samples were collected from the same 6 independent sourced as described above. The smokey flavour is normally formed during drying process of the tea if temperature exceeds the given limit for comparably long time. This flavour was formed in laboratory by drying black tea in oven at an elevated temperature until fumes started evaporating from the tea. Minty tea is quite common in Japan and the mint is found sometimes in the herbal tea. The herbal tea may also have more added flavours. The minty tea flavour sample was prepared in the laboratory by adding dry mint leave powder to black tea and then hermetically packaged it for at least two weeks so that the mint flavour well diffused into the black tea.

The baggy term is normally used to describe the tea flavour quality, which is loose packed into gunny bags for long in transportation and storage. This flavour was prepared by packing the loose black tea into gunny bag for 30 days. A characteristic smell of gunny bag is infused into the tea. This sample, therefore, represented a common tea variety available in the market by virtue of being packed in gunny bags.

The papery tea flavour sample was prepared similar to the baggy tea sample except that the black tea was packed in a thick paper bag instead of gunny bag. The idea behind this sample is also based on the fact that a good amount of tea packaging is done in paper bags. The characteristic smell of paper diffuses into tea grains, which makes papery flavour tea sample.

4.2.2 Experimental Procedure

The non-overlapping tea samples (listed in the section 4.2.1) are prepared in a different way than the overlapping samples (explained in sections 4.1.1 and 4.1.2). Tea samples are opened from the packs and put in a stainless steel container. The container, which was connected through pipes to the diaphragm pump system, is placed in an oven where the temperature is maintained at about 100 ^{o} C. The idea of keeping temperature at 100 ^{o} C was based on the fact that the tea is prepared by infusing it into boiling water (at the temperature of 100^{o} C). Elevated temperature caused increased release of VOC's, thus better response was observed due to increased parts per million (ppm).

The block diagram of experimental set up is shown in the figure 4.7. Four sensor chambers are connected through plastic pipes to the flow path junction, which is connected to the sample, and reference vessels through flow control devices. The reference vessel is placed in an open space in room environment at about 25^{o} C temperature and 65% relative humidity with deviation less than ±2%. Flow control devices provide constant flow of tea sample flavour VOCs and air to the sensors alternately at predefined time intervals.

One measurement comprises of taking, alternatively, a headspace sample for 200 seconds from the tea vessel followed by the reference vessel for 300 seconds. During the process of the measurements, responses were stored at every second (at the rate of 1 Hz.) in a data file for subsequent processing (Figure. 3.17, chapter-3). Each data set was acquired for 200 seconds and 300 seconds was allowed for refreshing to air. Similarly all ten samples were tested by E-nose and response was recorded. Similar to overlapping measurements, in this case also there was no significant change in the base line during the experiments and consistent classification results were observed while data processing.

4.2.3 Data Acquisition

Data acquisition was performed in a way similar to overlapping samples (section 4.1). The visual display of the data was performed in LabVIEW© (National Instruments Inc.) and GENIE (ADVANTECH®, American Advantech Corporation) environments.

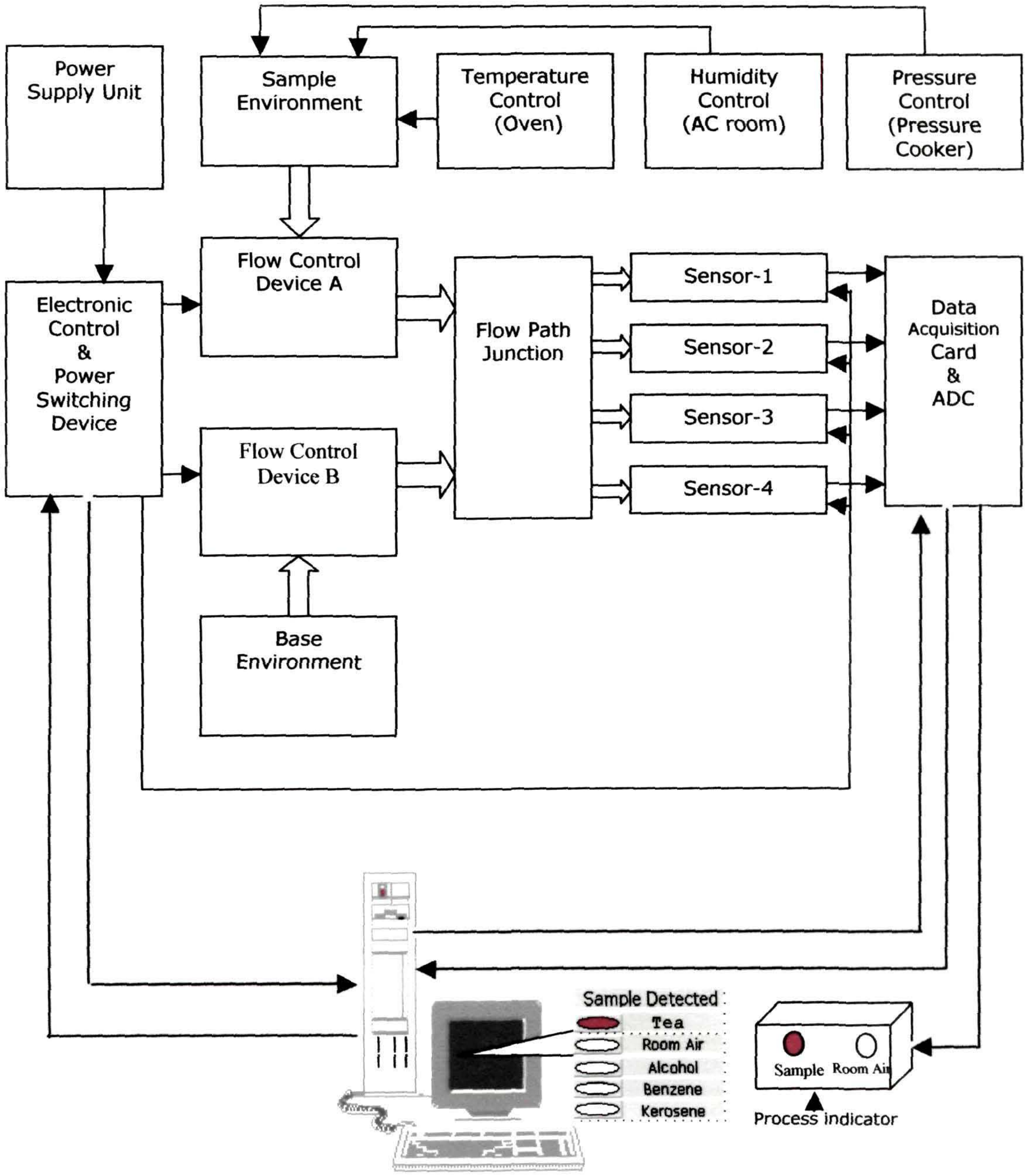


Figure 4.7: Functional block diagram of E-Nose System for non-overlapping tea samples

In this case also, the data range for analysis was chosen from the most stable part of the sensors' responses to achieve better classification results. The ten different non-overlapping tea sample response data sets were acquired from E-nose sensors' response. For each of the ten tea flavours, 40 data sets resulting in a total of 400 data sets for ten tea samples were acquired. In a cycle of one data set of dimensions 4×500 (200 seconds sample reading and 300 seconds refreshing time) a data set of dimensions 4×100 of most stable part of response is used for analysis. Each data set consisted of 4000 data vectors (total for 4 sensors for 100 data vectors in one data set) each of dimensions 1×100. Thus the ratio of data collected to data used is 5:1(actual measured dimensions were 4×500 but used 4×100 only). The experiment time was about 138.8 hours for each tea sample, thus a total of 1388 hours (about 58 days) including refreshing time for ten tea samples. A rest period of one day was observed between each tea sample testing to allow sensors to be completely free of tea vapour trace of one type. It took about 114 days to complete experimentations on non-overlapping tea samples.

4.2.4 Data Processing

From the E-nose data response, ten data matrices (each of dimensions of 4×100 for 4 sensors) for ten-tea samples were concatenated into a single matrix of dimensions of 4×1000. This resulted in a single data set made of the ten data sets from ten tea samples. A total of 200 such data matrices (each of dimensions 4×1000) were formed from 200 data sets and the remaining 200 were used for testing. The response of the E-nose sensors for non-overlapping samples is displayed in figure 3.17 in chapter 3, using GENIE environment.

The model used for the data pre-processing is the change in sensors' output voltage as given by the equation 4.2,

$$\mathrm{dV}_i = V_{\mathrm{air},\, i} - V_{\mathrm{odour},\, i} \tag{4.2}$$

Where:

dVi = Change in the output voltage of the i^{th} sensor when exposed to the tea vapour sample.

$V_{\mathrm{air},\ i}$ = The output voltage, also called baseline voltage of the sensor when exposed to the room air

$V_{\text{odour},i}$ = The output voltage of the sensor when exposed to the tea vapour sample.
i = 1,2...4, Number of sensors used in experiments.

The data sets were then normalized by dividing each dV by the maximum value of sensor output voltage to set the range of output voltage between *0* and *1*. The normalization was applied to the data sets prior to principal component analysis (PCA) and the analysis using the MLP, LVQ, PNN and RBF networks, which were simulated on the MATLAB 6.1 environment. For example, a sample of the normalization of sensor response data is shown in Table 4.6. The first 4 columns show the actual response (in volts) of 4 sensors used in E-nose experiments. The data set (4×125) for this sample is extracted from most stable part of the E-nose response from a baggy tea sample. Table 4.7 shows the statistical parameters of each sensor's response. Data vector for each sensor is divided by maximum response value for corresponding sensor. Columns 5 through 8 in table 4.6 show the normalized vectors for corresponding to 4 sensors' output response for baggy tea sample. The output response is in the range from *0* to *1* (normalised) and units are in volts. These data sets, before and after normalization are graphically plotted in figure 4.8 and figure 4.9 respectively.

Table 4.6: Sensors' response before and after normalizations for baggy tea sample (contd).

Sensors' actual response (volts)				Sensors' normalized response				Sensors' actual response (volts)				Sensors' normalized response			
Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-
0 4960	0 3470	0 3320	0 4880	0 9617	0 9345	0 9228	0 8062	0 5050	0 3590	0 3470	0 5640	0 9904	0 9759	0 9807	0 9880
0 4960	0 3470	0 3340	0 4930	0 9617	0 9345	0 9305	0 8182	0 5050	0 3560	0 3470	0 5620	0 9904	0 9655	0 9807	0 9833
0 4960	0 3470	0 3320	0 4980	0 9617	0 9345	0 9228	0 8301	0 5080	0 3560	0 3470	0 5640	1 0000	0 9655	0 9807	0 9880
0 4960	0 3470	0 3320	0 5030	0 9617	0 9345	0 9228	0 8421	0 5080	0 3540	0 3490	0 5620	1 0000	0 9586	0 9884	0 9833
0 4960	0 3490	0 3370	0 5030	0 9617	0 9414	0 9421	0 8421	0 5050	0 3540	0 3470	0 5620	0 9904	0 9586	0 9807	0 9833
0 4960	0 3490	0 3340	0 5050	0 9617	0 9414	0 9305	0 8469	0 5050	0 3520	0 3470	0 5620	0 9904	0 9517	0 9807	0 9833
0 4980	0 3490	0 3370	0 5130	0 9681	0 9414	0 9421	0 8660	0 5050	0 3560	0 3490	0 5620	0 9904	0 9655	0 9884	0 9833
0 4980	0 3520	0 3370	0 5180	0 9681	0 9517	0 9421	0 8780	0 5050	0 3540	0 3490	0 5620	0 9904	0 9586	0 9884	0 9833
0 4980	0 3520	0 3390	0 5180	0 9681	0 9517	0 9498	0 8780	0 5050	0 3560	0 3490	0 5620	0 9904	0 9655	0 9884	0 9833
0 5000	0 3540	0 3370	0 5180	0 9744	0 9586	0 9421	0 8780	0 5080	0 3560	0 3470	0 5620	1 0000	0 9655	0 9807	0 9833
0 4980	0 3520	0 3390	0 5180	0 9681	0 9517	0 9498	0 8780	0 5050	0 3540	0 3490	0 5620	0 9904	0 9586	0 9884	0 9833
0 5000	0 3540	0 3390	0 5220	0 9744	0 9586	0 9498	0 8876	0 5050	0 3540	0 3490	0 5590	0 9904	0 9586	0 9884	0 9761
0 5000	0 3540	0 3390	0 5220	0 9744	0 9586	0 9498	0 8876	0 5080	0 3540	0 3470	0 5590	1 0000	0 9586	0 9807	0 9761
0 5000	0 3540	0 3420	0 5250	0 9744	0 9586	0 9614	0 8947	0 5050	0 3540	0 3490	0 5620	0 9904	0 9586	0 9884	0 9833
0 5000	0 3560	0 3370	0 5250	0 9744	0 9655	0 9421	0 8947	0 5050	0 3520	0 3490	0 5570	0 9904	0 9517	0 9884	0 9713

0 5000	0 3590	0 3370	0 5300	0 9744	0 9759	0 9421	0 9067	0 5050	0 3540	0 3520	0 5590	0 9904	0 9586	1 0000	0 9761
0 5000	0 3590	0 3420	0 5300	0 9744	0 9759	0 9614	0 9067	0 5050	0 3540	0 3490	0 5590	0 9904	0 9586	0 9884	0 9761
0 5000	0 3590	0 3420	0 5320	0 9744	0 9759	0 9614	0 9115	0 5050	0 3540	0 3520	0 5590	0 9904	0 9586	1 0000	0 9761
0 5030	0 3610	0 3390	0 5350	0 9840	0 9828	0 9498	0 9187	0 5050	0 3540	0 3470	0 5570	0 9904	0 9586	0 9807	0 9713
0 5030	0 3610	0 3390	0 5370	0 9840	0 9828	0 9498	0 9234	0 5050	0 3540	0 3490	0 5590	0 9904	0 9586	0 9884	0 9761
0 5030	0 3610	0 3390	0 5400	0 9840	0 9828	0 9498	0 9306	0 5050	0 3520	0 3470	0 5590	0 9904	0 9517	0 9807	0 9761
0 5030	0 3640	0 3420	0 5420	0 9840	0 9931	0 9614	0 9354	0 5050	0 3520	0 3490	0 5620	0 9904	0 9517	0 9884	0 9833
0 5030	0 3640	0 3420	0 5400	0 9840	0 9931	0 9614	0 9306	0 5050	0 3520	0 3470	0 5620	0 9904	0 9517	0 9807	0 9833
0 5030	0 3640	0 3420	0 5420	0 9840	0 9931	0 9614	0 9354	0 5030	0 3540	0 3470	0 5590	0 9840	0 9586	0 9807	0 9761
0 5030	0 3640	0 3420	0 5470	0 9840	0 9931	0 9614	0 9474	0 5050	0 3520	0 3470	0 5570	0 9904	0 9517	0 9807	0 9713
0 5050	0 3610	0 3440	0 5470	0 9904	0 9828	0 9691	0 9474	0 5030	0 3540	0 3490	0 5570	0 9840	0 9586	0 9884	0 9713
0 5030	0 3610	0 3440	0 5470	0 9840	0 9828	0 9691	0 9474	0 5030	0 3540	0 3490	0 5540	0 9840	0 9586	0 9884	0 9641
0 5050	*0 3610*	*0 3470*	*0 5490*	*0 9904*	*0 9828*	*0 9807*	*0 9522*	*0 5030*	*0 3520*	*0 3490*	*0 5520*	*0 9840*	*0 9517*	*0 9884*	*0 9593*
0 5050	0 3640	0 3440	0 5520	0 9904	0 9931	0 9691	0 9593	0 5050	0 3520	0 3470	0 5540	0 9904	0 9517	0 9807	0 9641
0 5050	0 3640	0 3420	0 5540	0 9904	0 9931	0 9614	0 9641	0 5030	0 3520	0 3470	0 5540	0 9840	0 9517	0 9807	0 9641
0 5050	0 3640	0 3470	0 5520	0 9904	0 9931	0 9807	0 9593	0 5050	0 3490	0 3490	0 5520	0 9904	0 9414	0 9884	0 9593
0 5050	0 3610	0 3440	0 5520	0 9904	0 9828	0 9691	0 9593	0 5050	0 3490	0 3470	0 5540	0 9904	0 9414	0 9807	0 9641
0 5030	0 3590	0 3440	0 5540	0 9840	0 9759	0 9691	0 9641	0 5030	0 3470	0 3490	0 5540	0 9840	0 9345	0 9884	0 9641
0 5050	0 3590	0 3440	0 5570	0 9904	0 9759	0 9691	0 9713	0 5000	0 3470	0 3520	0 5520	0 9744	0 9345	1 0000	0 9593
0 5050	0 3610	0 3470	0 5590	0 9904	0 9828	0 9807	0 9761	0 5030	0 3470	0 3520	0 5520	0 9840	0 9345	1 0000	0 9593
0 5050	0 3610	0 3420	0 5590	0 9904	0 9828	0 9614	0 9761	0 5030	0 3440	0 3520	0 5520	0 9840	0 9241	1 0000	0 9593
0 5050	0 3610	0 3440	0 5590	0 9904	0 9828	0 9691	0 9761	0 5030	0 3420	0 3520	0 5470	0 9840	0 9172	1 0000	0 9474
0 5050	0 3640	0 3440	0 5590	0 9904	0 9931	0 9691	0 9761	0 5030	0 3420	0 3520	0 5490	0 9840	0 9172	1 0000	0 9522
0 5050	0 3640	0 3440	0 5590	0 9904	0 9931	0 9691	0 9761	0 5030	0 3440	0 3520	0 5470	0 9840	0 9241	1 0000	0 9474
0 5080	0 3660	0 3470	0 5590	1 0000	1 0000	0 9807	0 9761	0 5000	0 3440	0 3490	0 5490	0 9744	0 9241	0 9884	0 9522
0 5050	0 3640	0 3470	0 5620	0 9904	0 9931	0 9807	0 9833	0 5000	0 3470	0 3490	0 5470	0 9744	0 9345	0 9884	0 9474
0 5080	0 3610	0 3470	0 5640	1 0000	0 9828	0 9807	0 9880	0 5030	0 3420	0 3490	0 5470	0 9840	0 9172	0 9884	0 9474
0 5080	0 3640	0 3470	0 5640	1 0000	0 9931	0 9807	0 9880	0 5000	0 3420	0 3490	0 5440	0 9744	0 9172	0 9884	0 9402
0 5050	0 3660	0 3470	0 5620	0 9904	1 0000	0 9807	0 9833	0 5030	0 3420	0 3490	0 5470	0 9840	0 9172	0 9884	0 9474
0 5050	0 3640	0 3490	0 5620	0 9904	0 9931	0 9884	0 9833	0 5000	0 3440	0 3520	0 5440	0 9744	0 9241	1 0000	0 9402
0 5050	0 3590	0 3490	0 5640	0 9904	0 9759	0 9884	0 9880	0 5000	0 3420	0 3490	0 5440	0 9744	0 9172	0 9884	0 9402
0 5050	0 3610	0 3490	0 5640	0 9904	0 9828	0 9884	0 9880	0 5000	0 3420	0 3470	0 5420	0 9744	0 9172	0 9807	0 9354
0 5080	0 3610	0 3470	0 5640	1 0000	0 9828	0 9807	0 9880	0 5000	0 3390	0 3490	0 5440	0 9744	0 9069	0 9884	0 9402
0 5080	0 3590	0 3490	0 5620	1 0000	0 9759	0 9884	0 9833	0 5000	0 3390	0 3490	0 5400	0 9744	0 9069	0 9884	0 9306
0 5080	0 3610	0 3490	0 5690	1 0000	0 9828	0 9884	1 0000	0 5000	0 3390	0 3490	0 5400	0 9744	0 9069	0 9884	0 9306
0 5050	0 3610	0 3470	0 5640	0 9904	0 9828	0 9807	0 9880	0 5000	0 3420	0 3490	0 5370	0 9744	0 9172	0 9884	0 9234
0 5050	0 3610	0 3470	0 5640	0 9904	0 9828	0 9807	0 9880	0 5000	0 3420	0 3490	0 5420	0 9744	0 9172	0 9884	0 9354
0 5050	0 3560	0 3520	0 5640	0 9904	0 9655	1 0000	0 9880	0 4980	0 3420	0 3490	0 5400	0 9681	0 9172	0 9884	0 9306
0 5050	0 3590	0 3490	0 5640	0 9904	0 9759	0 9884	0 9880	0 5000	0 3420	0 3490	0 5400	0 9744	0 9172	0 9884	0 9306
0 5050	0 3590	0 3470	0 5640	0 9904	0 9759	0 9807	0 9880	0 5000	0 3440	0 3490	0 5350	0 9744	0 9241	0 9884	0 9187
0 5050	0 3560	0 3470	0 5620	0 9904	0 9655	0 9807	0 9833	0 5000	0 3440	0 3490	0 5320	0 9744	0 9241	0 9884	0 9115
0 5080	0 3560	0 3470	0 5640	1 0000	0 9655	0 9807	0 9880	0 4980	0 3420	0 3470	0 5350	0 9681	0 9172	0 9807	0 9187
0 5080	0 3540	0 3490	0 5620	1 0000	0 9586	0 9884	0 9833	0 4980	0 3420	0 3490	0 5320	0 9681	0 9172	0 9884	0 9115
0 5050	0 3540	0 3470	0 5620	0 9904	0 9586	0 9807	0 9833	0 5030	0 3390	0 3490	0 5300	0 9840	0 9069	0 9884	0 9067
0 5050	0 3520	0 3470	0 5620	0 9904	0 9517	0 9807	0 9833	0 4980	0 3420	0 3490	0 5320	0 9681	0 9172	0 9884	0 9115
0 5050	0 3560	0 3490	0 5620	0 9904	0 9655	0 9884	0 9833	0 4980	0 3420	0 3490	0 5300	0 9681	0 9172	0 9884	0 9067
0 5050	0 3540	0 3490	0 5620	0 9904	0 9586	0 9884	0 9833	0 4980	0 3420	0 3470	0 5300	0 9681	0 9172	0 9807	0 9067
0 5050	0 3560	0 3490	0 5620	0 9904	0 9655	0 9884	0 9833								

Table 4.8 Concluded

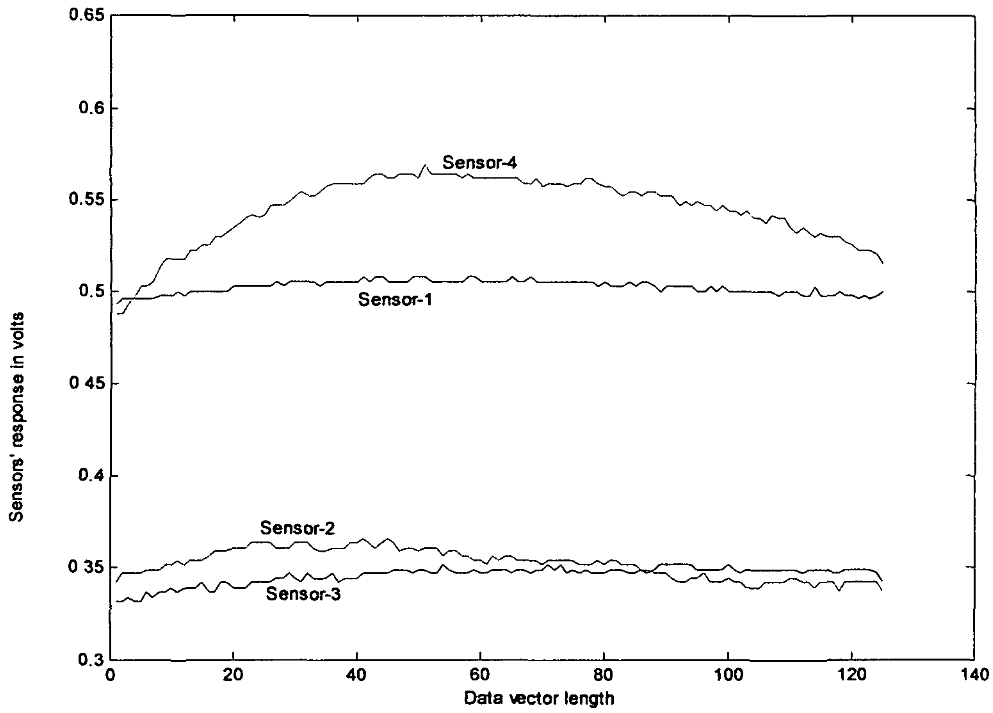


Figure 4.8: Display of the most stable part of sensors' response to baggy tea sample data given in Table 4.6 before normalization.

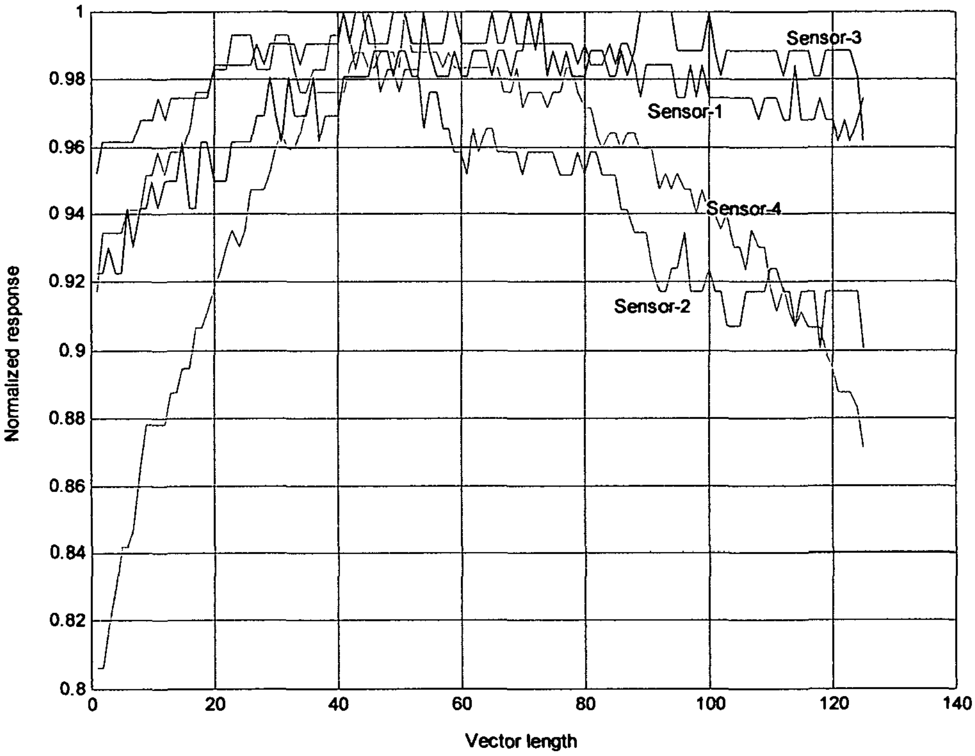


Figure 4.9: The graphical representation of normalized data set (Table 4.6) received as response from baggy tea sample.

Table 4.7: Sensors' response statistics before and after normalization of data for baggy tea sample.

Statistical property	Sensors' actual response (volts)				Sensors' normalized response			
	Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4
Max value	0.5080	0.3660	0.3520	0.5690	1.0000	1.0000	1.0000	1.0000
Min Value	0.4930	0.3370	0.3320	0.4880	0.9521	0.9000	0.9228	0.8062
Mean	0.5024	0.3522	0.3459	0.5446	0.9822	0.9526	0.9765	0.9416
Std. Dev.	0.0034	0.0081	0.0048	0.0185	0.0109	0.0279	0.0187	0.0443

Figure 4.8 is shown with the graphical representation of the most stable data response (for data, refer the Table 4.6) from 4 E-nose sensors prior to the normalization. It can be observed from these graphs that the spread in data is increased after normalization. This is also evident from the Table 4.7 that standard deviation is increased after normalization. Figure 4.8 and figure 4.9 are generated by using data set of Table 4.7 and MATLAB® function *plot*. The range of normalized data set (difference between sample and room air response) is higher resulting in better classification rate.

4.2.5 Data Clustering Analysis

Principal Component Analysis was the main method for the data clustering technique for non-overlapping tea samples. This cluster technique is quite commonly used and thus it was a natural choice for non-overlapping tea sample analysis. It generated concentrated clusters in the reduced dimensionality diagram display (figure 4.10).

Table 4.8: The results of PCA analysis for ten non-overlapping tea flavour quality samples

PC	% Variance	Eigen-value	Principal Components			
			Sensor$_1$	Sensor$_2$	Sensor$_3$	Sensor$_4$
PC$_1$	95.5182	14.57×10^{-5}	-0.0419	-0.6713	0.6794	0.2935
PC$_2$	2.0319	0.31×10^{-5}	-0.0468	-0.4099	-0.0144	-0.9108
PC$_3$	1.5129	0.23×10^{-5}	-0.9504	0.2330	0.1972	-0.0592
PC$_4$	0.9370	0.14×10^{-5}	-0.3045	-0.5719	-0.7067	0.2842

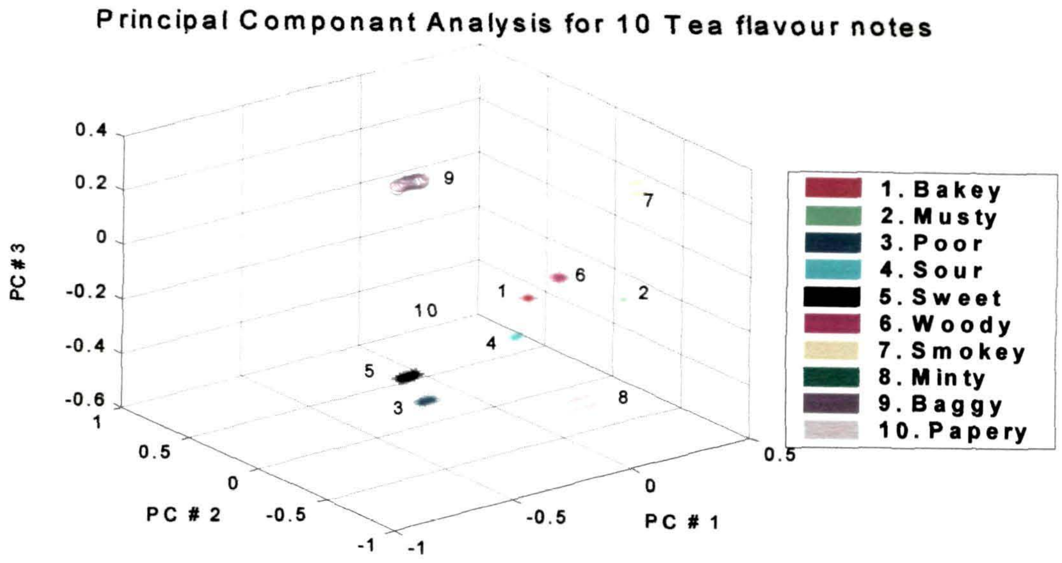


Figure 4.10: PCA analysis display for flavour classification of 10-tea samples of non-overlapping nature.

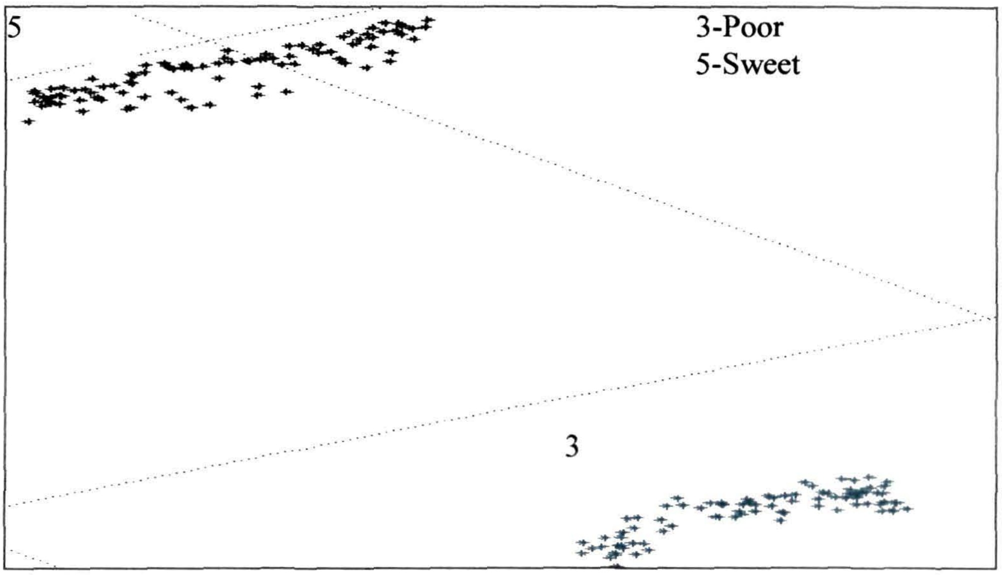


Figure 4.11: Microscopic view of graphical display in figure 4.10 that resulted when zoomed by over 100 times.

4.2.5.1 Principal Component Analysis

PCA, as described in earlier sections has been used to determine the clusters within non-overlapping tea flavours. The result is depicted in figure 4.10. The analysis has established very distinct and separate clusters for tea flavours. The clusters were so densely converged that each of these appeared as a single point in 3-dimensional graphical display (figure 4.10). Cluster diagram was zoomed to observe clusters as a set of separate points (figure 4.11). The normalized data set was used for this analysis and only first three Principal Components were taken into account since they represented most of the variance in the whole data. The statistical results of PCA analysis are shown in the Table 4.8.

The computational work of PCA analysis was performed in MATLAB®. From data matrix (it can be remembered that one data set was a concatenated matrix of dimensions 4 X 1000 made of response data of 10 tea samples) a covariance matrix is calculated by using MATLAB command *cov(dataset)*. The output of *cov(dataset)* function is a covariance matrix. In the data set matrix for *cov(dataset)*, each row is a tea sample observation and each column represents a sensor (1-4). The algorithm for the covariance calculation is:

[n, p] = size(dataset);

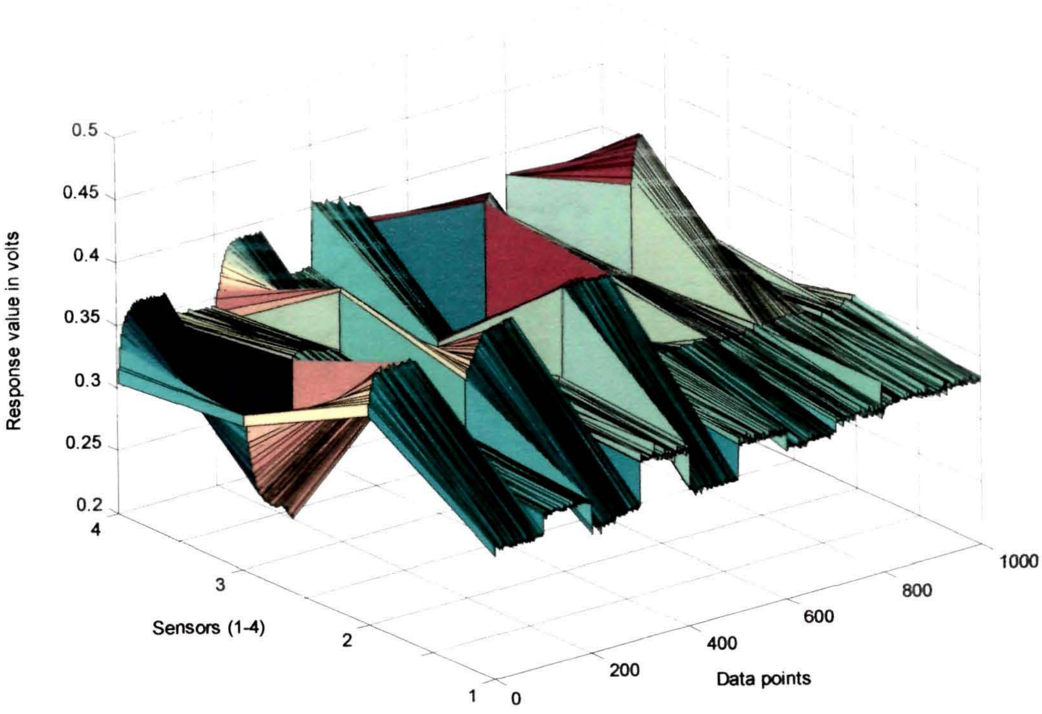


Figure 4.12: 3-D Scatter diagram of response data from E-nose for non-overlapping tea samples.

*dataset = dataset - ones(n,1) * mean(dataset);*

Y = (dataset)'(dataset)/(n-1);*

From the covariance matrix, principal components are calculated. The command at MATLAB prompt, [PC, LATENT, EXPLAINED] = PCACOV(Y) takes the covariance matrix Y and returns the principal components in PC, the eigenvalues of the covariance matrix of Y in LATENT and the percentage of the total covariance in the observations is explained by each eigenvector in EXPLAINED.

The first output 'PC', of the function *PCACOV* contains the principal Components (Table 4.8). These are the linear combinations of the original variables that generate the new variables. Among the first four principal component vectors (Table 4.8), the largest weights in the first row (first principal component) are the third and fourth elements, corresponding to the sensors 3 and 4.

The second output, LATENT, is the eigenvalues of the covariance in the new coordinate system defined by the principal components. The third output, EXPLAINED, is a vector containing the covariance explained by the corresponding column of LATENT values. The percent of the total variability explained by each principal component is calculated by the relation-

% Covariance = (100 * Eigenvalue of covariance) / (sum of all Eigenvalues) (4.3)

It can be seen that the first three principal components represents about 99.063 % of the total variability in the datasets.

4.2.6 Artificial Neural Network Analysis

The data sets of ten non-overlapping tea samples were analyzed using four supervised ANN classifiers, namely the MLP, LVQ, PNN and RBF, same as in the case of overlapping tea samples. Training of the neural networks was performed with 50% of the data sets and 50% of the data sets were used for testing the neural network paradigms. As described in the section 4.2.3 and 4.2.4, a total of 200 concatenated data matrices (each of dimensions 4 × 1000) were used for training and remaining 200 data sets (dimensions 4 X 100 for each of ten tea sample data, thus a total of 2000 data vectors for ten tea samples) were used to test the performance of the ANN classifiers.

The testing was done for individual tea flavour data. The compositions of the four ANN classifiers are similar to the overlapping samples (figure 4.6), except that there were only 4 input neurons and 10 output neurons in this case. The table 4.9 summarizes the architectures of ANN classifiers, which were used for experimental training and discrimination of non-overlapping tea flavour samples and thus its quality determination.

Table 4.9: Architecture of different Neural Networks and correct classification results in % for non- overlapping tea flavours

Neural Networks	Architecture
MLP	4 input Neurons, 6 hidden Neuron, 10 output Neurons, 0.35 Learning rate with Momentum 0.42
LVQ	4 input Neurons, 10 output Neurons, Learning rates 0.011 and 0.012 and Conscious factor 1
PNN	4 input Neurons, 10 Neurons in output layer, Spread constant 0.8
RBF	4 input Neurons, 10 Neurons in output layer, Spread constant 0.8

4.2.6.1 Multiple Layered Perceptron

The figure 4.13 shows a typical MLP neuron connectionist diagram based on the selection of number of input, hidden and output neurons for the non-overlapping tea flavour discrimination. In figure 4.13, for the simplicity, only first neuron is shown to be connected, however, each of the input neurons is connected to the each of the hidden neurons and similarly each of the hidden neurons is connected to the each of output neurons.

An MLP network was programmed in MATLAB environment with learning rate equal to 0.35 and a momentum term equal to 0.42. It had 4 input neurons, 6 hidden neurons and 10 output neurons for ten non-overlapping tea flavour samples (figure 4.13). The network was able to reach a classification rate of 87 %. MLP network completed its training after approximately 1,250,000 epochs taking about 21.8 hours of training time. The training was performed on a PENTIUM IV (2.4 GHz) processor based computer for 200 concatenated data matrices (data sets) of ten non-overlapping tea samples and the training performances are shown in the Table 4.12.

The testing part was done for individual tea flavour by using ANN paradigms one at a time for the 200 data sets for each of the ten tea samples. The comparative result of test performance of ANN paradigms is shown in the Table 4.10.

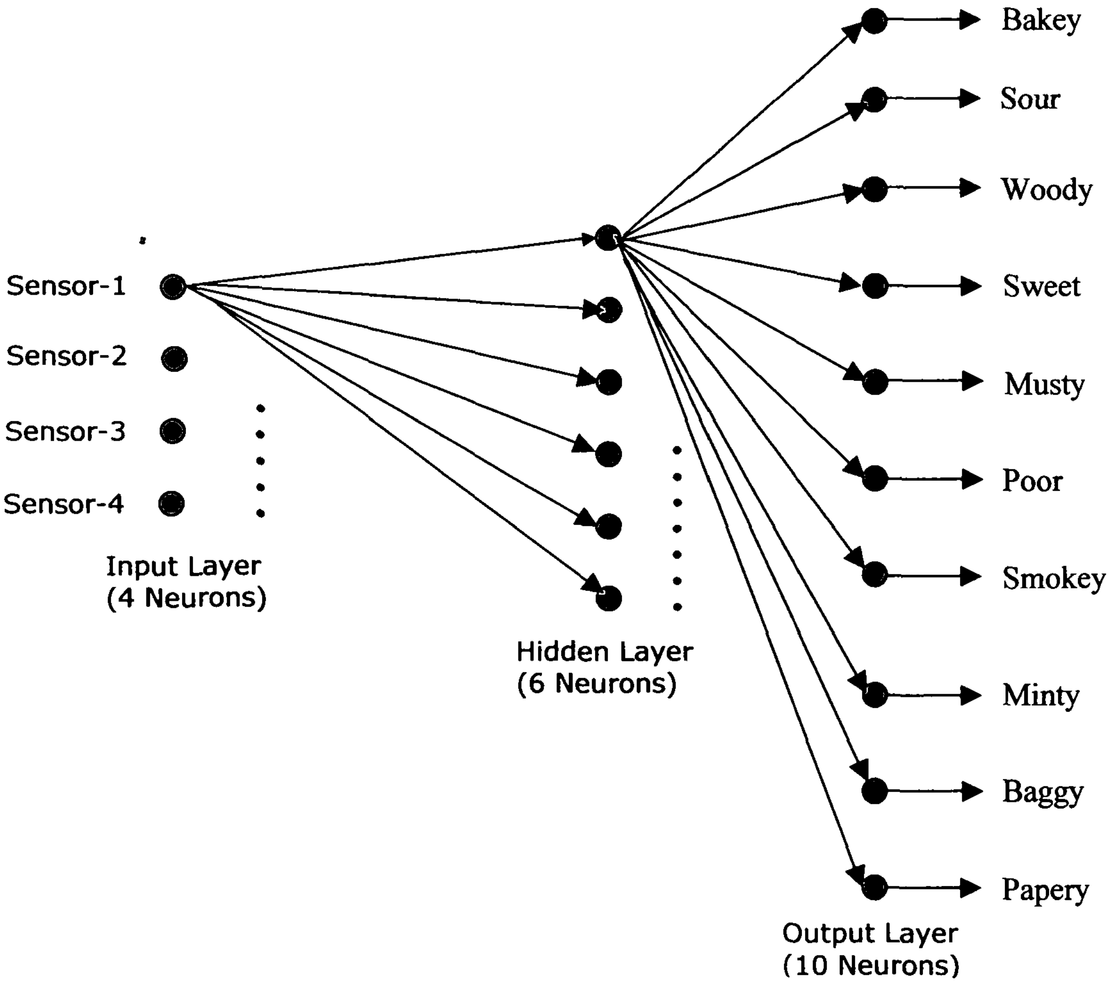


Figure 4.13: Architecture MLP network used for non-overlapping tea samples. Each neuron is connected to the next all neurons through connectionist weights.

4.2.6.2 Learning Vector Quantization

The LVQ network was configured for 4 input and 10 output neurons and a variable number of nodes in the competitive layer. Initially the network was trained with a learning rate equal to 0.011 and the conscience factor equal to 1. In the next stage, by setting learning rate equal to 0.012 further refined the solution. The LVQ net has correctly classify 92% of the 2000 response data vectors. The same 2000 data vectors (200 each of ten tea samples) were used for all 4 ANN techniques to check the relative performances. The number of test vectors, correctly classified samples and % correct rate are shown in the Table 4.11. The neuron configuration at input, hidden and output layers is similar to that of MLP (figure 4.13) except that the hidden neuron were variable, called competitive neurons.

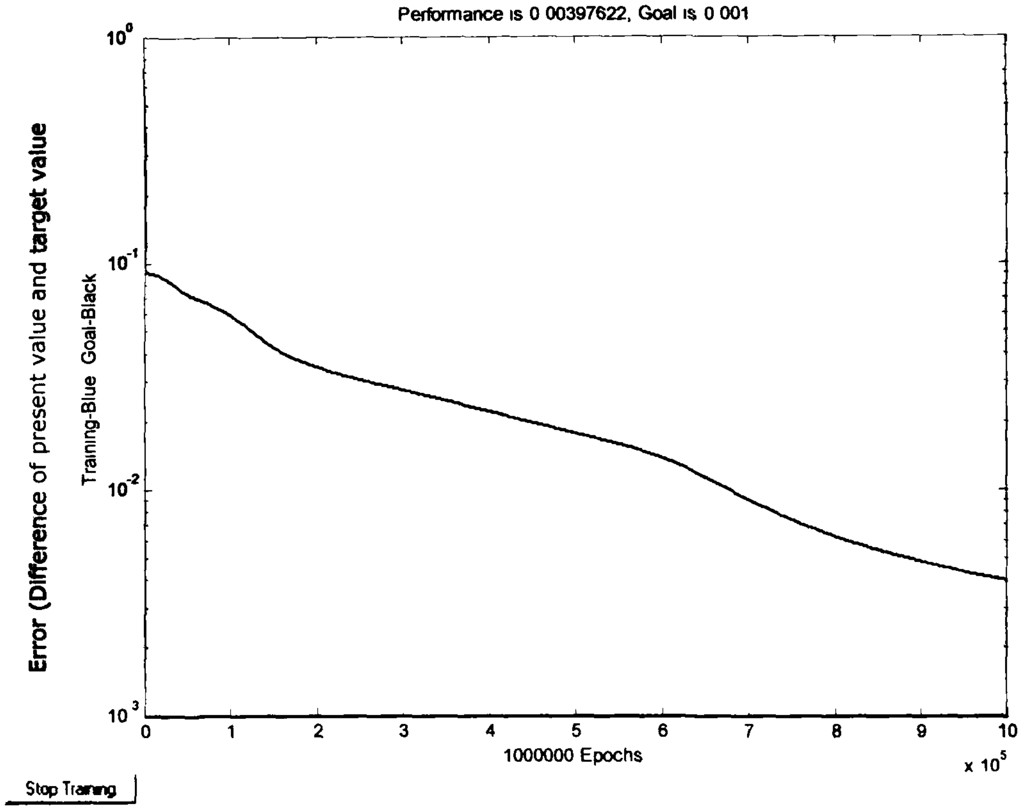


Figure 4.14: Training response of an MLP network in progress for non-overlapping tea samples. It reached about 75% of target value in 10^6 epochs.

4.2.6.3 Probabilistic Neural Network and Radial Basis Function

Neurons were created in net until the sum-squared error falls below the set goal of 0.00001. Table 4.9 shows the architectures of these networks. The spread parameter was kept high enough so that the radial basis neurons responded well to the overlapping regions of the input space. For both the networks the spread parameter was chosen as 0.8, the PNN and RBF classified correctly 93% and 96% respectively of the test vectors. RBF and PNN required 2.96 hours and 3.17 hours, 532 and 563 epochs respectively to complete the training.

Table 4.10: Training and testing of data sets on MLP Network and its correct classification results in % for non- overlapping tea flavours

Tea Sample	*No. of data vector* Used for Training	*No. of data vectors* used for testing	*No. of sample* data correctly classified	Percentage of correct classification
Bakey	200 (4x 1000)	200 (4 X 100)	179	89.5
Sour		200 (4 X 100)	185	92.5
Woody		200 (4 X 100)	167	83.5
Sweet		200 (4 X 100)	175	87.5
Musty		200 (4 X 100)	176	88.0
Poor		200 (4 X 100)	177	88.5
Smokey		200 (4 X 100)	174	87.0
Minty		200 (4 X 100)	166	83.0
Baggy		200 (4 X 100)	170	85.0
Papery		200 (4 X 100)	172	86.0
		Total=2000 (4X100)	Total=1741	87.05%

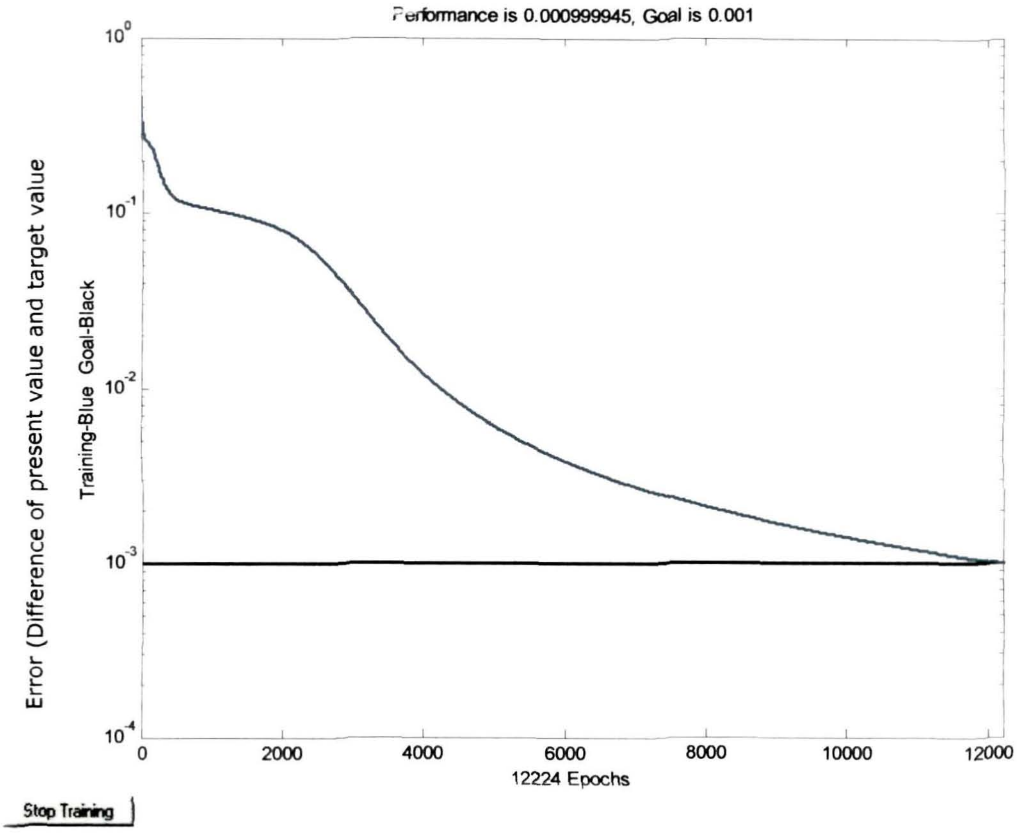


Figure 4.15: Typical training response of an LVQ network for non-overlapping tea samples. It reached the target value in 12224 epochs.

Table 4.11: No. of data vectors used for testing of the ANN paradigms and their correct classification results in % for ten non- overlapping tea flavours

Method	Total No. of data samples tested	No. of Correctly identified samples	% Classification
MLP	2000	1741	87.05
LVQ	2000	1846	92.30
PNN	2000	1862	93.10
RBF	2000	1923	96.15

Table: 4.12: The training performance of Artificial Neural Network paradigms for non-overlapping tea samples.

Network	Training Time required	No. of epochs to reach target error level
MLP	21.8 Hours	1,250,000
LVQ	6.25 Hours	12224
RBF	2.96 Hours	532
PNN	3.17 Hours	563

4.2.6.4 ANN Training Performance

The Table 4.12 shows training performances of the four ANN paradigms. The MLP network required typically 1,250,000 training iterations, LVQ required 12,224 training iterations, RBF required only 532 training iterations and PNN needed only 563 training iterations. Thus, the time required to train MLP is longer than that of RBF, PNN and LVQ. MLP got trained in approximately 21.8 hours duration with tea data vectors whereas LVQ took 6.25 hours, PNN and RBF took 2.96 and 3.17 hours respectively to complete the training. Figure 4.14 is shown with training progress in case of an MLP network and figure 4.15 is shown with training performance in case of LVQ network. Similarly, training performance of RBF and PNN can also be shown graphically. The target error (difference between present value and target value) is shown along y- axis. As training proceeds, the target error keeps reducing. If the training is reached to the exact target value, the training curve terminates on the x-axis. These figures were based on computations using a PC with processor speed of 2.4 GHz and Pentium IV processor.

4.2.7 Results and Discussions

In this part of research experiments the standard ten tea flavour terms are discriminated using an E-nose and PCA cluster analysis technique and supervised ANN techniques and hence to find out the possibility of improving existing analytical, PC based [13], profiling panel methods and Image Processing methods [14].

This result of classification of non-overlapping tea flavours using PCA cluster technique was encouraging. The E-nose sensors gathered sample data patterns as response from tea samples. The tea flavours were classified with the help of PCA, MLP, LVQ, RBF and PNN. From these result it is evident that metal oxide sensor based E-nose was capable of discriminating the flavours of tea samples which were analysed by linear and non-linear data processing ANN techniques. An accuracy of 96% was reached in the classification using RBF network compared with 93% using a PNN. It was noticed that these performances were comparable with those achieved with trained MLP 87% and LVQ 92%. As the MLP is not a good adapting network for uneven distribution of sample data and thus the classifications rate was less compared to the other paradigms. It was observed that RBF and PNN were much faster (Table 4.12) compared to MLP and LVQ nets.

It is concluded that the classification techniques suggested for tea flavour determination by using E-nose and Artificial Neural Network will be useful for Tea Industry to ascertain quality and flavour; and to standardise the same for the Tea Industries.

4.3 Spice Aroma Classification

The main reason for discriminating spice aroma by the same E-nose system was to test its robustness to other food related VOCs. Five most commonly used spices namely Chilli, garlic, ginger, onion and turmeric were chosen for experimentation and discriminated their aroma using E-nose. Flavour of spices is most critical and unique for delicious cuisine. These not only make food tasty but also have therapeutic uses and medicinal values. Spices are known mostly by their aromas.

Presently, the methods to discriminate the aroma of the spices are not adequately addressed. Possibly each consumer is the arbitrary judge with his biological nose to ascertain good aroma of the spice before buying for use. Human judgment of quality in terms of aromas of spices may not be standard and accurate due to varying perceptions of the individual at times. Perceptional variation from person to person creates difficult situation to ascertain aroma of spices to any specific standard. The aroma of spices depends upon large number of Volatile Organic Compounds (VOCs) and ratios in which they are present. The aroma has direct correlation with widely ranged properties of spices such as their seed quality, soil and irrigational conditions for growing, geographical and climatic factors, pesticide residues and appropriate ripening.

Indian Spices are most popular and well known all over the world. Proper ripening of the spices is utmost important for good aroma in them. Agricultural conditions such as a type of soil, hybridised seeds and water irrigation and ecological conditions such as climatic changes and tropical or non-tropical etc. can have strong correlation with the quality and aroma of the spice. It has recently been proved that smells may have an important role for ailment healing. Spices are well known for the health benefits for centuries.

4.3.1 Samples

The following five samples of local variety are used for the experimentations:

(i) Chilli

(ii) Garlic

(iii) Ginger

(iv) Onion

(v) Turmeric

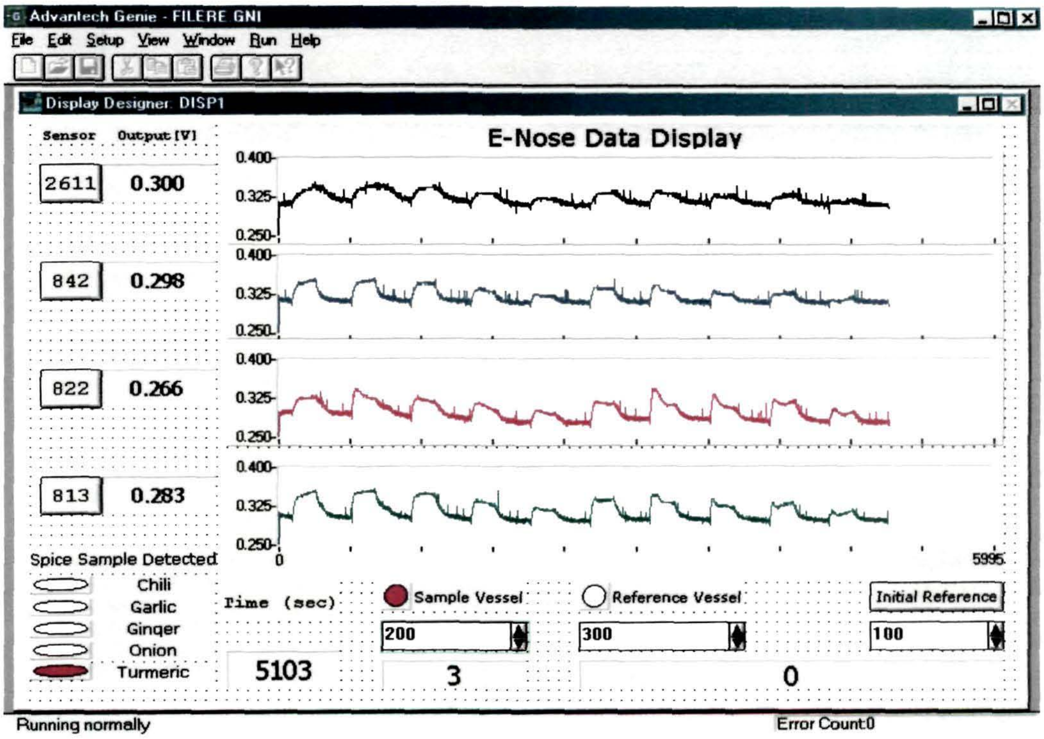
These samples are collected from the markets and farmhouses of the spices. Fifteen common local varieties of spice samples of chilli (6 varieties), garlic (2 varieties), ginger (2 varieties), onion (3 varieties) and turmeric (2 varieties) were collected and tested for their aroma with the help of E-nose.

Sample preparation includes removing skins and then chopping of spices into small pieces using a kitchen grater. The small cut pieces were crushed and then placed in sample container for testing by E-nose.

4.3.2 Experimental Method

The test procedure of E-nose system is similar to that of the tea sample testing method. The same E-nose system is used for spice testing. The samples were prepared as follows:

Fresh samples were collected from farmhouse and slightly crushed so that the more VOCs are released. The samples are placed in the stainless steel container, which, as described in case of the tea samples, is connected to the sensor chambers through flow control system, which provided constant flow of spice vapour to the sensors through connected pipelines. The block diagram of E-nose system is shown in figure 4.7.



Sensor	Output [V]
2611	0.300
842	0.298
822	0.266
813	0.283

Time (sec) 5103

● Sample Vessel 200 3

○ Reference Vessel 300 0

Initial Reference 100

Running normally

Error Count:0

Figure 4.16: A typical response of E-nose sensors to the sample of turmeric spice in an experimental process.

Primarily two methods were adopted for the aroma sniffing. Samples were first, tested at room temperature and then at elevated temperature of 100° C. In the first part of spice testing experiments, the temperature was maintained at room environment at about 25° C and relative humidity of 65% with deviation less than ±2%. In the second part of the spice experiment, the spice samples were heated to about 100 °C in an oven. This was done in the view of the fact that spices release more VOCs on heating. In this way, the samples of spices were tested experimentally to gather data set. A PC controlled the switching between reference vessel and spice sample vessel. The reference vessel is placed in an open space in room environment.

First, the spice aroma was sniffed at room temperature and then at 100° C and E-nose performed aroma-sensing process. Rest all experimental and procedural methods are similar to that of described in case of the tea sample testing.

4.3.3 Data Acquisition

GENIE environment is used for online logging and displaying results. A typical E-nose sensors' response is shown in figure 4.16. For each of the 5 spice samples, 20 data sets were acquired through experiments, thus a total of 100 data sets. Each data set consisted of 40 data vectors (each of dimensions 4 × 500) leading to a total of 4000 data vectors for all five-spice samples and 4 sensors (800 for each of five spice samples). Experimental time was 5.55 hours for each sample and thus the total experimental time was 27.75 hours for 5 spice samples. One-day gap was kept between each spice testing experiment for E-nose sensor refreshing. The spice testing continued for over 10 days.

4.3.4 Data Processing

The procedure for the data processing is similar to that of tea sample analysis. Five data matrices (each of dimensions of 4×100 for 4 sensors) for five spice samples were concatenated into a single matrix of dimensions of 4×500. This resulted in a single data matrix made of the five data matrices from five spice samples. A total of 400 such data matrices (each of dimensions 4×500) were formed from 800 vectors and the remaining 400 were used for testing. The response of the E-nose sensors for spice samples is displayed in figure 4.16 using GENIE environment and in figure 4.17 using MATLAB *plot* function.

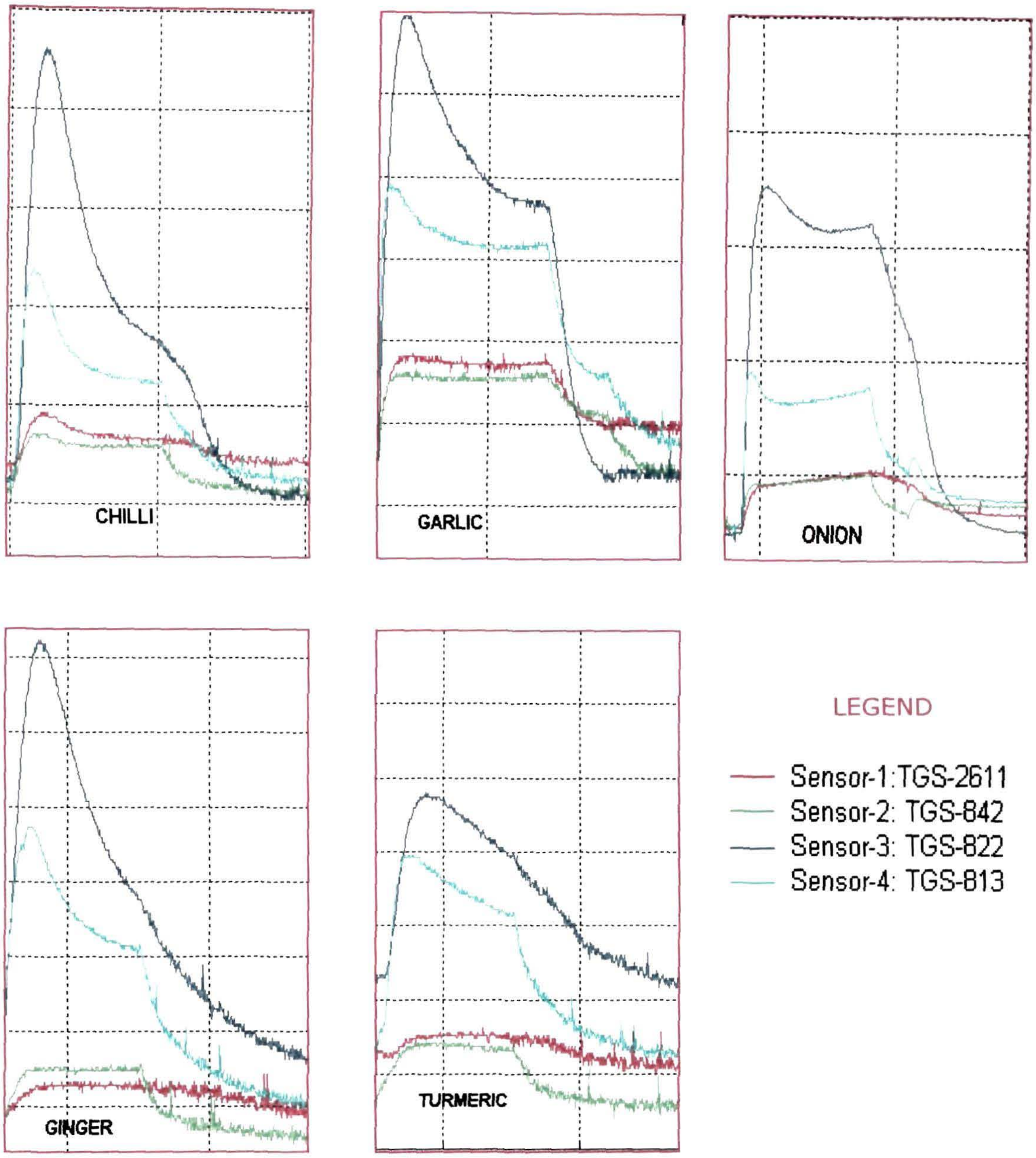


Figure 4.17: Typical responses of 4 E-nose sensors to spice flavours (Chilli, Garlic, Onion, Ginger and Turmeric). These response curves are electronic signatures of respective spice flavour.

The model used for the data pre-processing is the change in sensors' output voltage, similar to tea sample testing given by equation 4.2 in section 4.2.4. The data sets were normalized in the same way as described in tea testing procedure. The data analysis was simulated on the MATLAB 6.1 environment.

4.3.5 Principal Component Analysis

PCA results as graphical cluster displays are shown in figure 4.18. Only first three Principal Components were considered. Five aroma notes of spices appear to be distinct clusters. Table 4.13 is shown with the results of the PCA analysis on five spices samples. From data matrices (data set was a concatenated matrix of dimensions 4X500 made of response data of 5 spice samples) a covariance is calculated by using MATLAB command *cov(dataset)*. In the data set matrix for *cov(dataset)*, each row is a spice sample observation and each column represents a sensor (1-4). The algorithm for the covariance calculation is same as that of tea analysis given in section 4.2.5.1. Equation 4.3 (again in section 4.2.5.1) is used for the calculation of % variance.

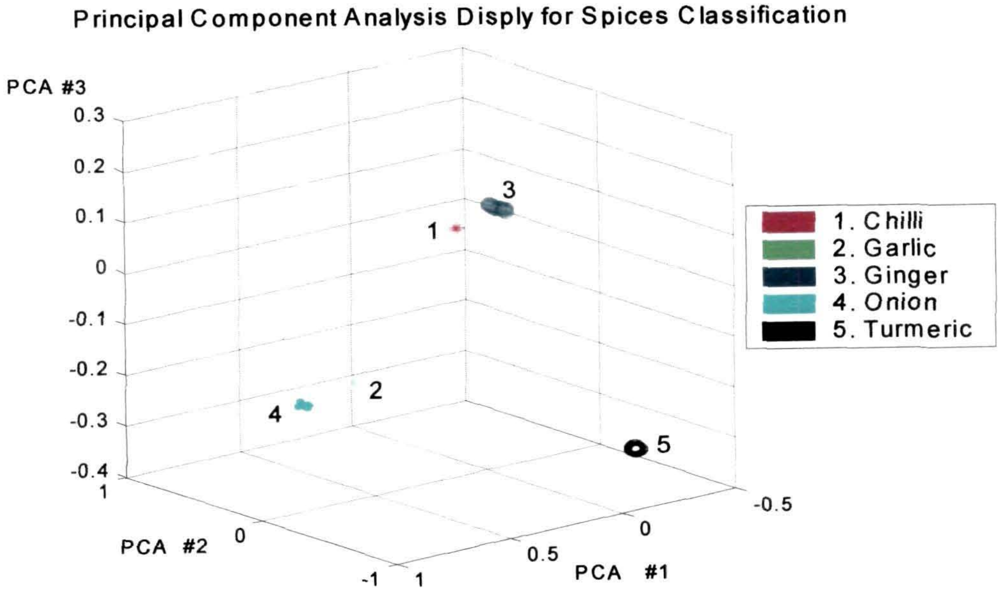


Figure 4.18: Cluster classification of five spices using PCA analysis

4.3.6 ANN Paradigm Analysis

The data sets of spice samples were analyzed using ANN classifiers, namely MLP, LVQ, PNN and RBF, same as in case of the tea samples. Training of the neural networks was performed with half of the data sets and other half of the data sets were used for testing the neural network paradigms. A total of 400 concatenated data matrices (each of dimensions 4 × 500) were used for training and remaining 400 data sets (dimensions 4 X 100 for each of five sample data, thus a total of 2000 data vectors for five spice samples) were used to test the performance of the ANN classifiers. The testing was done for individual tea flavour data. The compositions of the four ANN paradigms are similar to that of the tea samples (figure 4.6), except that there were only 4 input neurons and 5 output neurons in this case. The table 4.14 summarizes the architecture of ANN paradigms. ANN techniques were analysed in a comparative concept by using same data sets for all four ANN paradigms for training and discrimination of spice aromas.

Table 4.13: The results of PCA analysis for five spice flavour quality samples

PC	% Variance	Eigen-value	Principal Components			
			$Sensor_1$	$Sensor_2$	$Sensor_3$	$Sensor_4$
PC_1	84.9221	11.95×10^{-5}	-0.0510	0.0093	-0.9960	-0.0731
PC_2	12.3306	1.73×10^{-5}	-0.0323	-0.1840	0.0719	-0.9798
PC_3	1.4653	0.21×10^{-5}	-0.5691	-0.8044	0.0091	0.1705
PC_4	1.2821	0.18×10^{-5}	-0.8201	0.5648	0.0528	-0.0752

Table 4.14: Architecture of different Neural Networks for spice aroma discrimination.

Neural Networks	Architecture
MLP	6 hidden Neuron, 4 input Neurons, 5 output Neurons, 0.35 Learning rate with Momentum 0.42
LVQ	4 input Neurons, 5 output Neurons, Learning rate 0.011 and Conscious factor 1
PNN	4 input Neurons, 5 Neurons in output layer, Spread constant 0.8
RBF	4 input Neurons, 5 Neurons in output layer, Spread constant 0.8

4.3.6.1 Multiple Layered Perceptron

An MLP network was programmed in MATLAB environment for the classification of five spice samples. The learning rate was kept 0.35 and a momentum term was set to 0.42 with 4 input neuron, 6 hidden neuron and 5 output neurons. A total of 2000 sample vectors (400 each of five spice samples) were tested on trained MLP network. It classified 1721 samples correctly (86.05%). The results of correct classification rate are shown in Table 4.15.

MLP network completed its training after approximately 1,36,800 epochs taking about 9.6 hours of training time. The training was performed on a PENTIUM IV (2.4 GHz) processor based computer for 400 concatenated data matrices (data sets) of five spice samples and the training performances are shown in the Table 4.17

The testing part was done for individual spice flavour by using ANN paradigms one at a time for the 400 data sets for each of the five spice samples. The comparative result of test performance of ANN paradigms is shown in the Table 4.16. The MLP neuron connectionist network diagram is similar to that of figure 4.13 except that output neurons are 6 in this case.

Table 4.15: Training and testing of data sets on MLP Network and its correct classification results in % for five spice flavours

Spice Sample	No. of data vector Used for Training	No. of data vectors used for testing	No. of sample data correctly classified	Percentage of correct classification
Chilli	↑	400 (4 X 100)	346	86.50
Garlic		400 (4 X 100)	337	84.25
Ginger	400 (4x 500)	400 (4 X 100)	353	88.25
Onion		400 (4 X 100)	329	82.25
Turmeric	↓	400 (4 X 100)	356	89.00
		Total=2000 (4X100)	Total=1721	86.05%

4.3.6.2 Learning Vector Quantization

The network was configured for 4 input and 5 output neurons and a variable number of nodes in the competitive layer. Initially the network was trained with a learning rate equal to 0.011 and the conscience factor equal to 1. In the next stage, learning rate was set to 0.012, which further refined the solution. The LVQ net has correctly classify 93.35% of the response. Classification rates and training performance are shown in the Tables 4.16 and 4.17 respectively.

Table 4.16: No. of data vectors used for testing of the ANN paradigms and their correct classification results in % for spice flavours

Method	Total No. of data samples tested	No. of Correctly identified samples	% Classification
MLP	2000	1721	86.05
LVQ	2000	1867	93.35
PNN	2000	1803	90.15
RBF	2000	1899	94.95

Table: 4.17: The training performance of Artificial Neural Network paradigms for spice flavour samples.

Network	**Training Time required**	**No. of epochs to reach target error level**
MLP	9.6 Hours	1,36,800
LVQ	3.8 Hours	20527
RBF	2.6 Hours	1173
PNN	2.8 Hours	1209

4.3.6.3 Probabilistic Neural Network and Radial Basis Function

Neurons were created in net until the sum-squared error fell below the set goal of 0.00001. The spread parameter was kept high so that the radial basis neurons responded to overlapping regions of the input space, but not very high that all the neurons respond invariably. For both the networks the spread parameter was set 0.8. The PNN and RBF networks classified sample response vectors correctly to 90.15% and 94.95% respectively. Classification rates and training performance are shown in the Table 4.16 and Table 4.17 respectively.

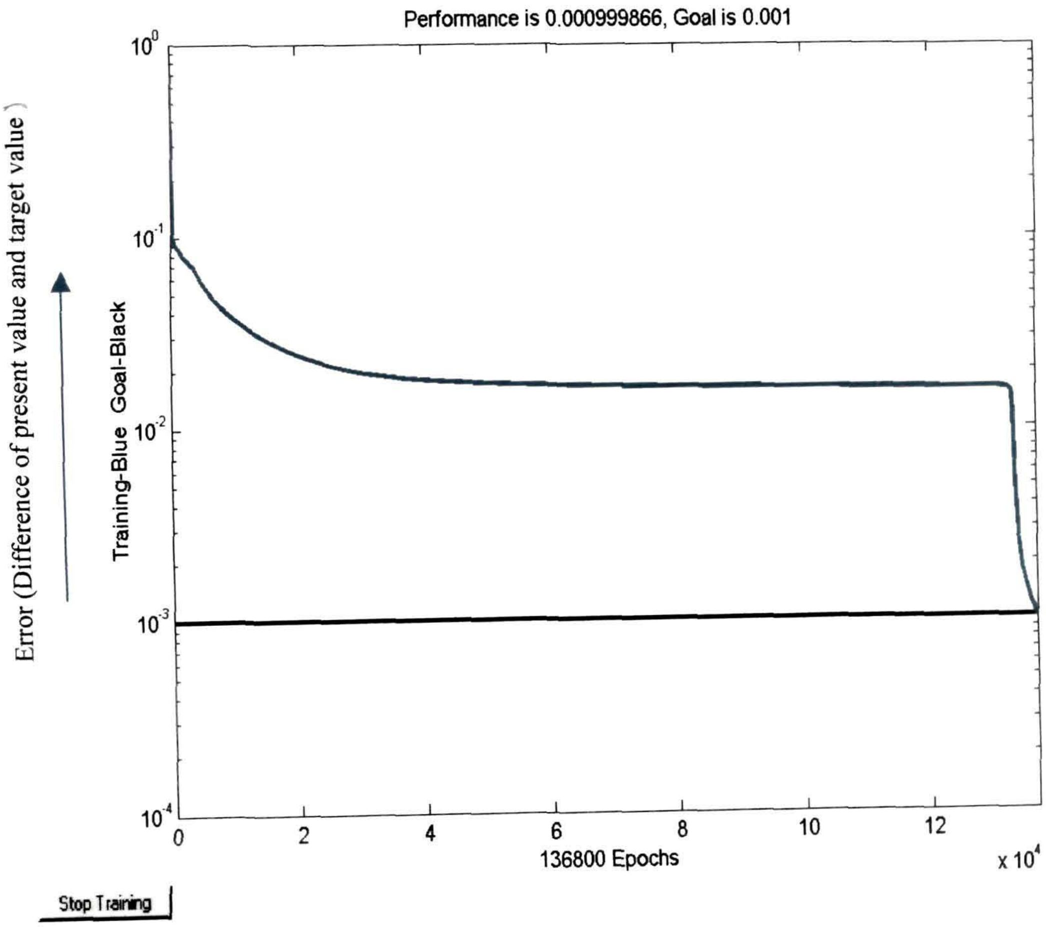


Figure 4.19: Training performance of an MLP network for spice samples. It reached the target value in 136800 epochs.

4.3.6.4 ANN Training Performance

The Table 4.17 shows training performances of the four ANN paradigms. The MLP network required typically 1,36,800 training iterations, LVQ required 20527 training iterations, RBF required only 1173 and PNN needed only 1209 training iterations. Thus, the time required to train MLP is longer than that of RBF, PNN and LVQ. MLP got trained in approximately 9.6 hours duration with tea data vectors whereas LVQ took 3.8 hours. PNN and RBF took 2.8 and 2.6 hours respectively to complete the training.

Figure 4.19 is shown with training progress in case of an MLP network. Y-axis indicated the target error. When the training performance is reached exactly to the target value, the training curve approaches to target line parallel to the x-axis. The results were based on the computations using a PC with processor speed of 2.4 GHz and Pentium IV processor.

4.3.7 Results

In this part of the experiments on spice aroma discrimination, it was established that the E-nose is a robust system, which can be generalized to discriminate a number of food aroma qualities. It becomes evident from the concept of generalized result (of tea and spices samples) of E-nose that the aroma classification and quality assurance processes in food processing industry will have new concept of automation and computerization in flavour testing.

It is also concluded that the sensor responses are linearly correlated. Since a reasonable correlation exists between different aroma classes, it becomes clear that the aroma established by PCA is consistent with different samples of the spices.

The results are enlisted in the Table 4.16. The experiments were performed to discriminate spice aromas consisting of five different spice samples using an E-nose and hence to find out the possibility of improving existing conceptual and human profiling panel methods. The E-nose sensors gathered odour data patterns as response from spice samples. The spice flavours were classified with the help of PCA, MLP, LVQ, RBF and PNN. From these result (Tables 4.14 through 4.17) it is evident that metal oxide sensor based E-nose system was capable of discriminating the flavours of spice samples analysed by linear and non-linear data processing ANN techniques. An accuracy of 95 % was reached in the classification using RBF network compared with 90 % using a PNN, 86 % using an MLP and 93 % using an LVQ. It was observed that RBF and PNN were much faster compared to MLP and LVQ nets (Table 4.17).

4.4 Optimum Sensor Selection: MLR and RMSE Analysis

In the experimentations of this research, 4 MOS sensors were used as sensing components for E-nose. It is not always true that more number of sensors can classify samples better. Sometimes a combination of fewer sensors does the better classification than the large group of sensors. Here, the problem is defined as to find out a combination of sensors, which produce the best results under given circumstance [15, 16]. A smaller number of sensors are, of course, desirable from the cost point of view and also for easier and quicker trainings on ANN analysis. Therefore, it is convenient to perform the PCA calculations with fewer sensors. A method called regression selection [15] is used to select the most significant sensors for the tea sample classification. The function makes a forward selection of variables, using the root-mean square error (RMSE) from a multiple linear regression (MLR) as the selection criterion. When the method is used on every combination of sensors, the resulting plot of RMSE versus number of sensors can be a useful visual perception [15]. The prediction error decreases with an increased number of used sensors only upto a certain number of sensors [17, 18]. The best results are achieved when combining different sensors in a group. However, the RMSE values differ depending on which sensors are used. The lowest RMSE value of sensor combination gives the most efficient group of sensors. A new PCA score plot can be seen using the sensors combination [19] and a comparison of the results of the combination. A step-by-step method [20] is presented in the following subsections to identify the best combination of the sensors to predict the tea quality. Firstly an overview of the mathematical modeling is described for easy understanding of the technique.

4.4.1 Regression Methods

Regression techniques and curve fittings attempt to find functions that describe the relationship among variables. The goal of regression is to get a model of the relationship between variables (sensors and response data). In effect, regression attempts to build mathematical models of a data set [21, 22]. MATLAB provides powerful matrix operators and functions to simplify the task of regression methods. Since the computational work for regression models for optimum number of sensor selection is done at MATLAB prompt, it will be useful to explain with the help of an example on data set executed at MATLAB prompt. The data set is a real response taken from one of the experiments on '*sour*' tea sample using E-nose as a sensing device. Data set for this example is given in Table 4.18.

4.4.1.1 Cluster Analysis

Cluster analysis, as discussed in earlier chapters, is a segmentation analysis or taxonomy analysis to partition a set of objects (sensors) into groups, or *clusters*, in such a way that the profiles of data points in the same cluster are very similar and the profiles of data points in different clusters are quite distinct [23, 24, 25]. The following steps are adopted to perform cluster analysis.

(i) Similarity or dissimilarity between every pair of data points in a data set is found by calculating the 'distance' between data points using the *pdist* function.

(ii) In this step, pairs of objects that are in close proximity are linked together using *linkage* function. The *linkage* function uses the distance information generated in the step (i) to determine the proximity (neighborhood) of objects to each other. As data points are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed.

(iii) In this step, the data points in the hierarchical tree are divided into clusters using the *cluster* function. The *cluster* function can create clusters by detecting natural groupings in the hierarchical tree or by cutting off the hierarchical tree at an arbitrary point.

The function *pdist* is used for finding the similarities between data points by calculating the distance between every pair of data set. The result of this computation is commonly known as a similarity matrix. The *pdist* function calculates the Euclidean distance between data points in a data set [23, 24].

For example if *m*-by-*n* (4 X 100, data set given in Table 4.18) data matrix *X*, which is treated as *m* (1-*by*-*n*) (1 X 100) row vectors $x_1, x_2, \ldots, x_m$, (4 rows for 4 sensors) the Euclidean distances between the vector x_r and x_s are defined as:

$$d^2_{sr} = (x_r - x_s)(x_r - x_s)' \qquad (4.4)$$

Now consider data set of the Table 4.18. Here, there are 4 rows (however, rows in Table 4.18 appear vertical) of matrix *X* corresponding to 4 sensors used. Each row has 100 data points corresponding to the responses of sensor from tea sample for 100 consecutive readings. The function *pdist(X)* at MATLAB prompt calculates the Euclidean distances as given in equation 4.5,

Table 4.18: Typical E-nose Sensors' response to 'sour' tea sample

Sl No	Sen-1	Sen-2	Sen-3	Sen-4	Sl No	Sen-1	Sen-2	Sen-3	Sen-4
1	0 3300	0 3440	0 3660	0 3490	51	0 3320	0 3470	0 3560	0 3520
2	0 3300	0 3440	0 3690	0 3520	52	0 3370	0 3490	0 3560	0 3520
3	0 3320	0 3470	0 3640	0 3520	53	0 3340	0 3470	0 3560	0 3520
4	0 3300	0 3470	0 3640	0 3520	54	0 3320	0 3490	0 3540	0 3520
5	0 3300	0 3490	0 3660	0 3520	55	0 3340	0 3490	0 3560	0 3520
6	0 3300	0 3470	0 3640	0 3520	56	0 3320	0 3490	0 3540	0 3520
7	0 3320	0 3440	0 3660	0 3520	57	0 3270	0 3490	0 3540	0 3520
8	0 3320	0 3440	0 3660	0 3520	58	0 3300	0 3490	0 3540	0 3520
9	0 3300	0 3490	0 3660	0 3520	59	0 3270	0 3490	0 3540	0 3520
10	0 3320	0 3470	0 3660	0 3490	60	0 3270	0 3470	0 3540	0 3520
11	0 3320	0 3470	0 3660	0 3520	61	0 3270	0 3490	0 3540	0 3520
12	0 3320	0 3440	0 3660	0 3520	62	0 3340	0 3490	0 3540	0 3490
13	0 3320	0 3470	0 3690	0 3490	63	0 3320	0 3470	0 3520	0 3520
14	0 3300	0 3490	0 3660	0 3540	64	0 3340	0 3470	0 3540	0 3490
15	0 3300	0 3470	0 3640	0 3520	65	0 3270	0 3490	0 3520	0 3520
16	0 3300	0 3440	0 3660	0 3490	66	0 3300	0 3470	0 3520	0 3520
17	0 3250	0 3490	0 3660	0 3520	67	0 3270	0 3470	0 3520	0 3520
18	0 3320	0 3440	0 3660	0 3520	68	0 3270	0 3490	0 3520	0 3490
19	0 3320	0 3470	0 3660	0 3520	69	0 3370	0 3490	0 3540	0 3520
20	0 3320	0 3470	0 3660	0 3490	70	0 3340	0 3490	0 3520	0 3520
21	0 3300	0 3470	0 3640	0 3520	71	0 3320	0 3470	0 3520	0 3520
22	0 3300	0 3470	0 3660	0 3490	72	0 3320	0 3490	0 3520	0 3520
23	0 3320	0 3470	0 3660	0 3520	73	0 3340	0 3520	0 3520	0 3520
24	0 3320	0 3490	0 3660	0 3520	74	0 3300	0 3470	0 3520	0 3520
25	0 3340	0 3470	0 3640	0 3520	75	0 3270	0 3490	0 3520	0 3540
26	0 3320	0 3470	0 3610	0 3520	76	0 3300	0 3520	0 3520	0 3520
27	0 3320	0 3470	0 3640	0 3520	77	0 3270	0 3490	0 3490	0 3520
28	0 3300	0 3470	0 3640	0 3520	78	0 3270	0 3490	0 3490	0 3520
29	0 3270	0 3470	0 3610	0 3520	79	0 3270	0 3490	0 3520	0 3520
30	0 3340	0 3490	0 3610	0 3520	80	0 3340	0 3490	0 3490	0 3540
31	0 3320	0 3470	0 3610	0 3520	81	0 3300	0 3490	0 3490	0 3520
32	0 3320	0 3470	0 3610	0 3520	82	0 3340	0 3490	0 3490	0 3520
33	0 3320	0 3470	0 3640	0 3520	83	0 3300	0 3490	0 3490	0 3520
34	0 3320	0 3470	0 3610	0 3490	84	0 3270	0 3490	0 3470	0 3520
35	0 3320	0 3490	0 3610	0 3520	85	0 3300	0 3490	0 3520	0 3520
36	0 3270	0 3490	0 3610	0 3520	86	0 3270	0 3490	0 3490	0 3540
37	0 3320	0 3470	0 3610	0 3520	87	0 3340	0 3490	0 3490	0 3520
38	0 3320	0 3470	0 3610	0 3520	88	0 3270	0 3490	0 3490	0 3490
39	0 3320	0 3470	0 3590	0 3490	89	0 3340	0 3490	0 3490	0 3520
40	0 3270	0 3470	0 3590	0 3490	90	0 3300	0 3490	0 3490	0 3520
41	0 3320	0 3490	0 3590	0 3520	91	0 3320	0 3540	0 3470	0 3520
42	0 3320	0 3490	0 3590	0 3520	92	0 3320	0 3490	0 3490	0 3520
43	0 3320	0 3490	0 3590	0 3520	93	0 3270	0 3490	0 3470	0 3540
44	0 3340	0 3470	0 3560	0 3520	94	0 3300	0 3490	0 3470	0 3520
45	0 3320	0 3470	0 3560	0 3520	95	0 3340	0 3490	0 3470	0 3520
46	0 3320	0 3490	0 3590	0 3520	96	0 3520	0 3520	0 3490	0 3520
47	0 3320	0 3470	0 3560	0 3520	97	0 3340	0 3490	0 3470	0 3520
48	0 3320	0 3490	0 3590	0 3520	98	0 3220	0 3490	0 3440	0 3490
49	0 3340	0 3490	0 3560	0 3520	99	0 3320	0 3490	0 3470	0 3520
50	0 3320	0 3490	0 3560	0 3520	100	0 3250	0 3520	0 3470	0 3520

$$Y = [0.1742 \quad 0.2685 \quad 0.2092 \quad 0.1185 \quad 0.0404 \quad 0.0876] \qquad (4.5)$$

The *pdist* function returns this distance information in a vector, Y, where each element contains the distance between a pair of objects. First value is between sensor-1 and Sensor-2, second between sensor-1 and sensor-3, third between sensor-1 and sensor -4, fourth between sensor-2 and sensor-3 and so on. To make it easier to see the relationship between the distance information generated by *pdist* and the sensors in the original data set, the distance vector Y can be reformatted into a matrix using the *squareform(Y)* function. The resultant distance between each possible pair is shown in Table 4.19

Table 4:19 Euclidean distance between each pair of sensors

Sensors	Sen-1	Sen-2	Sen-3	Sen-4
Sen-1	0	0.1742	0.2685	0.2092
Sen-2	0.1742	0	0.1185	0.0404
Sen-3	0.2685	0.1185	0	0.0876
Sen-4	0.2092	0.0404	0.0876	0

Once the distance between sensors in the data set has been computed, it can be determined that which objects in the data set should be grouped together into clusters, using the *linkage* function. The linkage function takes the distance information generated by *pdist* and links pairs of objects that are close together into binary clusters (clusters made up of two objects). The *linkage* function then links these newly formed clusters to other objects to create bigger clusters until all the objects in the original data set are linked together in a hierarchical tree.

The *linkage(Y)* function gives output (say *Z*) as shown in the Table 4.20. In the output, each row identifies a link between a pair of sensors. The least Euclidean distance appears first. The first two columns identify the sensors that have been linked, that is, sensor-2 and sensor-4 and so on. The third column contains the Euclidean distance between these sensors. For the sample data set, the linkage function begins by grouping together sensor-2 and sensor-4, which have the closest proximity (distance value = 0.0404). The linkage function continues by grouping sensor-5 and sensor-3 and so on.

The second and third row indicates that the *linkage* function grouped together sensor-5 and sensor-3 and sensor-1 and sensor-6. It can be remembered that the experiment was done with only 4 sensors. Sensor-5 and sensor-6 are the newly formed binary cluster sensors (representing combination of sensor-2 and sensor-4, and sensor-5 and sensor-3 respectively). When the *linkage* function groups two sensors together into a new cluster, it must assign the cluster a unique index value, starting with the value *m*+1, where *m* is the number of sensors in the original data set (here, *m* = 4). Values 1 through 4 are already used by the original data set and thus 5 and 6 for new sensors. Sensor-5 is the index for the cluster predicted by using sensor- 2 and sensor-4. As the final cluster, the *linkage* function grouped sensor-5 and sensor-3 to predict sensor-6. The Fig. 4.20 graphically illustrates the way function *dendrogram* groups the objects into a hierarchy of clusters [25].

Table 4:20 Euclidean distance between each pair of sensors used in experiments and predicted statistically.

Pair of	Sensors	Eucl. Distance between pair
2.0000	4.0000	0.0404
5.0000	3.0000	0.0876
1.0000	6.0000	0.1742

In the Fig. 4.20, the numbers along x-axis represent the indices of the sensors in the original data set. The links between objects are represented as upside down U-shaped lines. The height of the U indicates the Euclidean distance between the objects. For example, the link representing the cluster containing objects 2 and 4 has a height of 0.04 approximately.

One way to measure the validity of the cluster information generated by the *linkage* function is to compare it with the original proximity data generated by the *pdist* function. If the clustering is valid, the linking of objects in the cluster tree should have a strong correlation with the distances between objects in the distance vector [26, 27]. The *cophenet* function compares these two sets of values and computes their correlation, returning a value called the *cophenetic correlation coefficient*. The closer is value of the *cophenetic correlation coefficient* to *1*, the better clustering solution. The following relation returns cophenetic *correlation coefficients*:

$$C = \text{cophenet}(Z,Y) \tag{4.6}$$

Where Z (Table 4.20) is *linkage* of Y (*pdist,* equation 4.5). On execution of the equation 4.6 at MATLAB prompt, it returns the value as

$$C = 0.9255 \tag{4.7}$$

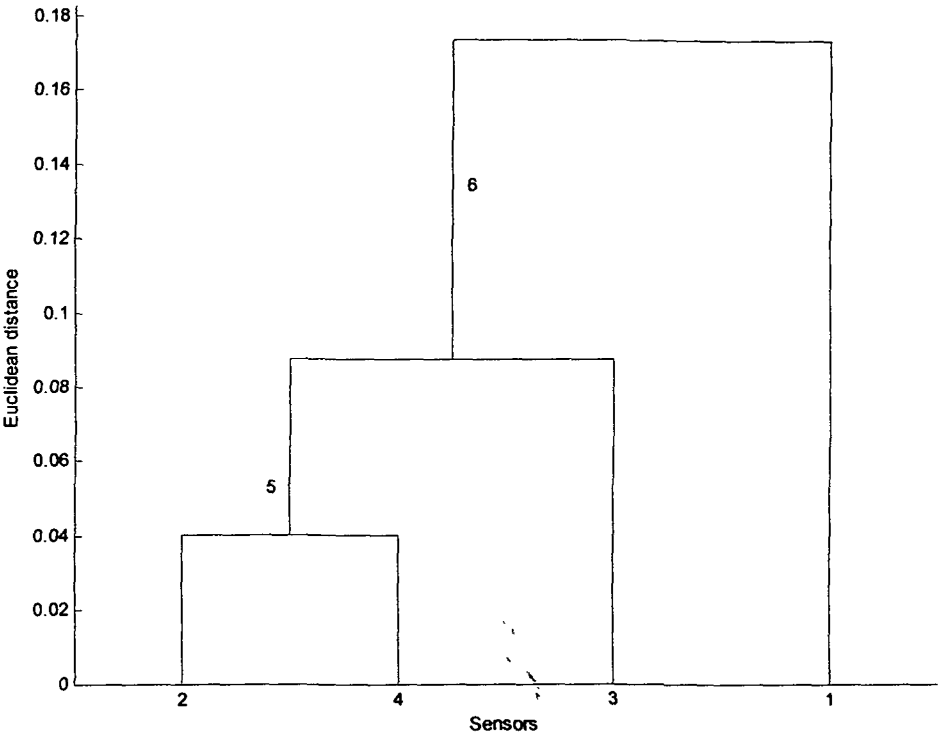


Figure 4.20: Clustering of 4 sensors in Dendrogram. Sensor-2 and sensor-4 are clustered to predict sensor-5 and then sensor-5 and sensor-3 are clustered to predict sensor-6. Sensor-6 is closest to sensor-1 in Euclidean distance.

From these results it was observed that sensor-2 and sensor-4 were strongly correlated indicating that they had highly overlapping response. While analyzing optimum sensor selection, only one sensor is selected from each such group. In this case sensor-2 and sensor-4 have almost similar response and thus any one of the two sensors would respond in a similar way. However, sensor-3 is in the neighbourhood of combination of sensor-2 and sensor-4 and sensor-1 farthest away in the group. From this analysis it becomes clear that a group of sensors 1, 2 and 3 or 2, 3 and 4 would be almost as efficient as a group of all four sensors. However, it is to be further analysed that which group out of two is more efficient.

4.4.1.2 Linear Regression Models

Linear regression models represent the relationship between a continuous response variables and predictor sensors. The common form of this model is given by

$$y = X\beta + \varepsilon \tag{4.8}$$

Where:

y is an n-by-1 vector of observations of the response variable.

X is the n-by-p design matrix determined by the predictor sensors.

β is a p-by-1 vector of parameters.

ε is an n-by-1 vector of random disturbances (error), independent of each other and usually having a normal distribution.

MATLAB provides general form of the linear model to solve a regression and Analysis of Variance (ANOVA, acronym for **AN**alysis **O**f **VA**riance) problems for optimum sensor selections, the columns of X are response values from E-nose. For analysis of covariance (ANOCOVA) models, X contains values of a continuous predictor sensors and codes for a categorical predictor sensor.

The purpose of one-way ANOVA is to find out whether data from 4 sensors have a common mean. That is, to determine whether the sensors are actually different in the measured characteristic. One-way ANOVA is a special case of the linear model. The one-way ANOVA model is given by

$$y_{ij} = \alpha_{.j} + \varepsilon_{ij} \tag{4.9}$$

Where:

y_{ij} is a matrix of observations in which each column represents a different group of sensors.

$\alpha_{.j}$ is a matrix whose columns are the group means. (The "dot j" notation means that α applies to all rows of the jth column. That is, the value α_{ij} is the same for all i.)

ε_{ij} is a matrix of random disturbances.

Now, data set of *sour* tea sample given in Table 4.18 is used to perform ANOVA analysis. This will examine whether sensors' response have any similarity. The function *anova1(Y)* does the computation at MATLAB prompt on data set *Y*. The results of ANOVA analysis are shown in Fig. 4.21.

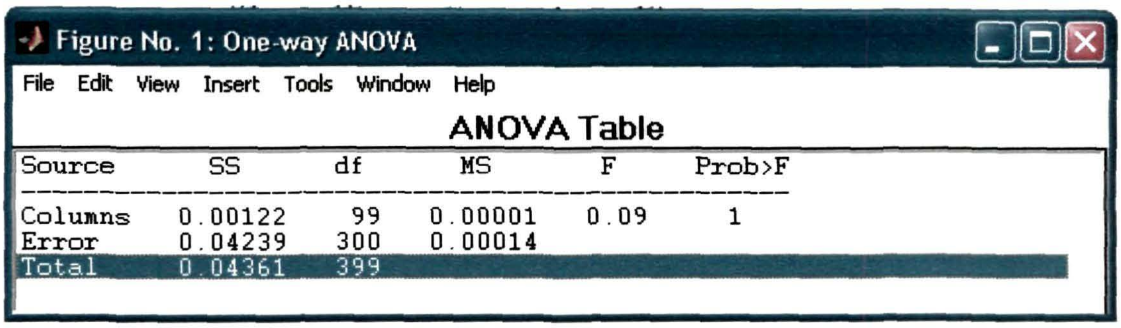


Figure No. 1: One-way ANOVA
File Edit View Insert Tools Window Help

ANOVA Table

```
Source       SS        df       MS         F      Prob>F
--------------------------------------------------------
Columns   0.00122     99    0.00001     0.09       1
Error     0.04239    300    0.00014
Total     0.04361    399
```

Figure 4.21: Results of one-way analysis of variance in MATLAB window.

The standard ANOVA Table 4.18 has columns for the sums of squares (SS), degrees of Freedom (df), mean squares (SS/df), F statistic, and p-value. The *F* value is used to do a hypothesis test to find out if the response is same from all sensors. Function *anova1* returns the p-value from this hypothesis test. In this case the p-value is 1 (ideally it should be very small for entirely different response), which is a strong indication that the response from the different sensors are somewhat similar. This means that there exists a redundancy among the sensors or in other words some of the sensors (for example sensor-2 and sensor-4 have strong correlation as predicted in cluster analysis) have overlapping characteristics.

4.4.1.2 Multiple Linear Regressions

The purpose of multiple linear regressions is to establish a quantitative relationship between a group of sensors (the columns of *X*) and a response, *y* [30, 31]. This relationship is useful for:

(i) Understanding that which sensor has the greatest effect.

(ii) Knowing the direction of the effect (i.e., increasing number of *sensors, X* increases or decreases *response, y*).

(iii) Using the model to predict future values of the response when only the current sensors are known.

The solution of equation (4.8) is a vector, *b*, which estimates the unknown vector of parameters, b. The least squares solution is:

$$b = (X^T X)^{-1} X^T y \tag{4.10}$$

An estimate of a parameter or a prediction from a model is achieved from the projection hat matrix H,

$$H = (X^T X)^{-1} X^T \tag{4.11}$$

The residuals are the difference between the observed and predicted *y* values.

$$r = (I - H))y \tag{4.12}$$

The residuals are useful for detecting failures in the model assumptions, since they correspond to the errors, ε, in the model equation (4.8). By assumption, each of these errors has independent normal distributions with mean zero and a constant variance.

4.4.1.3 Stepwise Linear Regressions

Stepwise regression is a technique for choosing the number of sensors in an experiment to include in a multiple regression model. Forward stepwise regression starts with no model terms [23]. At each step it adds the most statistically significant sensor (the one with the highest F statistic or lowest p-value) until there are none left. Backward stepwise regression starts with all the sensors in the model and removes the least significant sensor until all the remaining sensors are statistically significant. It is also possible to start with a subset of all the sensors and then add significant terms or remove insignificant terms. An important assumption behind the method is that some of the sensors in a multiple regression do not have an important explanatory effect on the response. If this assumption is true, then it is a convenient simplification to keep only the statistically significant sensors in the model. One common problem in multiple regression analysis is multicollinearity of the number of sensors. The sensors may be as correlated with each other as they are with the response. If this is the case, the presence of one sensor in the model may mask the effect of other. The following computation for the E-nose sensor response data of Table 4.18 gives MLR analysis.

Table 4:21 Regression analysis results

Sensors	R^2	F statistic	p-value
Sensor-1	0.4193	70.7579	0.0000
Sensor-2	0.0002	0.0164	0.8984
Sensor-3	0.0010	1.9668	0.0000
Sensor-4	0.0305	3.0789	0.0824

The function *regress(y, X),* where y is response and X sensors, gives the statistical analysis as shown in table 4.21. The syntax for the regress function is:

```
[b1,bint1,r1,rint1,stats1] = regress(sour_data1(1,:)',x);
[b2,bint2,r2,rint2,stats2] = regress(sour_data1(2,:)',x);
[b3,bint3,r3,rint3,stats3] = regress(sour_data1(3,:)',x);
[b4,bint4,r4,rint4,stats4] = regress(sour_data1(4,:)',x);
```

where:
b = confidence intervals
r = residual vector
bint = confidence intervals for b
rint = confidence intervals for r
1,.....,4 = sensor index used.
stats = contains R^2, F stats and p value

The elements of the vector *stats* contain the R^2 statistic along with the F statistic (for the hypothesis test that all the regression coefficients are zero) and p values for the regression associated with F statistic. The R^2 value is the ratio of the regression sum of squares to the total sum of squares thus, percentage indication of variability in sensors' response. For example R^2 for sensor-1 (Table 4.21) is 0.4193, indicating the model accounts for only 41.93% of the variability in the observations. This shows that sensor-1 has contributed quite well in the observation. The F statistic of 70.7579 and its p-value of 0.000 (ideal) indicate that it is highly significant. Sensor-2 has contributed poorly, which confirms the results of Dendrogram and cluster analysis.

4.4.1.4 Root Mean Square Error Analysis

Root Mean Square Error (RMSE) is a significant statistics for neural networks and regression models. Once the network weights and biases have been initialized, the network can be trained for function approximation (nonlinear regression), pattern association, or pattern classification. The training process requires a set of examples of proper network behavior - network inputs and target outputs. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function. This performance function for Feed-forward networks is normally Mean Square Error (MSE) - the average squared error between the network outputs and the target outputs [33, 34].

Performance function MSE is a network performance function. It measures the network's performance according to the mean of squared errors. The syntax for this function is:

Table 4:22 Root mean square error values for sour tea sample.

Sensors	RMSE
Sensor-1	1.3×10^{-3}
Sensor-2	3.4×10^{-3}
Sensor-3	1.5×10^{-3}
Sensor-4	1.1×10^{-3}

```
perf = mse(e, x, pp)
perf = mse(e, net, pp)
info = mse(code)
RMSE = sqrt(perf)
```

Here *mse(e, x, pp)* functions takes from one of the following three arguments,

e - matrix or cell array of error vector(s) from residuals of regression analysis.

x - vector of all weight and bias values (ignored in this case).

pp - performance parameters (ignored in this case).

mse(code)- returns useful information for each code string.

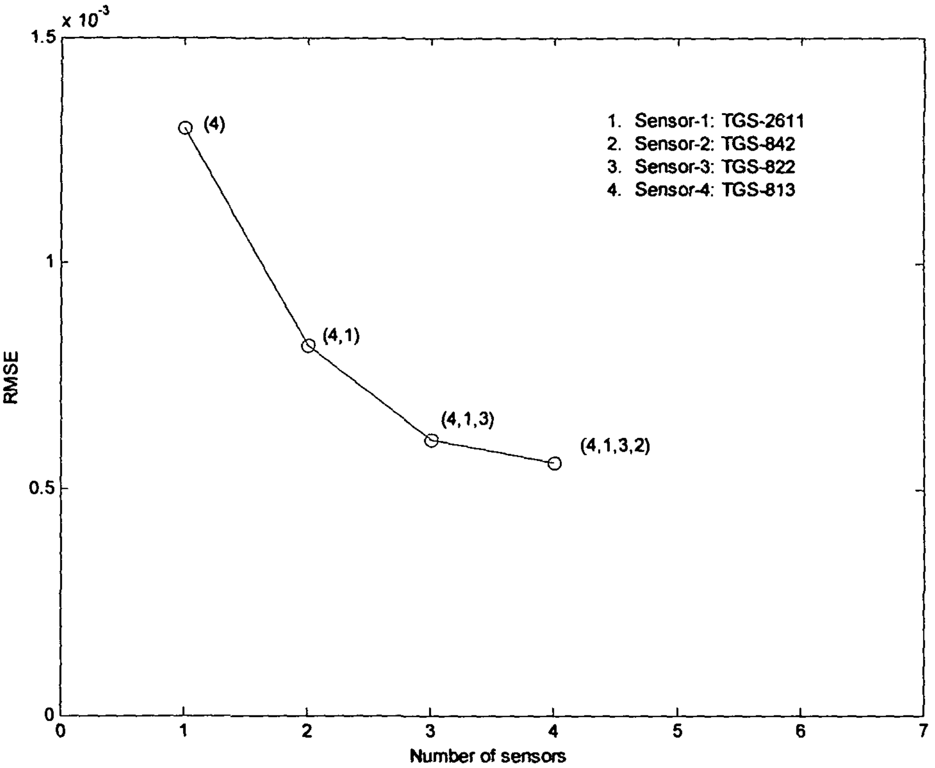


Figure 4.22: Plot of the total RMSE value of the given combination of sensors versus number of sensors in the combination using the regression selection method. Bracketed numbers at every node shows indices of the combination of different sensors.

The function *mse(e, x, pp)* returns the mean squared error. The model of *mse* is applied on four sensors' responses and root mean square error is calculated from *mse* results. RMSE analysis is indicative of sensors' individual performance based on regression residuals. The Table 4.22 shows that error is minimum in case of sensor-4 and maximum in case of sensor-2. From this analysis it becomes clear that if at all a sensor is to be excluded from the selected group of four sensors, then in that case it must be sensor-2 as it has appeared least significant in all three analysis (Dendrogram, regression and RMSE).

Fig. 4.22 shows the RMSE analysis result of different combinations of the four sensors. As an individual performance, sensor-4 is emerged with minimum RMSE

(Table 4.22). This point is taken as first node in Fig. 4.22. Then a combination of responses of sour tea sample using sensor-4 and sensor-1 is analysed. The resultant RMSE is reduced (second node in Fig. 4.22). Similarly, the other combinations are analysed. At node 4 when sensor-2 added, there is very little change in RMSE, indicating that sensor-2 is less significant.

4.4.2 PCA Analysis for sensor selection criterion

As described in earlier sections, PCA is used for cluster separation and mostly a visual graphical aid to understand the cluster existence and the distinctness of the clusters in 3-D scatter diagrams [23, 24, 29, 32, 33]. Therefore, it is relevant to perform the PCA calculations with combinations of fewer sensors for tea flavour discrimination. Using this method on every combination of sensor, the resulting plot of PCA scores of number of sensors in a combination is shown in Fig. 4.23. The prediction error decreases with an increased amount of used sensors upto a group of 4 sensors (Fig 4.22). Sample for this analysis was selected from ten non-overlapping tea flavours as described earlier.

Figure 4.23 (a) is plotted using all four available sensors. The other figures in (b), (c) and (d) are plotted with a sensor removed at a time from the group of four sensors in successive experiments. These plots show a similarity to certain degree but not much encouraging, as the clusters are more distinct only when all four sensors are used at a time as seen in Fig. 4.23 (a). The scores for the four different test sets of sensor combinations projected on the PCs from the PCA is an apparent similarity among the groups of the sensors. This shows that only a few sensors are needed to identify different tea flavours. The results presented here show PCA is efficient to ascertain the quality of the tea while keeping an optimum number of sensors in an experiment.

4.4.3 Results

In this part of experiments a method is analysed for using an optimum number of sensors, which can almost produce the same results as in case of whole sensor array.

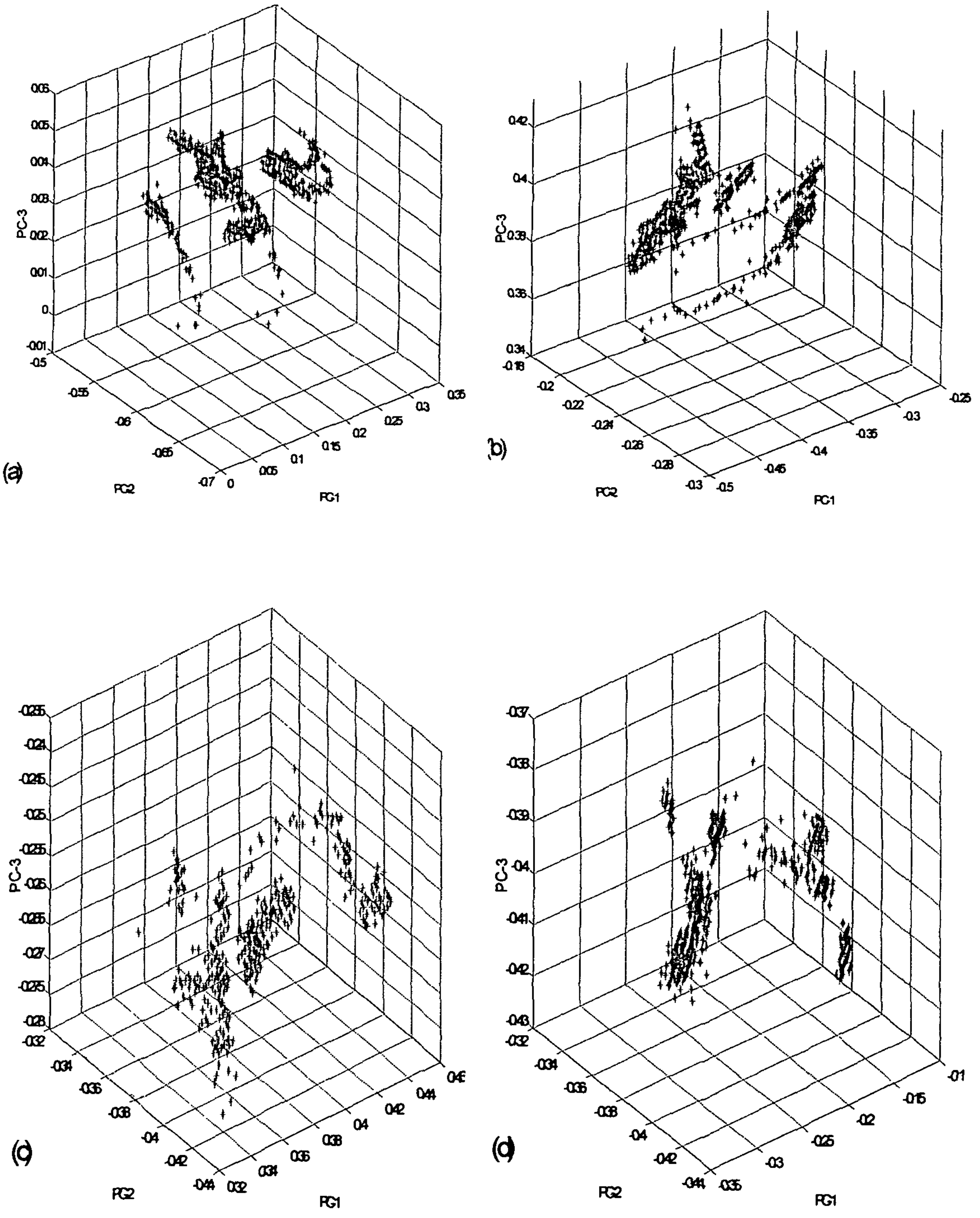


Figure 4.23: PCA score plot using four MOS based TGS sensors' response from E-nose on ten non-overlapping tea flavours. The first three principal components are used for these plots. These graphs are plotted in MATLAB using different combinations of three sensors at a time out of total four sensors available at the time of experiments. The combinations are (a) TGS-2611, TGS-842, TGS-822 and TGS-813 (b) TGS-2611, TGS-822 and TGS-813 (c) TGS-2611, TGS-842, and TGS-813 (d) TGS-2611, TGS-842 and TGS-822

Reducing number of sensors can give results slightly less accurate but it is desirable from the economical point of view. The cluster analysis using Dendrogram [34, 35] and Euclidean distance principles has indicated that sensor-2 and sensor-4 are strongly correlated. This correlation suggests that these two sensors have overlapping feature selection ability. Sensor-2 appears to be less efficient on the score of RMSE. However, sensor-3, and sensor-1 are different and produced more desirable results on RMSE analysis. Based on the cluster analysis two possible groups were identified for better trade offs between results and cost or number of sensors. The first group consists of sensors-1, sensor-2 and sensor-3 and the second group consist of sensor-2, sensor-3 and sensor-4. An analysis using regressions models was carried out on four sensors. In case of sensor-1 the F statistic of 70.7579 and its p-value of 0.000 (ideal) indicate that it is more significant. Sensor-2 has scored bad confirming the results of Dendrogram and cluster analysis. The results of RMSE analysis shows that sensor-4 had minimum RMSE. When a combination of different sensor is used it results in reduced RMSE. The resultant RMSE is plotted in Fig. 4.22. At node 4 in Fig 4.22, sensor-2 is added to the combination, however that had very little change in RMSE, which indicated that sensor-2 was again non-significant.

This method is more suitable for selection of optimum number of sensors in a large group of sensor array. In this research the number of sensors were limited, however, the method is general and can be used for better results for large arrays.

4.5 Conclusion

In this chapter a complete analysis of tea and spice flavour discrimination is carried out. Data preprocessing and data analyzing using different techniques was performed. In the first part of experiments a faulty manufactured samples of tea were used for flavour discrimination. These tea samples had different flavour notes due to non-compliance of temperature, humidity and timing requirements in tea processing. In the second part of experiments non-overlapping samples of tea were used for flavour discrimination. These samples were of different tea flavours due to different tea species. LabVIEW© (National Instruments Inc.) and GENIE (ADVANTECH®, American Advantech Corporation) were used to acquire and display the results. The model used for the data preprocessing was static change in sensor resistance in first part and output voltage in the second part. Four different cluster classification methods, PCA, SOM, FCM and Combination of FCM, SOM and 3D Scatter diagrams were used for data clustering for overlapping tea samples. The results of PCA, using the normalized data vectors were encouraging, which accounted for 99.8556% of the variance in case of overlapping and 99.063 % in case of non-overlapping tea samples. Cluster centers for different tea flavour classes were obtained using FCM algorithm for overlapping tea samples. An SOM network was created and trained with the data set. Subsequently, samples were associated with one of the output neurons. A new data clustering method was analysed for E-nose response data. By combining the 3D scatter plot, FCM and SOM network produced clusters of overlapping tea samples. Using 3D scatter plots, the most probable clusters were formed. SOM network had learned about the data distributions and SOM nodes resulted in probable clusters' centers. These three un-supervised data processing algorithms, independently and effectively generated the same results.

The data sets were analyzed using four supervised ANN classifiers, namely the MLP, LVQ, PNN and RBF paradigms. Training of the neural networks was performed with 50% of the data sets and other 50% of the data sets were used for testing the neural network paradigms. The time required to train MLP is longer than that of RBF, PNN and LVQ. A *t*-test was performed to check whether the RBF, PNN and LVQ algorithms were performing better than the MLP in terms of the total number of patterns correctly classified.

Lastly a criterion for optimum number of sensors' selection from an array was analysed. The aim of this technique was based on the fact that sometimes a combination of fewer significant sensors does the better classification compared to the large group of sensors. A method called 'regression model' using the root-mean square error (RMSE) from a multiple linear regression (MLR) as the selection criterion was analysed and a sub group of sensors was identified. A new PCA scores were plotted using the responses from a combination of different sensors.

References

[1] K. R. Kashwan, M. Bhuyan, *Robust Electronic-Nose System with Temperature and Humidity Drift Compensation for Tea and Spice Flavour Discrimination,* IEEE Conference, AsiaSense-2005, Asian Conference on Sensors

And International Conference on New Techniques in Pharmaceutical and Biomedical Research, September 5-7, 2005, Kuala Lumpur, Malaysia.

[2] Ritaban Dutta, K. R. Kashwan, M. Bhuyan, E. L. Hines, J. W. Gardner *Electronic Nose based tea quality standardization,* Neural Networks Special issue, Volume 16, Issue 5-6 (2003) 847-853, Elsevier Science Ltd, UK.

[3] M Bhuyan and S Borah, *Use of Electronic Nose in Tea Industry,* EAIT, 2001, IIT, Kharagpore.

[4] J.W. Gardner, *Detection of vapours and odours from a multisensor array using pattern recognition. Part 1. Principal components and cluster analyses,* Sensors and Actuators B 4 (1991) 108–116.

[5] http://www.mathwork.com.

[6] Ritaban Dutta, E. L. Hines, J. W. Gardner, K. R. Kashwan, M. Bhuyan. *Determination of tea quality by using a Neural Network based Electronic Nose,* International Joint Conference on Neural Network (IJCNN 2003), Portland, Oregon, USA. Conference sponsored by IEEE Neural Network Society and International Neural network Society, July 20-24 Vol. 1, pp. 404-409, 2003

[7] Ritaban Dutta, E. L. Hines, J. W. Gardner K. R. Kashwan, M. Bhuyan. *Tea quality prediction using a tin oxide based electronic nose*: Elsevier Science Journal: Sensors and Actuators B 94(2003) Volume 94, Issue 2 (1 September 2003) Pages: 228-237. Elsevier Science Ltd. Oxford, UK

[8] J.S.R. Jang, C.T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Upper Prentice-Hall, Saddle River, NJ, 1997, pp. 423–433.

[9] T. Kohonen, *Self-Organization and Associative Memory,* third ed., Springer, Heidelberg, 1989, Chapter 7, pp. 185–209.

[10] J.W. Gardner, K.C. Persaud, *Electronic Noses and Olfaction 2000*, Institute of Physics Publishing (IoP), Bristol, 2000.

[11] K.Z. Mao, RBF neural network center selection based on Fisher ration class separability measure, IEEE Trans. Neural Network 13 (5) (2002) 1211–1217.

[12] R. Dutta, E.L. Hines, J.W. Gardner, P. Boilot, *Bacteria classification using Cyranose 320 electronic nose*, BioMed. Eng. Online 1 (4) (2002).

[13] M Bhuyan, *An Integrated PC based Tea Process Monitoring and control System,* Ph.D. Thesis, Guwahati University, October 1997.

[14] S Borah and M Bhuyan *Non-destructive testing of tea fermentation using image processing* Insight 45, 2003.

[15] J.H. Zar, *Biostatistical Analysis,* 4th edn., Prentice-Hall, Englewood Cliffs, NJ, 1999.

[16] W.P. Carey, B.R. Kowalski, in: R.F. Taylor, J.S. Schultz Eds., *Handbook Chemical and Biological Sensors,* IOP, Philadelphia, 1996, pp. 287–315.

[17] K.C. Persaud, P.J. Travers, in: E. Kress-Rogers Ed., *Handbook of Biosensors and Electronic Noses: Medicines, Food and the Environment,* CRC Press, Boca Raton, FL, 1997, pp. 563–592.

[18] J.W. Gardner, P.N. Bartlett, in: J.W. Gardner, P.N. Bartlett Eds., *Sensors and Sensory Systems for Electronic Noses* vol. 212 Kluwer Academic Publishing, 1992.

[19] J.W. Gardner, P.N. Bartlett, in: P. Moseley, J. Norris, D. Williams (Eds.), *Techniques and Mechanisms in Gas Sensing,* Adam Hilger, Bristol, 1991, pp. 347–380.

[20] B.M. Wise, N.B. Gallagher, PLS Toolbox 2.0, *Eigenvector Technologies,* Washington, 1996.

[21] O. Weidemann, M. Hermann, G. Steinhoff, H. Wingbrant, A. Lloyd Spetz, M. Stutzmann, and M. Eickhoff, *"Influence of surface oxides on hydrogen-sensitive Pd:GaN Schottky diodes," Appl. Phys. Lett.*, vol. 83, no. 4, p. 773, 2003.

[22] Y. Gurbuz, W. P. Kang, J. L. Davidson, and D. V. Kerns, *"Current conduction mechanism and gas adsorption effect on device parameters of the Pt/SnO$_x$ /diamond gas sensor," IEEE Trans. Electron Devices,* vol. 46, no. 5, p. 914, May 1999.

[23] S. Wold, K. Esbensen, and P. Geladi, "Principle component analysis," *Chem. Intell. Lab. Syst.*, vol. 2, p. 37, 1987.

[24] T. Eklöv, P. Mårtensson, and I. Lundström, *"Selection of variables for interpreting multivariate gas sensor data," Anal. Chim. Acta.*, vol. 381, pp. 221–232, 1999.

[25] T. Artursson, T. Eklöv, I. Lundström, P. Mårtensson, M. Sjöström, and M. Holmberg, *"Drift correction for gas sensors using multivariate methods," J. Chemometr.*, vol. 14, 2000.

[26] R. E. Shaffer, S. L. Rose-Pehsson and R. A. McGill, *"Improved probabilistic neural network algorithm for chemical sensor array pattern recognition," Anal. Chem.*, vol. 71, no. 19, Oct. 1999.

[27] J. B. Doleman, M. C. Lonergan, E. J. Severin, T. P.Vaid, and N. S. Lewis, *"Quantitative study of the resolving power of arrays of carbon blackpolymer composites in various vapor-sensing tasks," Anal. Chem.*, vol. 70, no. 19, pp. 4177–4190, Oct. 1998.

[28] A. J. Ricco, R. M. Crooks, and G. C. Osbourn, *"Surface acoustic wave chemical sensor arrays: New chemically sensitive interfaces combined with novel cluster analysis to detect volatile organic compounds and mixtures," Acc. Chem. Res.*, vol. 31, no. 5, pp. 289–296, 1998.

[29] W. P. Carey, K. R. Beebe, and B. R.Kowalski, *"Multicomponent analysis using an array of piezoelectric crystal sensors," Anal. Chem.*, vol. 59, no. 11, pp. 1529–1534, 1987.

[30] T. A. Dickinson, K. L. Michael, J. S. Kauer, and D. R. Walt, *"Convergent, self-encoded bead sensor arrays in the design of an artificial nose," Anal. Chem.*, vol. 71, no. 11, June 1999.

[31] J. W. Grate, B. M. Wise, and M. H. Abraham, *"Method for unknown vapor characterization and classification using a multivariate sorption detector. Initial derivation and modeling based on polymer-coated acoustic wave sensor arrays and linear solvation energy relationships," Anal. Chem.*, vol. 71, no. 20, pp. 4544–4553, Oct. 1999.

[32] J. Waldemark, T. Roppel, D. Wilson, K. Dunman, M. L. Padgett, and T. Lindblad, *"Neural networks and PCA for determining region of interest in sensory data pre-processing,"* in *VI-DYNN*, Stockholm, Sweden, June 1998, pp. 22–26.

[33] J. W. Gardner, E. L. Hines, and M. Wilkinson, *"Application of artificial neural networks to an electronic olfactory system," Meas. Sci. Technol.*, vol. 1, pp. 446–451, May 1990.

[34] P. Keller, *"Overview of electronic nose algorithms,"* presented at the *Conf. Proc. IJCNN*, Washington, DC, July 1999.

[35] F. M. Ham and I. Kostanic, *Principles of Neurocomputing for Scienceand Engineering*. New York: McGraw-Hill, 2001, pp. 106–135.

§§

CHAPTER 5

DRIFT PARAMETER DETERMINATION

Chapter 5

Drift Parameter Determination

An experimental method for determination of the drift coefficients in the E-nose system due to temperature, humidity and pressure variation in the sample gas has been described in this chapter. Since sensor drift can invalidate some of the classification models [1], these may require correction or retraining. Without correction for sensor drift, databank for sample recognition may be limited in size because of the impracticability of frequent retraining of E-nose system. Drift is a serious problem in sensors that often prevents reliable sample identification over long timescales. Many efforts are focused on statistically removing the components of the data that contained the most drift in E-nose system. However, at some point, even successful statistical methods fail, requiring retraining of the system [2].

Drift correction is a more time-intensive method but may be the only robust method for determining precise information regarding the degradation of the classification model in the presence of drift due to change in physical parameters. Drift coefficients are established by conducting rigorous experiments on the samples response to E-nose system under the operating conditions of a large range of temperature, humidity and pressure variations. This will result calibration of the sensors for the specific combination of sensor-sample. The temperature and humidity coefficients are determined for the specific tea sample and pressure coefficients for the kerosene gas (vapour) sample at high pressure. The same method can also be applied for other specific combinations of sample-sensor for the E-nose calibration.

5.1 The Nature of Drifts in E-nose Sensors

Drift is a dynamical process caused by physical changes in the sensors and the chemical backgrounds, which gives an unstable signal over the time. The drift could be both reversible and irreversible. The causes of the drift include variation in the surrounding in which the sensors and samples are placed [3]. These factors are normally temperature, humidity and vapour pressure (in case of leakage detection). Another cause of drift is measurement history, called memory effects, which means that the measurements at time *t* is highly influenced by measurements at time *t-n*. Under memory effects, the same gas mixture may not give a well-defined pattern [4]. The aging of the sensor is also a considerable cause of the drift. If sensor signals are not corrected or compensated then pattern recognition models will need a continuous retraining and re-gathering the data response [5].

MOS sensors for E-nose are operated within predefined temperature range due to inbuilt heating mechanism subjected to power dissipation limitation. Inbuilt heating allows these sensors to operate at temperature much higher than the surrounding. Humidity and pressure of the sample gas also plays a role strictly according to the type of the sample used. However, there had always been a difficulty to analyse and eliminate the drifts due to temperature, humidity and vapour pressure variation in the surrounding environment in which samples are placed or housed for odour detection by MOS based E-nose sensors [6].

Drift parameters of MOS E-nose sensors due to temperature, humidity and vapour pressure of the sample vapour has not been adequately addressed so far. These parameters can be used by neural networks to compensate certain amount of drift in accordance with set algorithms for the particular sensor and sample combination. It, however, appears that drifts in sensors due to temperature, humidity and vapour pressure are highly sample specific. It, therefore, becomes necessary to determine the nature of the drifts in E-nose sensors for a particular sensor-sample combination. Each time an experiment is conducted for the given sample, the compensation is required to classify a sample correctly. Experiments were performed on tea samples during this research work to determine the drift coefficients.

5.2 Drifts Reduction in E-nose Sensors

Electronic noses consist of an array of non-specific sensors with overlapping selectivity combined with pattern recognition tools such as principal component analysis (PCA), partial least squares (PLS) or artificial neural networks (ANNs) [6]. These techniques measure sample odour in gas phase. Unfortunately, drift in sensor signals is often a severe problem in these measurement techniques. For example, Frank [7] simulated the effect of linear drift, which resulted in a decrease of cluster separation of different samples in a multivariate space in gas sensors. It is clear that sensors must be treated properly during stable measurement conditions otherwise the sensor surfaces are deteriorated by physical changes such as ageing, poisoning, humidity and temperature effects, giving rise to long-term drift. The use of mathematical algorithms to minimize drift effects has been reported in the literature. For example, additive [8] and multiplicative [9] drift effects of MOSFET (metal oxide semiconductor field-effect transistor) and MOS gas sensors were successfully corrected using reference gas samples.

Increasing refreshing time duration between the sample readings might, in most of the cases, eliminate drifts related to memory effects. Aging problem in the sensors can be tackled by the replacement of old sensors. Recently, a new multivariate method called component correction (CC) was developed [2]. Here, drift direction of reference gas samples in a multivariate space obtained from electronic nose measurements was subtracted from other samples [10]. These methods are rather simple and straightforward to use, but require the drift of different samples to be linear in the multivariate space. Also the drift in reference samples (used for standardisation of E-nose response) should be similar to other samples, especially for additive and multiplicative drift correction. Other, more sophisticated methods without using reference samples have also been reported [11, 12]. However, drift can still be introduced by metal oxidation resulting in an increased porous surface layer, which increases background signals. Therefore, the use of mathematical algorithms can be a supplement but not permanent solution in preventing drift.

Hence the drift introduced by temperature, humidity and pressure needs to be duly analysed. It is well understood that these techniques require physical treatments and stable measurement conditions combined with good calibration routines to become successful [13]. This research was planned to find out a technique of removing drift effects in MOS based E-nose responses by determining suitable drift coefficients. This will effectively eliminate the burden of the frequent and long training sessions of the E-nose system. Four MOS based E-nose sensors are calibrated for the sample of tea and kerosene gas under drift conditions.

5.3 Temperature Drift Determination

Sensor response deteriorates under temperature fluctuations in the ambient (normally sample vapour temperature). Normally temperature is kept constant during the sampling of the tea vapour but for determination of the temperature drift coefficients, the temperature of sample vapours is increased and sensor response is observed. Figure 5.2 displays the sensor response as the temperature is raised.

5.3.1 Physical Principles

If the influence of ambient temperature is small, a reasonable sensor operation with a stable signal baseline may be maintained. This can be achieved either by keeping the operational temperature of the device constant at an elevated level or by reducing the influence of temperature to the device as much as possible. In practice, both methods can be applied. An external temperature controlled heater can realize the first method. For the second method, at first a good comprehension of the temperature influence to the sensor surface is required.

Gases, unlike solids and liquids have indefinite shape and volume. As a result, they are subject to changes in pressure, volume and temperature. Real gas behavior is actually complex. An ideal gas is considered to be a point mass and collisions between ideal gases molecules are assumed to be perfectly elastic. Pressure (P), Volume (V), Temperature (T) and the Moles (n) for a gas are related by the following relationship, known as ideal gas equation:

$$PV = nRT \tag{5.1}$$

or

$$n = PV / RT \tag{5.2}$$

Here, R is ideal gas law constant. The most common used value of R, when dealing with gases, is given by 0.0821 L· atm/mol· ^{0}K. When pressure is increased, at constant temperature and volume, the number of moles increases, thereby increasing the odourant molecule density or in other word parts per million (ppm) increases. This is in conformity with the results obtained in our experimentations since increased ppm increases E-nose response. If the temperature is increased and the volume is naturally allowed to expand at constant pressure then there is no effect on ppm of the sample initially present in the air. However, increasing temperature can affect in two ways, first the evaporation rate in the sample molecules is high and second, the average kinetic energy of the sample molecules is increased. Temperature influences the adsorption processes at the sensitive surface. Both the quantity of adsorbed sample vapour and the adsorption dynamics will be affected [14]. The change in the value of work function and the time for saturation of sensor surface depends on temperature

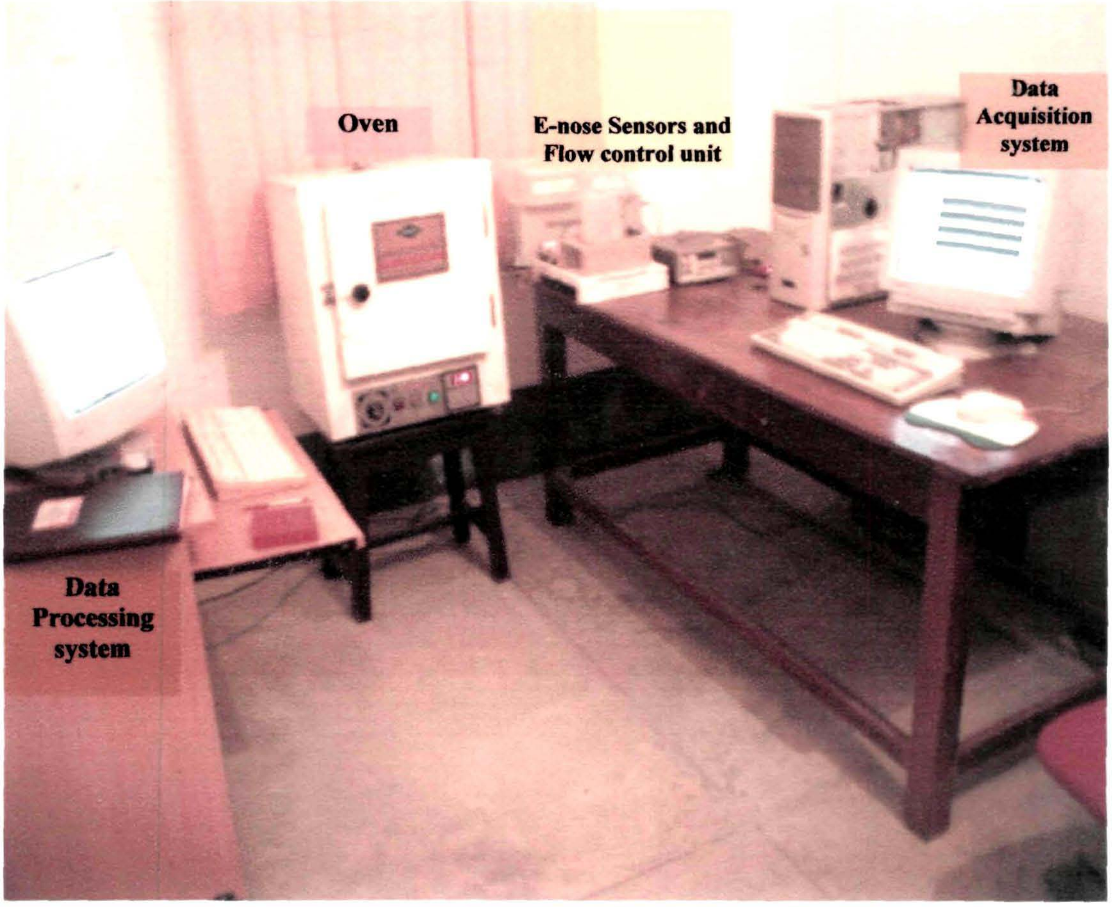


Figure 5.1: Photographic view of Oven and E-nose system fo determination of temperature drift coefficient. Oven is used for varying th temperature of sample from 25^0 C to 110^0 C under constant VOC.

[15]. Temperature affects both the mobility of charge carriers and the level of the Fermi Energy [16].

It can be reasonably assumed that this phenomenon possibly causes greater diffusion of the odourant molecules into the lattice of the sensor film. The temperature alters mobility and thereby conductivity of the sensor surface, resulting drift in the sensors.

5.3.2 Experimental Procedure

The experimental block diagrams are shown in Fig. 3.8 and Fig. 4.7. Procedure is explained in the following sub sections.

5.3.2.1 Sample Collection

A special tea sample was received from the garden of Science and Technology Entrepreneurs' Park, Indian Institute of Technology, Kharagpur. Raising of the Tea Garden and processing of Tea has been done with the expertise provided by the Department of Agricultural and Food Engineering, IIT, Kharagpur, sponsored by the department of Science and Technology, New Delhi. The samples were collected in hermetically sealed packs and tear opened just before beginning of the experiments.

5.3.2.2 Test Procedure

Tea sample was placed in a stainless steel sample vessel. Sample vessel is kept in a temperature controllable oven (Fig. 5.1). It is connected to the sensor chamber through pipeline. With the help of diaphragm pump the vapour of the tea sample is transported to the sensor chamber at constant flow rate. The oven temperature is varied from 25° to 110° C. The sample response from the sensors was sampled at the rate of 1 Hz and stored in data file through data acquisition card. Temperature was increased at the rate of approximately 1° C per minute. The typical sample of response is shown in the Fig. 5.2. This figure indicates that sensor response is increased with increase in temperature. The reference vessel is placed in room environment to provide fresh airflow for removing any residual traces of tea sample vapour left on the sensor surface. When the temperature is allowed to return to its initial value, sensor response does not immediately return to the base line. There is a considerable drift in the response.

Table 5.1: Data set composition for the temperature drift determination

Experiment	No. of data Sets	No. of vectors (for 4 sensors)	Each Vector length	Total sampling time (hrs)
1	16	64	1 X 5100	22.66
2	20	80	1 X 5100	28.33
3	18	72	1 X 5100	25.50
4	16	64	1 X 5100	22.66

5.3.2.3 Data Acquisition

Online logging for data collection and graphical display (Fig. 5.2) of sensor response are performed in GENIE environment. Data acquisition was continuous for particular experiment and lasted for long timescales (about 22 to 28 hours for each experiment) to get wide response from the E-nose sensors. A total of 16 to 20 data sets in each experiment were gathered. The data set in each experiment consisted of 64 to 80 vectors (each 1 X 5100 dimensions) over the range of temperature from 25° C to 110° C. Four such experiments were conducted allowing one-day rest for E-nose system after each experiment. The data set composition and time scales are listed in Table 5.1. The experiments continued for about 10 days.

5.3.3 Data Processing

Once the data acquisition is completed, the same are processed using MATLAB 6.0 platform and plotted in GENIE and MATLAB environments. Fig 5.2 is shown with the graphical displays of the actual E-nose sensors' response. The graphical analysis was used to determine the slopes of the curves.

The Fig. 5.3 is shown with actual and noisy data display. All 70 data sets (Table 5.1) are averaged element by element (concatenated average of total of 70 matrices) of matrices of dimensions of 4 X 5100 each. After taking concatenated average, a single matrix of dimensions of 4 X 5100 is formed. This matrix is then re-plotted as the average response of four E-nose sensors on the voltage (response) – temperature semi log graph scales. The resultant response plot is shown in figure 5.4. It shows that response is linear on semi log scales and most of the noisy fluctuations are filtered out. The slopes of sensor response curves are determined to estimate the change in the sensor response per unit change in the temperature.

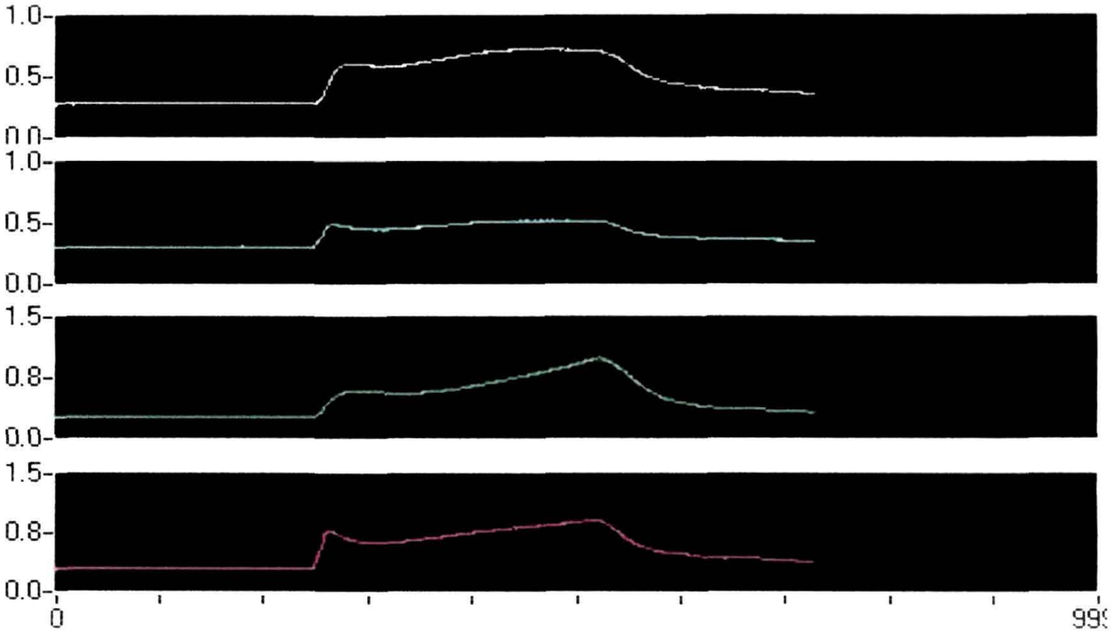


Figure 5.2: Four E-nose sensors response data display in GENIE GUI as the temperature was increased from 25° C to 110° C linearly under constant VOC.

5.3.4 Drift Coefficients

It can be seen in figures 5.2 and 5.3 that increasing temperature of tea sample vapour has altered the output response of the E-nose sensors. Data response has shown drift effects under increasing temperature of tea sample vapour. Based on the slops of the curves in figures 5.2 and 5.3, temperature coefficients are determined.
The following relation gives temperature drift coefficient:

$$\lambda_{i\,T} = \Delta Vi / \Delta T_i \quad |_{\text{at constant RH and VP}} \tag{5.3}$$

Where:

$\lambda_{i\,T}$ = Temperature Coefficient for i^{th} sensor, volt / ° C

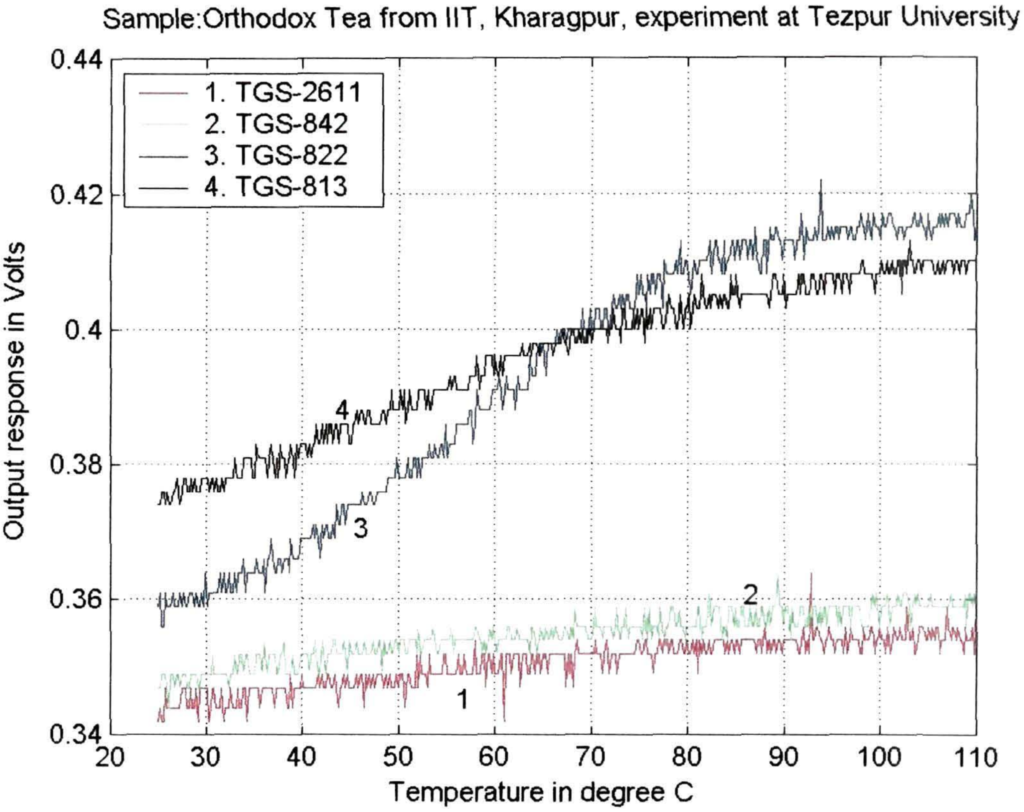


Figure 5.3: Actual E-nose sensors response display in MATLAB under the influence of temperature drift

Table 5.2: Drift Coefficients for E-nose sensors

Name of Sensor	Temperature Coefficient λ_{iT} (mV / °C)
TGS- 2611	0.20
TGS-842	0.28
TGS-822	1.30
TGS-813	0.72

ΔV_i = Change in i^{th} sensor response, V

ΔT_i = Change in sample vapour temperature o C for i^{th} sensor

RH = Relative Humidity, (60 %, constant)

VP = Sample Vapour Pressure (at atmospheric pressure, constant)

i = $1, ..., 4$ (Number of sensors in experiment)

The temperature drift coefficients are evaluated by using equation 5.3 in case of four TGS sensors ($i = 1, ..., 4$) used in E-nose system. The coefficients are enlisted in Table 5.2.

For example the average response voltage of the sensor TGS 822 at 50 ° C is about 0.375 volts and at 70 ° C is about 0.400 volts (figure 5.4). The ratio of change in response voltage to the change in temperature is about 0.00125 volts per ° C (this value is only for example and not the actual). Similarly the ratios of temperature change can be estimated for other sensors' response. The average of all ratios estimated over the temperature range for all data samples for a sensor is termed as temperature drift coefficient, $\lambda_{(TGS\text{-}822)\ T}$ (for TGS-822). In the similar way temperature coefficients (Table 5.2) for other sensors, TGS-2211, TGS-813 and TGS-842 are evaluated.

5.3.5 Results and Discussions

E-nose sensors were tested for the VOCs of tea sample vapour at a temperature ranging from 25° C (room temperature) to 110° C. It was observed that the variation due to temperature of tea vapour sample does not affect sensor response in a considerable form below the temperature of 30° C. This is perhaps threshold for tea vapour sample above which drift starts deteriorating the E-nose sensor response. However, on increasing temperature further above 30 ° C, it was observed that output response increased considerably. The increase in response continues up to a temperature of about 80° C as shown in Fig. 5.3 and Fig. 5.3. Beyond the temperature of 80° C, once more sensor response becomes almost stable with very little increase. This can be attributed to the saturation of the sensor surface [17]. There are slight variations in these results for different sensors, which may be due to different semiconductor physics of manufacturing sensors.

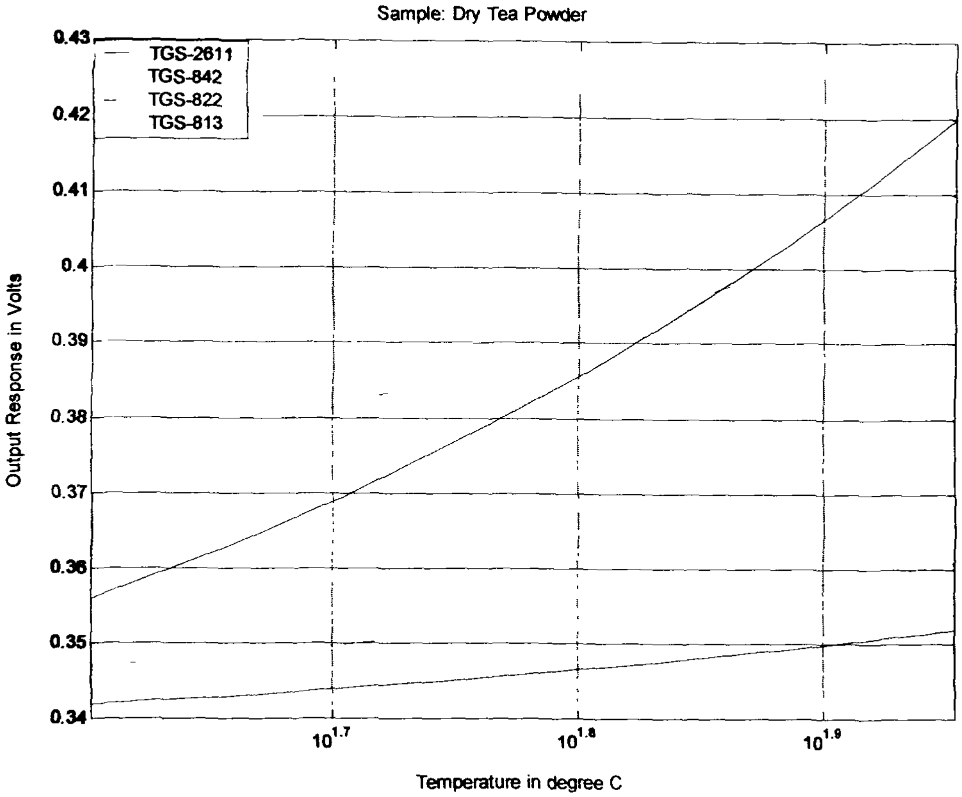


Figure 5.4: Average E-nose sensors response display on semi log graph in MATLAB under the influence of temperature drift. X-axis has temperature units on logarithmic scale. Temperature varies from 25° C (initial) to 110° C (final)

5.4 Humidity Drift Determination

The humidity is a dynamic physical quantity, which, if not controlled can affect sensors' response adversely. To conduct the experiments under variable relative humidity conditions, an AC controlled room was identified. Hygrometer indicated humidity value in continuous time. Initially, the room was open for atmospheric relative humidity value for a given climatic and weather conditions on the day of experiment. It was observed that at the beginning of the experiment relative humidity was about 90% (coincidentally, it was rainy season and hence atmospheric humidity was high). AC room was put on and simultaneously E-nose experiments were started on tea samples. Relative humidity decreased to 40% after about 5000 seconds (83.33 minutes) time.

Effects of humidity drifts on sensor response are shown in figure 5.5. Sensor response increases under increasing humidity (or as shown in Fig. 5.5, sensor response decreases as humidity is decreased) in the ambient (normally humidity in sample vapour) environments. The humidity is required to be constant during the sampling of the tea vapour to avoid drifts in sensors. For determination of the humidity drift coefficients, the humidity in sample vapour is increased and sensor response was observed. Fig. 5.5 displays the sensor response as the relative humidity is decreased from 90% (initial) to 40% (final).

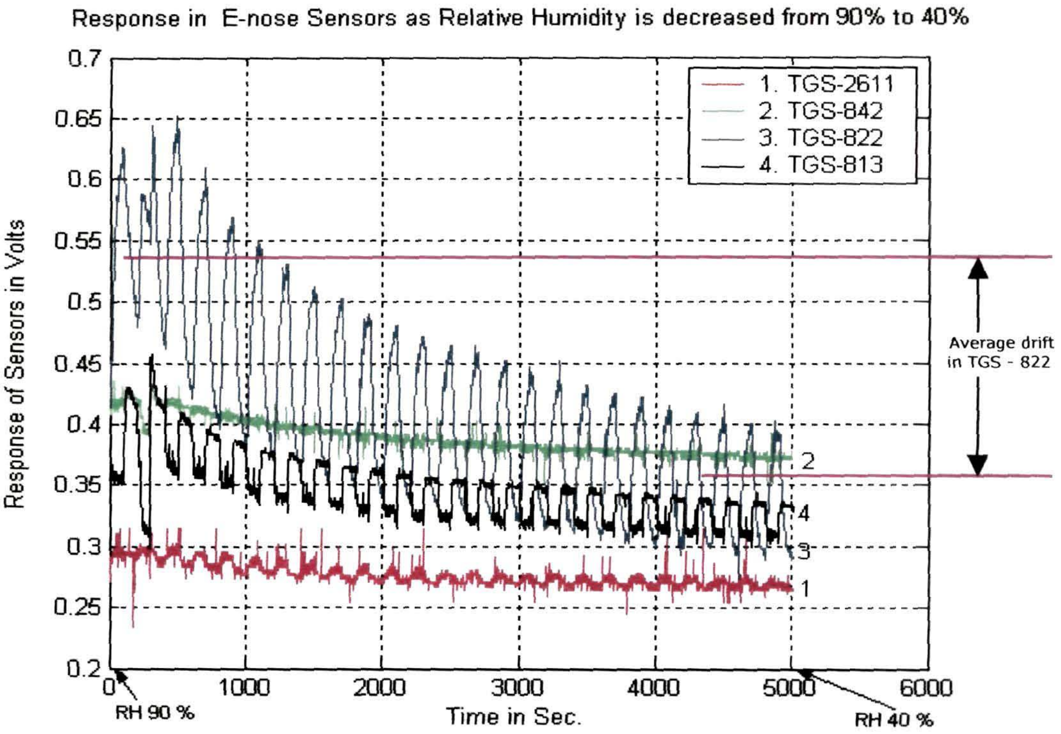


Figure 5.5: E-nose sensors response data (actual, with noise) display as humidity is decreased from 90 % to 40 % for humidity coefficient determination.

5.4.1 Physical Principles

It is well known that water vapour strongly influences both the conductance in air and the gas response of chemo-resistive tin dioxide (SnO_2) gas sensors [18, 19, 20]. The other drift phenomena of ageing, appears in the operation of chemoresistive SnO_2 gas sensors is also due to the action of water vapour [21]. Another effect of water vapour is that of accentuating the deviations from linearity observed at low partial pressure of the sample vapours. Experimental results were explained in a tentative model [21] in which a hypothesis was assumed that water reacts dissociatively with one type of lattice oxygen on the SnO_2 surface, but in two different ways, giving rise two different types of OH ions (surface reaction products). One of OH^- ion replaces the reactive oxygen sites without producing any free carrier. The other, OH^- ion on the contrary, produces free carriers without blocking the oxygen sites. In rapid transitions from dry to wet air, the sudden increase in conductance is observed, which may be explained by the generation of OH^- ions and subsequent decay by the recombination and trapping on different paths of the charged particles generated [22]. A possible recombination path is that of OH^- ions with the lattice oxygen vacancies resulted in the interaction with water [22].

The water vapour competes with reducing gases (tea vapour VOC's are reducing agents in chemical reactions on the sensor surface) in reacting on the SnO_2 surface with the same oxygen reaction sites, thus an additional response is generated in the sensors due to water vapour. This is the reason that there is a higher response in E-nose sensors in presence of higher relative humidity (Fig. 5.5)

5.4.2 Experimental Procedure

The setup for experimentations for humidity drift coefficient determination is shown in Fig. 3.8 and Fig. 4.7. The procedure for conducting experiments is explained in the following sub sections.

5.4.2.1 Sample Collection

The same type of tea samples, as in case of a temperature drift determination, is used for humidity drift determination. The samples were brought from the Garden of Science and Technology Entrepreneurs' Park, Indian Institute of Technology, Kharagpur. The samples were hermetically sealed in packs and opened just before beginning of the experiments.

5.4.2.2 Test Procedure

Tea sample was placed in a stainless steel sample vessel. The sample vessel is kept in a room where humidity can be changed as shown in Fig. 4.7. The sample vessel is connected to the sensor chamber through pipeline. With the help of diaphragm pump the vapour of the tea sample is transported to the sensor chamber at constant flow rate. The room humidity was decreased from 90 % (initial) to 40 % (final) at a rate of 0.6 % per minute (approximately). The sample responses from the sensors were sampled at the rate of 1 Hz and stored in data file through data acquisition card. Each data set took about 108 minutes. The typical samples of responses are shown in the Fig. 5.5. These figures indicate that sensor response is decreased with decreasing humidity. The reference vessel is placed in a normal room environment to provide fresh airflow for removing any residual traces of tea sample vapour left on the sensor surface. When the humidity is allowed to return to its initial value, sensor response does not immediately return to the base line. These sensors took long time to return the initial base line and needed a quite long refreshing sessions. The water vapour caused considerable drift in the sensor surface as indicated in Fig 5.5.

5.4.2.3 Data Acquisition

GENIE environment is used for online logging for data collection and graphical display (figure 5.5) of sensor response. Data acquisition continued for about 17 to 22 hours to get wide response from the E-nose sensors. A total of 12 data sets in each experiment were gathered. The data sets in each experiment consisted of 48 vectors (each of 1 X 6495 dimensions in experiment 1 and 1 X 5000 dimensions in experiment 2) over the range of relative humidity varying from 90 % to 40 %. Two such experiments were conducted allowing two days rest for E-nose system after each experiment. The data set composition and time scales are listed in Table 5.3. The experiments continued for about 8 days (2 days for initial refreshing, 1 day for first experiment, 2 days for refreshing and then 1 day for second experiment followed by 2 days for final refreshing)

Table 5.3: Data set composition for the humidity drift determination

Experiment	No. of data Sets	No. of vectors (for 4 sensors)	Each Vector length	Total sampling time (hrs)
1	12	48	1 X 6495	21.65
2	12	48	1 X 5000	16.66

5.4.3 Data Processing

After the data acquisition is accomplished, the data were plotted for visual analysis and processed using MATLAB 6.0 platform and GENIE environments. Fig. 5.5 is shown with the graphical displays of the actual E-nose sensors' responses. These graphical plotting was used to determine the slopes of the curves. Quadratic curve fitting was estimated for each sensor response and then slope of the curve is evaluated.

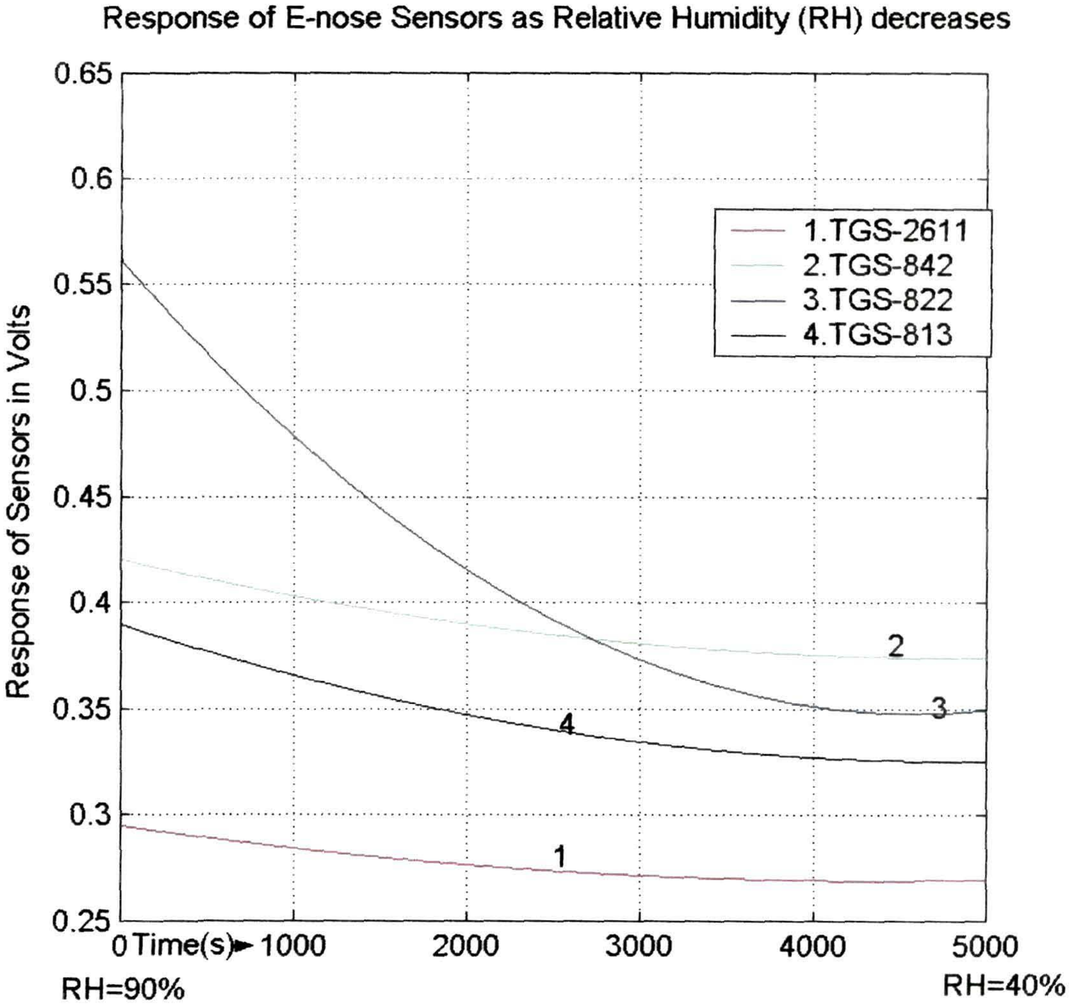


Figure 5.6: E-nose sensors response data (averaged and noise filtered out by curve fitting function) display as humidity is decreased. The slopes of curves are equivalent to humidity drift coefficient.

All 24 data sets (Table 5.3) are averaged element by element (concatenated average of total of 24 matrices) of matrices of dimensions of 4 X 5000 of 12 data sets in experiment 1 and dimensions of 4 X 6495 of 12 data sets in experiment 2. After taking concatenated average, two matrices of dimensions of 4 X 5100 and 4 X 6495 are formed. These matrices are then re-plotted as the average response of four E-nose sensors on the voltage (response) – relative humidity (RH in %) scales. Noise was filtered out from the data sets using quadratic curve fitting functions in MATLAB programming. This can, more aptly, be called curve fitting with negligible residues. The resultant response plot is shown in figure 5.6. It shows that response is linearly decreased with humidity. Most of the noisy fluctuations are filtered out. The slopes of sensor response curves are determined to estimate the change in the sensor response per unit change in the % relative humidity.

5.4.4 Drift Coefficients

The Fig. 5.5 shows that variation in relative humidity of tea sample vapour has altered the output response of the E-nose sensors. Data response has shown drift effects under decreasing humidity in tea sample vapour. Based on the slopes of the curves in Fig. 5.6, humidity coefficients are determined.

The following relation gives humidity drift coefficient:

$$\lambda_{i\,H} = \Delta Vi \,/\, \Delta H_i \quad |_{\text{at constant T and VP}} \qquad (5.4)$$

Where:

$\lambda_{i\,H}$ = Humidity Coefficient for i^{th} sensor, volt / unit change in RH (%)

ΔV_i = Change in i^{th} sensor response, V

ΔH_i = Change in sample vapour humidity in % for i^{th} sensor

T = Temperature (25° C, constant)

VP = Sample Vapour Pressure (at atmospheric pressure, constant)

i = $1,\ldots,4$ (Number of sensors in experiment)

The humidity drift coefficients are evaluated by using equation 5.4 in case of four TGS sensors ($i = 1,\ldots, 4$) used in E-nose system. The coefficients are listed in Table 5.3

For example the average response voltage of the sensor TGS 842 at 90 % RH is about 0.425 volts and at 50 % RH is about 0.360 volts (figure 5.6). The ratio of change in response voltage to the change in RH is about 0.001625 volts per unit change in % RH (1.625 mV / % RH). Similarly the ratios for other sensors for humidity change can be estimated. The average of all ratios estimated over different humidity response data samples is termed as humidity drift coefficient, $\lambda_{(TGS\text{-}842)\ H}$, for the sensor TGS 842. In this way humidity coefficient for other sensors, TGS - 2211, TGS – 813 and TGS-842 are evaluated (Table 5.4).

Table 5.4: Humidity Drift Coefficients for E-nose sensors for tea flavours

Name of Sensor	Humidity Coefficient $\lambda_{i\ H}$ (mV/% RH)
TGS- 2611	0.89
TGS-842	1.62
TGS-822	7.59
TGS-813	2.93

5.4.5 Results and Discussions

E-nose sensors were tested for the VOCs of tea sample vapour at a relative humidity ranging from 90 % (initial) to 40 % (final). The variation due to relative humidity of tea sample does not affect sensor response in a considerable form below the RH of 50 %. It can be seen from figure 5.6 that E-nose response becomes almost horizontal below 50 % RH. This is perhaps threshold for tea vapour sample above which drift has deteriorating effects on the E-nose sensor response. However, while % RH above 50 %, it was observed that output response increases considerably. The increase in response continues up to a RH of about 90 % (as our experimental limitations allowed us only up 90 % of RH) as shown in figure 5.5 and 5.6. There are variations in these results for different sensors, which may be attributed to different lattice formation and crystallization of solids during manufacturing of metal semiconductor sensors.

5.5 Partial Pressure Drift Determination

Sensor response increases under increasing partial pressure in the ambient (normally partial pressure of sample vapour) environments. The pressure is required to be constant during the sampling of the sample vapour to avoid drifts in sensors. For determination of the pressure drift coefficients, the pressure in sample vapour was increased and observed sensor response. Figure 5.7 shows a modified pressure cooker used to buildup the kerosene sample vapour partial pressure (component of pressure in an air mixture due to kerosene sample molecules is called partial pressure of kerosene sample). The pressure is increased from 1 atm (101.3 kPa) to 2 atm (202.6 kPa). Pressure cooker was fitted with a non-return valve in the lid. The pressure release knob was connected to the sensors' chamber through pipes. A hand usable air-inflating pump was used to pump the room air into the pressure cooker. The pressure value is read from pressure gauge.

5.5.1 Physical Principles

If ambient pressure of sample vapour is equal to that of atmospheric pressure, the influence of it on sensor operation may be a stable signal baseline. In practice, normally sampling of vapours is done at normal atmospheric conditions (STP).

Gases behaviour is explained in section 5.3.1. The equations (5.1) and (5.2) give physical relationships between physical parameters of temperature, pressure and humidity.

If the number of moles and thus partial pressure of sample vapour is increased at constant temperature and volume then the number of moles increases. This increases the odourant molecule density and so parts per million (ppm). This is in conformity with the results obtained in experimentations since increased ppm increases E-nose response. The increasing pressure can affect sensor response in large amount. Although, the evaporation rate in the sample molecules is not changed significantly, but additional air inflow at constant temperature and volume can cause the average kinetic energy to increase drastically. High pressure influences the adsorption processes at the sensitive surface [14]. The quantity of adsorbed sample vapour increases. Water vapour also has increased influence on the sensors' response. These all factors have cumulative drift effects on the sensor response. Diffusion increases and more chemical reactions take place with pre-absorbed oxygen and water vapour on the surface of sensor giving rise to higher response in the sensors.

5.5.2 Experimental Procedure

The experimental setup for pressure coefficient determination is shown in figures Fig. 5.7 (also Fig. 3.8 and Fig. 4.7). The procedure for experiments is explained in the following sub sections.

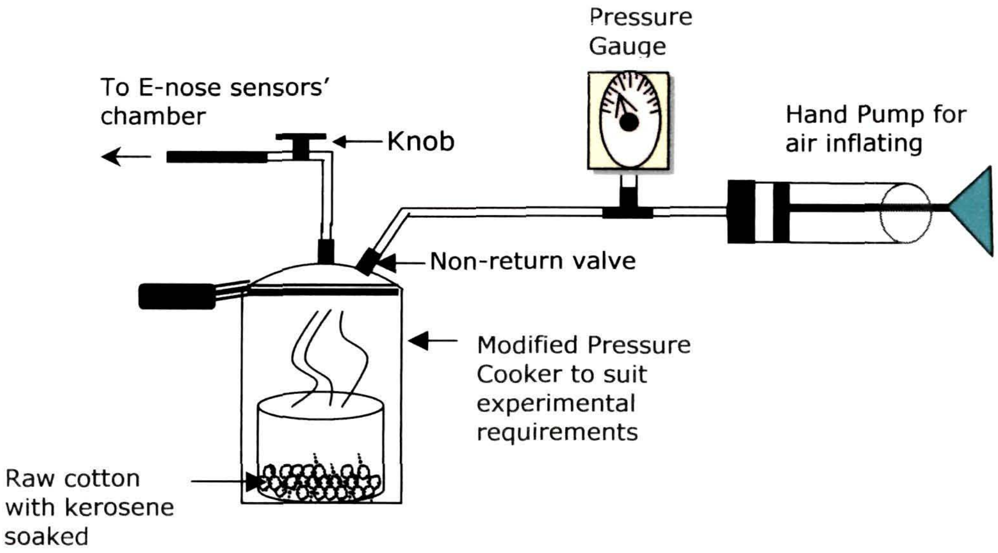

Pressure Gauge
To E-nose sensors' chamber
Knob
Hand Pump for air inflating
Non-return valve
Modified Pressure Cooker to suit experimental requirements
Raw cotton with kerosene soaked

Figure 5.7: Modified Pressure Cooker for determination of the pressure drift coefficients for distilled kerosene sample

5.5.2.1 Sample Collection

The sample choice for the pressure coefficient is made based on the fact that normally pressure effects occur in leakage related areas. These could be chemical spillovers, leakages of different hydrocarbons in chemical plants and hydraulic leaks in machineries. In the view of these, kerosene vapour at high pressure was a choice to choose as a sample for analysis of pressure drift coefficients.

The distilled kerosene under natural evaporation process forms the kerosene vapour, which was used as sample vapour at higher pressure for the E-nose. Kerosene sample was prepared by fractional distillation method in the Chemical Science laboratory. Distillation was necessary to purify the laboratory kerosene from possible pollutants.

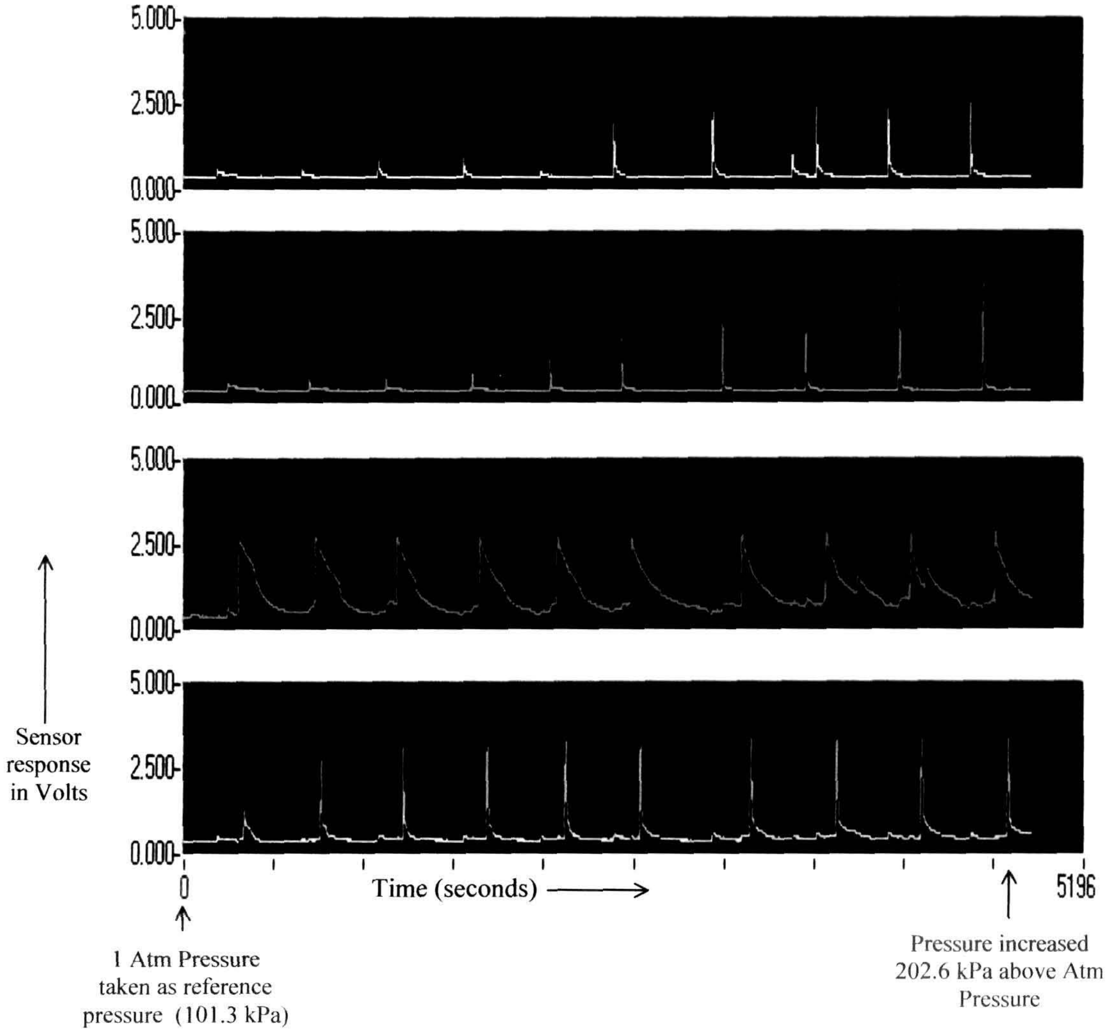


Figure 5.8: Typical response of the 4 E-nose sensors as pressure of kerosene sample was increased above STP from 101.3 to 202.6 kPa linearly.

5.5.2.2 Test Procedure

E-nose sensors were housed in a chamber, which was connected through pipes to a pressure cooker modified to function as a pressure control device. The set up is shown in Fig. 5.7. Kerosene sample was soaked into a raw clinical cotton lump and then it was placed in pressure cooker. Soaking with cotton increases evaporation similar to the famous wet bulb hygrometer principle. Then, through air inflating pump, room air was pumped into the cooker to increase pressure. Pressure was observed through a pressure gauge fitted on the cooker lid as shown in the Fig. 5.7.

Once a pressure is buildup to a desired value, knob was open to allow the flow of kerosene vapour to E-nose sensors. This condition was similar to a leakage in pressurized systems. It is believed that the kerosene had evaporated into the air in the pressure cooker and uniformly diffused through out the volume in accordance with Pascal's Law of Pressure. Pressure was increased from 1 atm (initial, equal to 101.3 kPa) in steps of 10 kPa in each step up to 2 atm (final, equal to 202.6 kPa). At each pressure value, E-nose response was observed for all data sets consisting of data vectors as given in Table 5.5. Five experiments were conducted in total. The sample response from the sensors was sampled at the rate of 1 Hz and stored in data file through data acquisition process. Each data set took about 87 minutes. A typical kerosene sample response from E-nose is shown in the Fig. 5.8 under varying pressure. The Spikes in the Fig. 5.8 corresponds to the pressure value.

Table 5.5: Data set composition for the pressure drift coefficients determination.

Experiment	No. of data Sets	No. of vectors (for 4 sensors)	Each Vector length	Total sampling time (hrs)
1	10	40	1 X 5196	14.44
2	8	32	1 X 5000	11.11
3	4	16	1 X 4000	04.44
4	5	20	1 X 5100	07.08
5	5	20	1 X 5100	07.08

The Fig. 5.8 indicates that sensor response is increased almost linearly with increasing pressure. After each experiment refreshing of the sensors was done by providing a fresh airflow for removing any residual traces of kerosene vapour left on the sensor surface. When the pressure is allowed to return to its initial value, sensor response immediately return to the base line. These sensors took very short time to return the initial base line and didn't need long refreshing sessions.

5.5.2.3 Data Acquisition

GENIE GUI is used for monitoring and online logging of data collection and graphical display (Fig. 5.8) of sensor response. Data acquisition continued about 7 to 15 hours for a single experiment. A total of 16 to 40 data sets in each experiment were gathered. The data sets in all experiments consisted of 128 vectors (each of 1 X 5196 dimensions in experiment 1, 1 X 5000 dimensions in experiment 2, 1 X 4000 dimensions in experiment 3 and 1 X 5100 dimensions in experiments 4 and 5) over the range of pressure varying from 1 atm to 2 atm. Five such experiments were conducted allowing one day rest for E-nose system after each experiment. The data set composition and time scales are listed in Table 5.5. The experiments continued for about 10 days (1 day for each of five experiments and 1 day refreshing after each experiment).

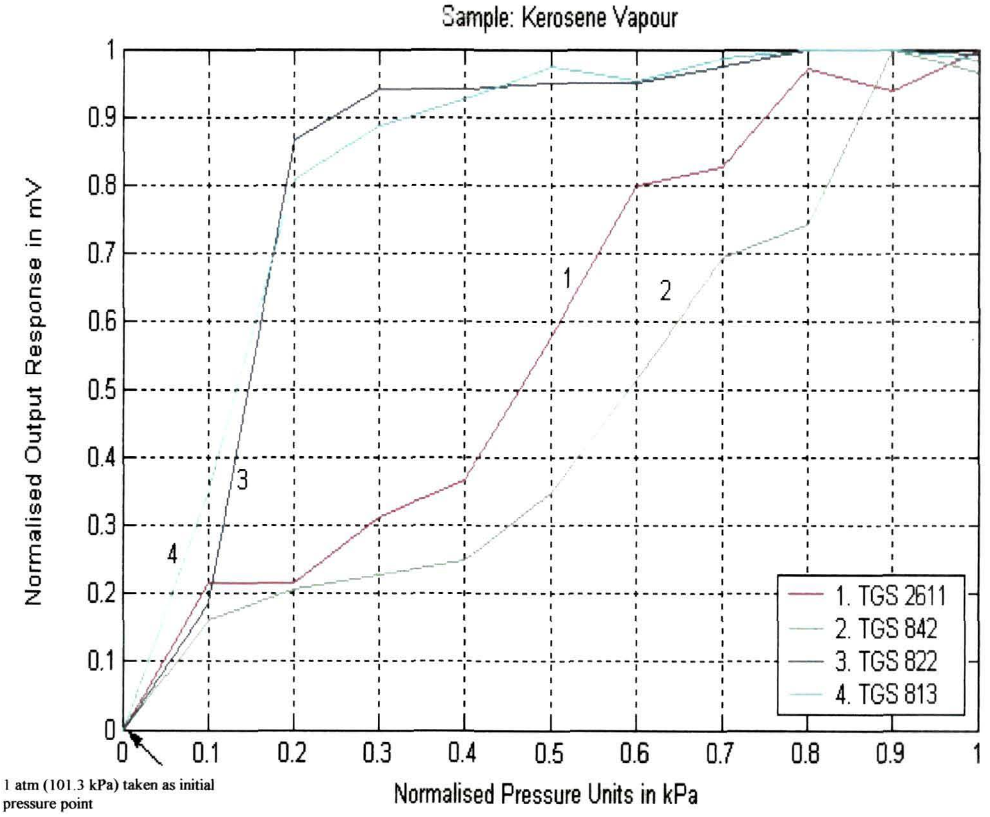


Figure 5.9: E-nose sensors actual response display for pressure coefficient evaluation.

5.5.3 Data Processing

The data sets were gathered during online logging. Once the data acquisition is accomplished, the data sets are plotted for analysis of drift effects using MATLAB 6.0 platform and GENIE environments. Figures 5.9 are shown with the graphical displays of the actual E-nose sensors' responses. These graphical plotting was used to determine the slopes of the curves. Quadratic curve fitting was estimated for each sensor response and then slope of the curve is evaluated (Fig. 5.10).

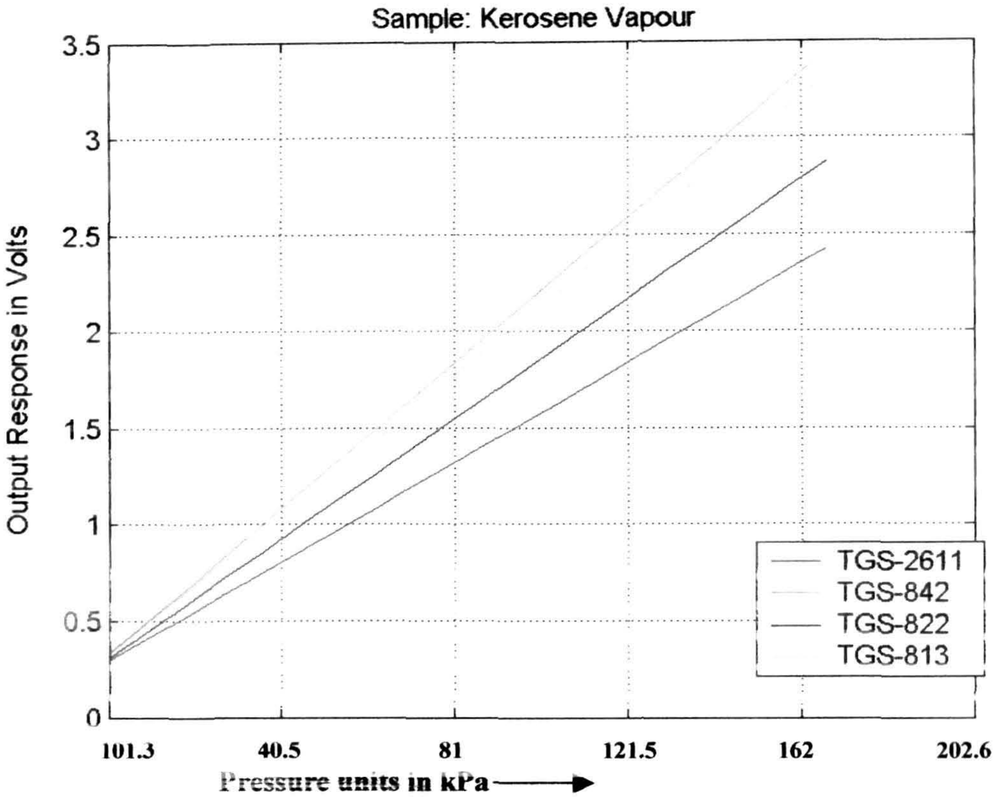


Figure 5.10: E-nose sensor response display for pressure variation of kerosene. Data response is displayed in quadratic curve fitting function with minimum of residues.

All 32 data sets are averaged element by element (concatenated average of total of 24 matrices) of matrices of dimensions as mentioned in Table 5.5 These matrices are then re-plotted as the average response of four E-nose sensors on the voltage (response) – pressure (kPa) scales. Noise in data was filtered out using quadratic curve fitting functions in MATLAB programming. The resultant response plot is shown in Fig. 5.10. It shows that response linearly increases with pressure. Most of the noisy fluctuations are filtered out. The slopes of sensor response curves are determined to estimate the change in the sensor response per unit change in pressure.

To ascertain high accuracy and validity of coefficients determined, a rigorous data analysis and calculations involving various statistical properties were performed. These include taking mean, standard deviation, maximum and minimum range of the data, eigen vectors, curve fittings and peak values etc.

Table 5.6: Pressure drift Coefficients for Kerosene sample

Name of Sensor	Pressure Coefficient λ_{iP} (mV / kPa) Above STP
TGS- 2611	10.3
TGS-842	15.0
TGS-822	12.4
TGS-813	14.4

5.4.4 Drift Coefficients

The Fig. 5.9 shows that variations in pressure of kerosene sample vapour have altered the output response of the E-nose sensors. Data response has shown that output from E-nose has drifted away from normal values under increasing pressure in kerosene sample vapour. Based on the slopes of the curves in Fig 5.10, pressure coefficients are determined.

The following relation gives pressure drift coefficient:

$$\lambda_{i\,P} = \Delta Vi / \Delta P_i \quad |_{\text{at constant T and H}} \qquad (5.5)$$

Where:

$\lambda_{i\,P}$ = Humidity coefficient for i^{th} sensor in volt / unit change in pressure (kPa)

ΔV_i = Change in i^{th} sensor response in volts

H = Sample vapour humidity, (65%, constant)

T = Temperature, (25° C, constant)

P_i = Sample Vapour Pressure (from 101.3 kPa to 202.6 kPa)

i = *1,...,4* (Number of sensors in experiment)

The pressure drift coefficients are evaluated by using equation 5.5 in case of four TGS sensors (*i = 1, ..., 4*) used in E-nose system. The pressure coefficients are enlisted in Table 5.6

For example the average response of the sensor TGS - 813 at 50 kPa is about 1.05 volts and at 150 kPa is about 2.5 volts (Fig. 5.9). The ratio of change in response voltage to the change in pressure is about 0.0144 volts per unit change in pressure (14.4 mV / kPa). Similarly the ratios for entire range of pressure change can be estimated. The average of all ratios estimated over different pressure values is termed as pressure drift coefficient, $\lambda_{(TGS-813)\ P}$, for the sensor TGS 813. In these way pressure coefficients for other sensors, TGS – 2211, TGS – 813 and TGS - 842 are evaluated (Table 5.6).

5.5.5 Results and Discussions

E-nose sensors were tested for the VOCs of kerosene sample vapour at a high pressure ranging from 1 atm to 2. The variation due to increasing pressure of kerosene vapour sample has strong effect on the sensor response. It can be seen from Fig. 5.8 that E-nose response increases almost linearly for whole range of increase in pressure. There are variations in these results for different sensors, which may be attributed to the similar factors of temperature and humidity drift cases. The increase in the sensor response is in conformity with the physical principles. Pressure coefficients are much larger in values compared to the temperature and humidity coefficients. Partial pressure of sample vapour plays far more role in the drift of sensor response.

5.6 Validation of Drift Compensation

Compensation for drift is an important factor to improve results. The aim of drift validation is to verify improvement of classification of E-nose system under drift compensation. For this the flavours of the tea under drift condition were classified by using Electronic-nose (E-nose) system with application of temperature and humidity drift compensation techniques. Pressure drift was determined for the sample of kerosene. The drift coefficients for E-nose sensors due to temperature, humidity and pressure variations in samples of tea and kerosene are determined separately as listed in Tables 5.2, 5.4, 5.6 respectively. The coefficients were used to eliminate drift in E-nose response data during online capturing and processing. PCA and ANN based pattern recognition (PARC) techniques are used for discrimination and classification of electronic nose response data.

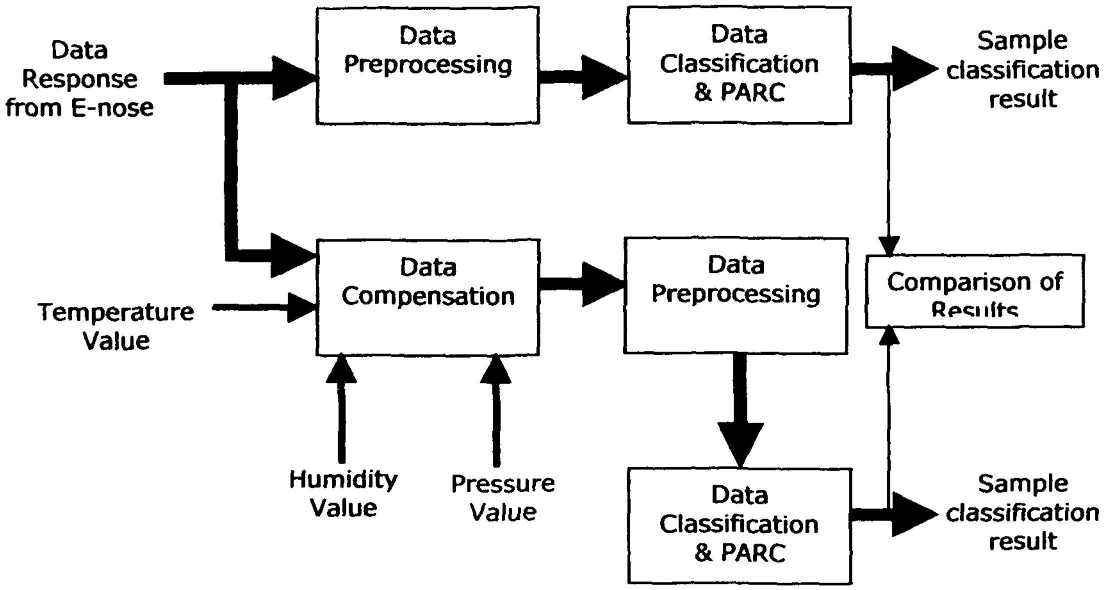

Data Response from E-nose
Data Preprocessing
Data Classification & PARC
Sample classification result
Temperature Value
Data Compensation
Data Preprocessing
Comparison of Results
Humidity Value
Pressure Value
Data Classification & PARC
Sample classification result

Figure 5.11: Drift compensation procedure for E-nose sensors response from sample by using temperature, humidity and pressure values for the given surrounding.

5.6.1 Drift Compensation

Drift coefficients are indicative of deviation in E-nose response per unit change in the surrounding conditions of temperature, humidity and pressure. Temperature, humidity and pressure are measured continuously in real time and net deviation is calculated in accordance with change in them. Figure 5.11 is shown with the procedure for the drift compensation. Net deviations estimated from drift calculations using drift coefficients are compensated to the E-nose response instantly and automatically by virtue of programming. There are two ways to analyse the response data from the E-nose. In the first method the data set is preprocessed and then presented to the cluster technique and PARC engine for the classification of the data sample. In the second method, after acquiring response, data are compensated in accordance to the temperature, humidity and pressure values depending upon sample and programming algorithms. Compensation is done taking temperature and humidity values into account in case of tea sample data and pressure value in case of kerosene sample. The method has improved cluster classification as shown in Fig. 5.12 and Fig. 5.13, before and after compensation for tea and kerosene samples respectively.

5.6.2 PCA and ANN Analysis

Firstly, experiments were conducted to gather data for ten-tea sample and then PCA and ANN analysis was done for uncompensated data. Uncompensated results are observed. The normalized data were used for PCA and only first three Principal Components were considered for analysis, which resulted in most of the variance. Uncompensated cluster analysis using PCA is shown in Fig. 5.12(a). Secondly, after acquiring data from experiments, compensation is done to remove the effects of temperature, humidity and pressure variations, if any. It is observed that after compensation, the clusters are better grouped as shown in the Fig. 5.12(b). For tea samples, temperature and humidity compensation was applied simultaneously and pressure compensation was applies only for the kerosene sample. Analysis for the kerosene is performed separately. Statistical properties of data sets are shown in Table 5.7. It can be observed that standard deviation is much lower after compensation.

Neural networks were given training with data and then tested for robustness. MLP, LVQ, PNN and RBF networks were used for classifications in the similar way as described for tea analysis in chapter-4. Structures of these nets are given in Table 5.9. The statistical results of PCA before and after compensation due to temperature and humidity variations are shown in the Table 5.8.

Table 5.7: Statistical properties of E-nose response data before and after compensation due temperature changes in the tea sample.

Quantity	Before compensation				After compensation			
	Sen-1	Sen-2	Sen-3	Sen-4	Sen-1	Sen-2	Sen-3	Sen-4
Minimum	0.2689	0.3738	0.3475	0.3249	0.2689	0.3738	0.3475	0.3249
Maximum	0.2950	0.4206	0.5626	0.3898	0.2736	0.3849	0.3921	0.3399
Mean	0.2765	0.3890	0.4134	0.3458	0.2701	0.3773	0.3601	0.3296
Median	0.2736	0.3849	0.3921	0.3399	0.2694	0.3763	0.3547	0.3281
Std	0.0078	0.0140	0.06446	0.01941	0.001353	0.003308	0.01327	0.00450
Range	0.0261	0.0468	0.215	0.06486	0.004712	0.01104	0.04461	0.01502

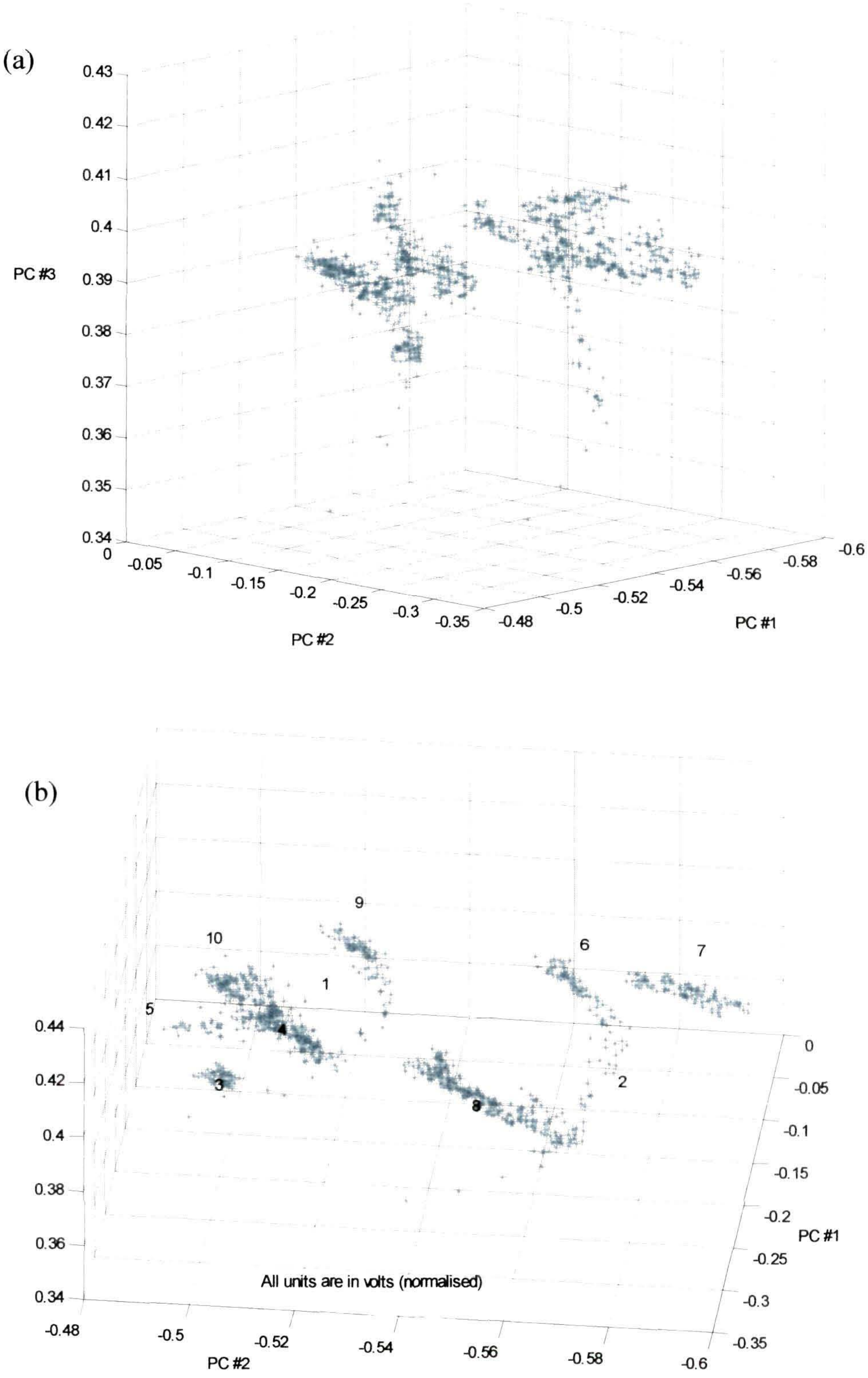


Figure 5.12: PCA cluster display for flavour classification of tea samples, (a) before and (b) after drift compensation.

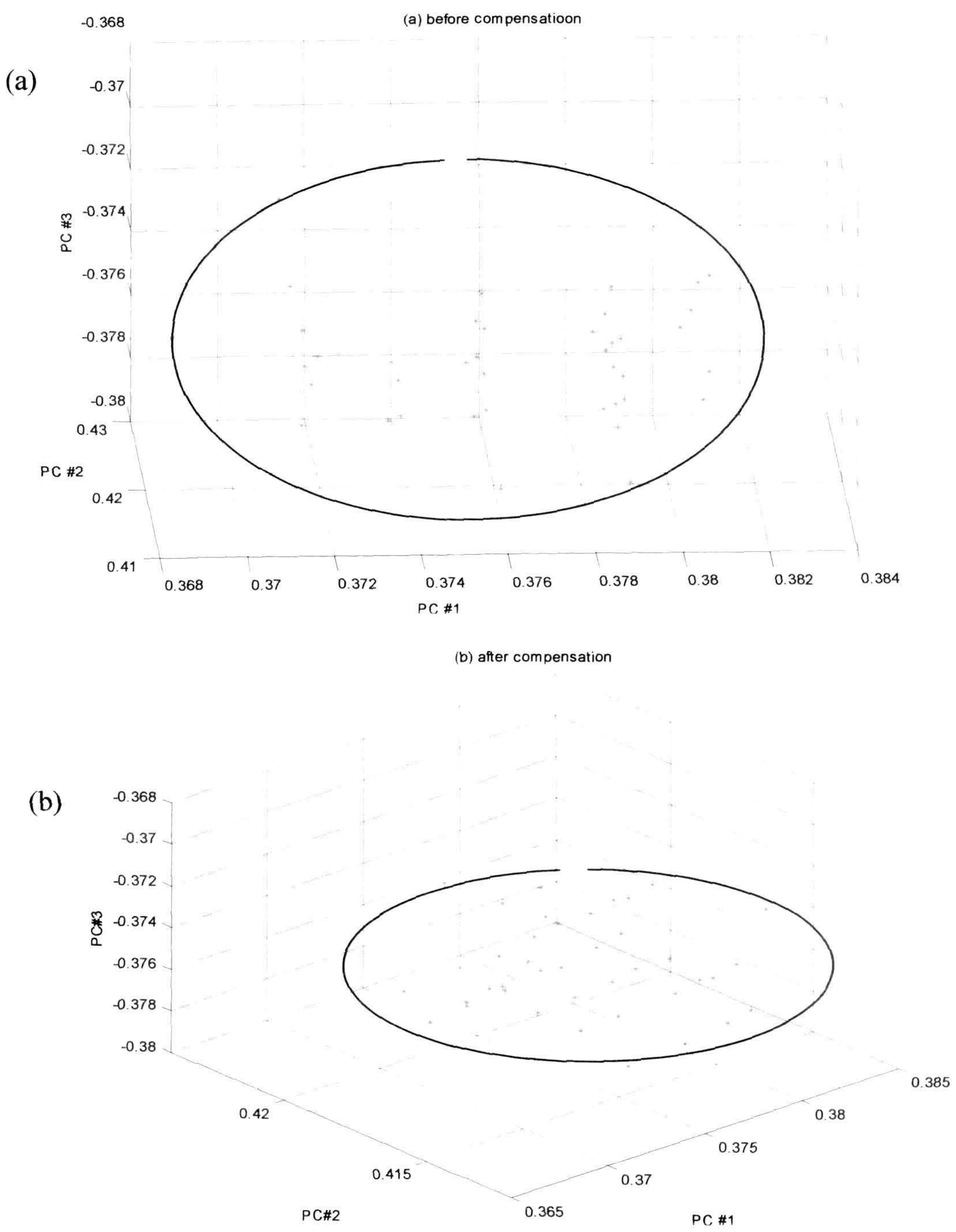


Figure 5.13: PCA cluster analysis display for kerosene sample classification (a) before and (b) after drift compensation. Cluster is more concentrated after compensation.

5.6.3 Results and Discussions

Experiments are conducted by varying temperature, humidity and pressure ranges in samples of tea and kerosene. Subsequently the drifts are eliminated during online capturing, processing and analyzing the data for classification. Table 5.9 is listed with the architecture of different ANN techniques and their classification rates for tea and kerosene samples before and after drift compensations. RBF net has classified samples 97 % of accuracy in case of the tea samples. It can be observed from the Table 5.9 that drift compensation has improved results about 4 to 5%, which is significant.

Table 5.8: The results of PCA analysis for tea flavour quality

PC	% Variance	Eigen-value	Principal Components			
			Sensor$_1$	Sensor$_2$	Sensor$_3$	Sensor$_4$
PC$_1$	75.0797	0.0030	-0.1727	0.1930	0.8982	-0.3552
PC$_2$	22.3322	0.0009	0.3906	-0.7109	-0.0034	-0.5849
PC$_3$	1.7584	0.0001	-0.9005	-0.3955	-0.1356	-0.1199
PC$_4$	0.8298	0.0000	0.0822	-0.5486	0.4182	0.7193

Table 5.9: Architecture of different Neural Networks and correct classification rates in % for tea flavour before and after temperature and humidity compensation.

Neural Network	Architecture	Classification (%) Before Compensation	Classification (%) After Compensation
MLP	4 input Neurons, 5 hidden Neuron, 10 & 5 output Neurons, 0.35 Learning rate with Momentum 0.42	87	92
LVQ	4 input Neurons, variable number of nodes in the competitive layer, 10 & 5 output Neurons, Learning rate 0.011 and Conscious factor 1	92	96
PNN	4 input Neurons, 10 & 5 Neurons in output layer, neurons added until sum-squared error falls below 0.000001, competitive output layer, Spread constant 0.8	93	95
RBF	4 input Neurons, 10 & 5 Neurons in output layer, neurons added until sum- squared error falls below 0.000001, hidden radial basis layer and linear output layer, Spread constant 0 8	96	97

5.7 Conclusion

In this chapter, a method for determining the drift coefficients for temperature, humidity and pressure variations in samples of tea and kerosene is presented. Subsequently drifts were eliminated during the analysis and processing of the data for the classification of the samples of tea and kerosene. The corrective coefficients, as found by a series of experiments, have calibrated E-nose sensors according to ambient conditions (i.e. temperature, humidity and pressure) of samples during performance of experiments. The corrective factors

The results of drift determination experiments are encouraging. There is a strong correlation to temperature, humidity and pressure drifts with deteriorating classification results of the samples used for classification by E-nose system. Temperature and humidity coefficients were determined for the samples of tea. Another important area of application of electronic nose is leakage detection in pressurized systems, which are always susceptible to fire hazards due to leakage of various hydrocarbons, normally at higher pressure. Keeping that in mind, the pressure drift coefficients were determined for the kerosene vapour samples at higher pressure.

It is assumed that E-nose sensors will eventually increase the precision in tea and spice-testing processes that are presently done manually or by PC based Techniques [23, 24, 25]. In such technological advances it becomes necessary to measure the precision of the sensors used for odour classification. It has been tried to measure precision in E-nose sensors for tea and kerosene vapour by determining drift coefficients. It is concluded that three coefficients are expected to be useful for the analysis and quality determination for the samples of tea and kerosene gas. The same concept can also be applied to other sample-sensor combination for the calibration such as Conducting Polymers.

E-nose and ANN techniques with compensations as described have the possibility of improving existing analytical, PC based, profiling panel methods and Image Processing methods for sample classification.

References

[1] Sankar K. Pal and Sushmita Mitra. *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing,* Wiley Series on Intelligent Systems, John Wiley & Sons, Inc.

[2] Susanne Holmin, Christina Krantz-Rulcker, Ingemar Lundstrom and Fredrik Winquist *Drift correction of electronic tongue responses,* Measurement. Science and Technology 12 (2001) 1348–1354.

[3] J. W. Gardner and P. N. Bartlett *A brief history of electronic nose* Sensors and Actuators B, 1994, 18-19, 211-220.

[4] J. W. Gardner, E. L. Hines and M. Wilkinson, *Application of artificial neural networks to an electronic olfactory system,* Measurement Science and Technology, 1990, 1, 446-451

[5] C.D. Natale, E. Martinelli, A. D'Amico, *Counteraction of environmental disturbances of electronic nose data by independent component analysis*, Sens. Actuators B 82 (2002) 158–165.

[6] C. Distante, T. Artursson, P. Siciliano, M. Holmberg, I. Lundström, *Odour identification under drift effect*, electronic noses and olfaction 2000, Institute of Physics Publishing (IoP), Philadelphia, 2000.

[7] M. Holmberg, F.A.M. Davide, C.D. Natale, A. D'Amico, F. Winquist, I. Lundström, *Drift counteraction in odour recognition applications: lifelong calibration method,* Sens. Actuators B 42 (1997) 185–194.

[8] Fryder M, Holmberg M, Winquist F and Lundstr¨om I 1995 A calibration technique for an electronic nose *Transducers'95 and Eurosensors IX (Stockholm)* pp 683–6

[9] Haugen J-E, Tomic O and Kvaal K 2000 A calibration method for handling the temporal drift of solid state gas sensors *Anal. Chim. Acta* **407** 23–9

[10] Artursson T, Ekl¨ov T, Lundstrom I, Mårtensson P, Sjostrom M and Holmberg M 2000 Drift correction for gas sensors using multivariate methods *J. Chemom.* **14** 711–23

[11] Salit M and Turk G 1998 A drift correction procedure *Anal. Chem.* **70** 3184–90

[12] Holmberg M, Davide F, Di Natale C, D'amico A, Winquist F and Lundstrom I 1997 Drift counteraction in odour recognition applications: lifelong calibration method *Sensors Actuators* B **42** 185–94

[13] K. R. Kashwan, M. Bhuyan, *Robust Electronic-Nose System with Temperature and Humidity Drift Compensation for Tea and Spice Flavour Discrimination,* IEEE Conference, AsiaSense-2005, IEEE Catalog: 05EX1161, ISBN: 0-78003-9370-8, Asian Conference on Sensors And International Conference on New Techniques in Pharmaceutical and Biomedical Research, 5 - 7 Sep, 2005, Kuala Lumpur, Malaysia.

[14] S.M. Sze, Physics of Semiconductor Devices, Wiley, New York, 1981.

[15] T. Hattori (Ed.), Ultraclean Surface Processing of Silicon Wafers, Secrets of VLSI Manufacturing, Springer, Berlin, 1998.

[16] T. Doll, K. Scharnagl, R. Winter, M. Bogner, I. Eisele, B. Ostrik, M. Schoning, Work function gas sensors—reference layers and signal analysis, in: Proceedings of the 12th European Conference on Solid- State Transducers, Southampton, UK, September 1998.

[17] M. Burgmair, J. Wollenstein, H. Bottner, A. Karthigeyan, K. Anothainart, I. Eisele, Ti-substituted chromium oxide in work function type sensors: ammonia detection at room temperature with low humidity cross sensitivity, IEEE Sens. J., submitted for publication.

[18] D.E. Williams, in: P.T. Moseley, B.C. Tofield Eds. , *Solid State Gas Sensors,* Chaps. 5.1, Adam Hilger IOP, Publishing, 1987, 71.

[19] 2 K. Ihokura, J. Watson, *The Stanic Oxide Gas Sensor,* CRC Press, Boca Raton, FL, 1994.

[20] 3 N. Barsan, R. Ionescu, *The mechanism of interaction between CO and SnO surface the role of water vapour,* Sensors and Actuators, B 12 (1993), 71–75.

[21] R. Ionescu, *Ageing and p-type conduction in SnO_2 gas sensors,* National Institute of Materials Physics, Bucharest, Magurele RO 76900 Romania, 1998

[22] R. Ionescu, A. Vancu, C. Moise, A. Tomescu. *Role of water vapour in the interaction of SnO_2 gas sensors with CO and CH4*, National Institute of Materials Physics, P.O. Box MG-7, 76900 Bucuresti, Magurele, Romania, July 1999

[23] M Bhuyan and S Borah, *Use of Electronic Nose in Tea Industry,* EAIT, 2001, IIT, Kharagpore.

[24] S Borah and M Bhuyan *Non-destructive testing of tea fermentation using image processing* Insight (Journal of British Institute of Non-destructive Testing) Vol. 45, No. 1, January 2003.

[25] M Bhuyan, *An Integrated PC based Tea Process Monitoring and control System* Ph.D. Thesis, Guwahati University, October 1997

§§

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Chapter 6

Conclusion and Future Scope

This thesis is a descriptive and analytical report on a number of experiments in the field of tea and spice aroma discrimination using a novel technique of Artificial Neural Network based Electronic-Nose. A smart and intelligent E-nose system is developed and applied for prediction of tea flavours and discrimination of spice aromas.

This research can be divided into five broader areas in a field of flavour discrimination and classification for tea and spice samples. These areas are (i) review of literature of state of art of E-nose technology, its applications and ANN classifiers (ii) development of an E-nose system based on MOS sensor technology (iii) classification and discrimination of tea and spice flavours using E-nose setup and

ANN classifiers (iv) calibration of E-nose by determining drift coefficients under varying conditions of temperature, humidity and pressure and (v) optimum sensor selections criteria by Regression methods and RMSE techniques.

Chapter-1 and Chapter-2 are described with introduction to a comprehensive literature as basic knowledge for the research work. The E-nose technology and biological olfaction system are described in parallel. A detailed review of ANN, Artificial Intelligence and Fuzzy System is described. The E-nose is described in detail and concluded that it is an intelligent instrumentation system since it can mimic function of human nose. The database of odour is similar to the database of image processing and feature selection problem. The Artificial Neural Networks (derived from the human brain anatomy) have already been used for number of applications in many fields. It was an obvious choice to used ANN classifiers for the flavour classification since ANN functions are analogous to human brain functions.

At present the E-nose is limited in the efficiency by the type and number of sensors used in the setup. E-noses are preferably designed for specific purpose rather than generalizing these like biological nose comprising of billions of epithelium cells. E-noses developed for specific applications are more conveniently practical and easy to use. Nano technology may in near future make it possible to pack millions of sensors in a single chip and render E-nose somewhat closure to the human nose. The lifeline of the E-nose is pattern recognition software capable of performing huge computational work at very fast speed. This implies that E-nose is a kind of intelligent instrumentation system that is used to sense odourant molecules in an analogous manner of the human nose.

Chapter-3 is described with the procedures of design and development of E-nose setup. E-nose is represented as an intelligent instrumentation system with procedural description of design and development including materials, Printed Circuit Boards (PCBs) and PC interfacing. E-nose data handling and data acquisition are described including feature selection and graphical user interface (GUI) procedure.

The development of the E-nose setup was based on the four available MOS based gas sensors, TGS-2611, TGS-842, TGS-822, and TGS-813-J01 from FIGARO INC, Japan [10]. These sensors were selected on the basis of sensitivity to different odours such as cooking vapors, alcohol, volatile organic compounds (VOCs) etc. A computer controlled switching and diaphragm flow systems were designed. Also an intractable

GUI in GENIE was programmed. Sensor circuits were optimized using PCB technology subjected to the power limitations. The circuits were fabricated manually based on the fact that E-nose response was reasonably high enough leading to the conclusion that noise effects were negligible. Data acquisition and online reading of the response from E-nose sensors was achieved by high performance data Acquisition card PCL-208 from Dynalog (India), Ltd, Mumbai-400086, India.

Chapter-4 is a description of tea and spice flavour discrimination. Sample collection, testing procedure, analysis of data collected from E-nose sensors using ANN techniques are described illustriously. Using additional flavour samples of common spices checked the robustness of the system.

In the first part of research experiments the flavours of five overlapping tea samples were discriminated. Five tea flavours were identified as flavour clusters with the help of PCA, FCM and SOM of the sensor responses. The E-nose system has successfully discriminated the flavours of tea samples manufactured under nonstandard processing conditions, viz. over-fermented, over-fired, under-fermented with the help of linear data processing technique of PCA. Then MLP, LVQ, RBF and PNN network algorithms were applied to the response data for classification of the five overlapping tea samples. Finally, the training time of RBF and PNN were found to be faster than MLP and LVQ. It was also found that the networks had a good performance reaching 100% accuracy on the classification rates of patterns belonging to previously trained samples. If a new and untrained tea sample were offered during testing, RBF, PNN and LVQ associated this pattern to a tea sample which was the nearest to the actual tea sample already known. These characteristics termed the RBF and PNN network superior in the pattern classification in the context of intelligent instruments. The two major drift effects due to environmental temperature and humidity variations were considered. Variations of temperature and humidity were acquired online as two additional parameters for the drift analysis. To consider temperature and humidity drift effects on the sensors' responses data, two extra input nodes were added to the ANN paradigms. This resulted in faster and more accurate training of the ANNs, which increased classification rate RBF, PNN and LVQ were able to classify most of the patterns corresponding the tea samples, MLP was less able to adapt to the uneven distribution of samples. RBF and PNN were able to adjust their network weights of generalization to match the morphological variability of the patterns. They produced better results and training performances than MLP in the discrimination of the tea sample flavours. ANN and MOS based E-nose system had provided high discrimination rate for the five flavours of the overlapping tea samples.

In the second part of the experiments ten standard tea flavours are discriminated using an E-nose and PCA cluster analysis technique and supervised ANN techniques and hence to find out the possibility of improving existing analytical, PC based and profiling panel methods.

This result of non-overlapping classification of tea flavours using PCA cluster technique was encouraging. The E-nose sensors gathered sample data patterns as response from tea samples. The tea flavours were classified with the help of PCA, MLP, LVQ, RBF and PNN. From the results, it was evident that metal oxide sensor based E-nose was capable of discriminating the flavours of tea samples which were analysed by linear and non-linear data processing ANN techniques. An accuracy of 96% was reached in the classification using RBF network compared with 93% using a PNN. It was noticed that the performances were comparable with those achieved with trained MLP (87%) and LVQ (92%). We observed that RBF and PNN were much faster compared to MLP and LVQ nets.

It is concluded that the classification techniques suggested for tea flavour determination using E-nose and Artificial Neural Network will be useful for Tea Industry to ascertain quality and flavour and to standardise the same for Tea quality determination. It will make possible for an improved testing procedure compared to old techniques to determine flavour of tea grades. Implementation of automation and intelligent system will enable the Tea Industry to be independent of human constraints and limitations.

Chapter-5 describes a detailed procedure of determination of the drift parameters. Causes and remedial actions of drifts due to temperature, humidity and pressure are described. Validation of the drift compensation techniques is examined.

In this research an effort has been made to determine suitable coefficients for temperature, humidity and pressure variations in samples and subsequently to eliminate them during online capturing and processing of data for quality and flavour determination of samples in field experimentation. The corrective coefficients, as found by experimental methods, can be applied for calibration of electronic nose instrumentation system according to ambient conditions (i.e. temperature, humidity and pressure) of samples. The corrective factors as determined, may also be incorporated onboard for the hand held electronic nose instruments, as custom based menu or corrections may be applied automatically in more advanced intelligent systems.

The result of drift determination experiments is encouraging because there was a strong linear correlation with temperature, humidity and pressure of the samples used for classification by electronic nose system. Temperature and humidity coefficients were determined for a specific sample of tea. Another important area of application of electronic nose is leakage detection in pressurized systems, which are always susceptible to fire hazards due to leakage of various hydrocarbons, normally at higher pressure. Pressure coefficients are determined for the kerosene vapour as sample at higher pressure.

E-nose sensors were tested for the VOCs of dry tea powder over a temperature ranging from 25^o C (room temperature) to 110^o C. The variation in sensor response due to temperature change in tea sample does not result in a measurable form below the temperature of 30^o C, however on increasing temperature further, it was observed that output response increases up to a temperature about 80^o C. Beyond the temperature of 80^o C, sensor response does not vary significantly. Similarly experiments were conducted for humidity variations from 90% to 40% in tea samples. The sensor response decreased as the humidity decreased. Finally, for pressure variation in kerosene vapour it has been verified that increase in pressure resulted in increased output response in sensors.

The E-nose sensors will eventually increase the precision in Tea testing process that is presently done manually. In age of technological advances it becomes necessary to incorporate precision by the use of E-nose sensors for odour detection. The precision of E-nose sensors was improved by incorporating drift compensation for tea and kerosene vapour. It is concluded that the three coefficients are expected to be useful for the analysis and quality determination of tea samples and leakage detection of kerosene gas. The same concept can be applied to calibrate the other sample-sensor combinations such as Conducting Polymers.

Lastly a method called *regression* is analysed for optimum number of sensors' selection from a group of available sensors. Here, the problem is defined as to find out a combination of sensors, which produce the best results under a given circumstances.

A smaller number of sensors are, of course, desirable from the cost point of view and also for easier and quicker trainings on ANN analysis. The regression technique is based on the principle that a combination of less number of significant sensors can do a classification at better rates compared to the large number of sensors. The 'regression model' is based on the root-mean square error (RMSE) from a multiple linear regression (MLR) criterion. The selection criterion was analysed and a sub group of sensors was identified. A new PCA scores were plotted using the responses from a combination of different sensors.

In the conclusion, a novel intelligent E-nose System has been developed and the classification of fifteen different flavours of tea in overlapping and non-overlapping flavour profile is tested with intelligent PARC techniques. The robustness to non-overlapping characteristics of flavour is justified by testing with five distinct spice flavors. Economical sensor combination selection by RMSE measure is adopted. Additionally to increase the precision, coefficients of drift due to temperature, humidity and pressure variation in samples were determined.

6.1 Future Scope

A quite accurate and capable E-nose system is developed during this research, nevertheless, further work is needed to assess the long-term reliability of the EN system. The use of a commercial headspace auto sampler can be very useful with the sampling process and thus system performance can be improved. This will result in a system that is more suitable for commercial applications for continuous monitoring of tea sample and its optimum processing in manufacturing.

Discrimination of Tea Flavour Specific to clone varieties and zonal varieties is a new field to be explored. FPGA implementation for hardware based classification and discrimination techniques for E-nose and online Tea Flavour Discrimination also seems to be a potential study area. The Nano-technology can ultimately bring E-nose much closer to the biological nose in efficiency and functioning by packing sensors in large number on single chip.

E-nose will have a very crucial role in the near future in clinical and pathological diagnosis applications. This technology is tipped to be alternative to the costly diagnosis method for many diseases at present. E-nose systems can be developed for cheap and fast result production in the field of diagnosis. It is very certain that E-nose will make treatments affordable to the common people by early detection of ailments and also by enhancing Tele-surgery system.

In the conclusion, E-nose has a potential area for further research and study. It is a very wide and complex field involving a huge computational mathematics and database processing to name a few. It has high challenges to offer for further explorer.

§§

APPENDICES

Appendix 1

Glossary

A1.1 Tea Qualities and allied terms

Aroma - Denotes that both the tealeaf and liquor have at least one of a certain number of smells those are desirable and are highly valued. Such aroma is connected with flavour and is often highly fragrant. Tea contains about 300 components many of which have desirable aromas.

Bakey - An aroma of high-fired teas, or tea from which too much moisture has been driven off.

Banking - In tea tasting the retention on the drained leaves of considerable liquor, which may only be squeezed out.

Basket Fired - Japanese tea which has been dried by firing in a basket.

Biscuity - A pleasant aroma occasionally smelt in the leaf or liquor of well-fired Assam tea, often associated with malty.

Black Tea - Any tea that has been thoroughly fermented before being fired.

Blend - A mixture of different growths of tea or different types of tea.

Blistered - Leaf, which is swollen and hollow inside. Blisters are formed during the firing of leaf, which has been dried too quickly.

Body - liquor having both fullness and strength as opposed to a thin liquor.

Bold - Large leaf, which could to advantage have been cut smaller.

Bright - Sparkling clear liquor. Denotes a good tea, which has life as opposed to a dull looking leaf or liquor.

Brisk - Pleasantly astringent, not flat liquor. Usually it is of pungent character.

Broken - Leaf that has been broken into smaller pieces by rolling or passing through a cutter.

Broker - A person who negotiates the sale of tea from one dealer to another and takes a brokerage commission from the seller.

Brownish - Leaf which is brown in colour rather than black, generally resulting from firing under-withered tea at too high a temperature or due to poor plucking. However, some tippy teas have a brown leaf, which is desirable.

Burnt - An unpleasant taste of burnt organic matter in the liquor and a similar smell in the infused leaf is a characteristic of teas that have been fired at too high a temperature.

Caffeine
The stimulating constituent in tea, bitter in taste but cannot be detected at the normal drinking strength.

Ceylons
The old name for Sri Lankan teas. Both black and green teas are made in Sri Lanka, but the blacks predominate. They are known by the name of the district, and are further identified by garden marks.

Character - When tasting teas, a desirable liquor quality that permits identification of country of origin and district within that country.

Choppy - A term used to describe leaf chopped after processing in the breaker or cutter rather than in the roller. Often used in regard to broken pekoe made by cutting a pekoe or orange pekoe.

Chunky - Term usually applied to broken types that are too large in size.

Congou - A general term used to describe all China blacks, irrespective of district.

Coarse - Tea liquor with certain undesirable characteristics resulting from coarse plucked leaf or irregular firing.

Coppery - Bright infusion the colour of a 'new penny', from good quality well manufactured black tea.

Creaming Down - A milky film rising to the surface of the tasting cup as the liquor cools, accompanied by the thickening of the liquor in certain high grade teas; believed to be caused by precipitation of the polyphenols and caffeine.

Creepy - A cutting process (cut, tear, curl), used to make small leaf grades required for teabags. Produces coloury fast liquoring teas.

Darjeeling -Considered to be the finest and most delicately flavoured of the Indian teas. Grown mainly in the high altitude region of the Himalaya mountains ranging from 2,000 feet to 6,000 feet. The famed 'muscatel' flavour is often elusive but is perfection when present.

Dark - Colour of liquor denoting a poor tea.

Denaturised Tea - Tea which has been deemed unfit for consumption. It is often used for mulching tea bushes.

Dull -Tea liquor that is not clear and bright.

Dust - The smallest siftings resulting from the sieving process or leaf that has been reduced to a fine powder. Good quality dust gives the strongest tea, best colour and quickest infusion.

Earthy - An unpleasant liquor taste found in tea stored under damp conditions.

Estate - A property or holding, which may comprise more than one garden or plantation under the same management or ownership.

Fannings - Small grainy particles of leaf sifted out of the graded tea. Sought after for tea bags as they give a good quick infusion.

Fermenting - Black tea is fermented (oxidised) due to the action of its own enzymes. Fermenting causes green polyphenols (catechins) to change to orange red theaflavins and thearubigins; fermenting takes 60-180 minutes.

First Flush - The first growth of the season often the very best quality due to its relatively slow growth. 'Second Flush' is usually more plentiful, more consistent, but sometimes not such good flavour.

Flat - Lacking briskness and pungency.

Fluff - Thick hairy down or bloom on the tealeaf, which becomes loosened in the blending and sifting process. It is collected and used in the manufacture of caffeine and instant tea.

Flush - The new shoot of a tea bush consisting of two leaves and a bud as plucked for quality teas. It takes an average of 40 days for a leaf bud to develop into a shoot containing a full complement of leaves, which is known as a flush.

Full -Strong tea, without bitterness, having colour and substance.

Fully Fired - Liquor from a tea that has been slightly over fired. See also 'high fired'.

Garden - Used interchangeably with 'plantation' in some tea countries (India and Sri Lanka particularly), but usually referring to an estate unit.

Golden Tip - Golden coloured tip. Visible buds in the made tea, coloured golden with dried tea juice.

Gone Off - A tea that is moulded, tainted, out of condition or old.

Grainy - Term applied to well make fannings and dust.

Green - Colour of infused leaf, which has undergone poor withering or rolling, or is underfermented.

Green Tea - Tealeaves that have been processed either in live steam, hot air or hot pans, whereby fermentation is prevented, and then rolled and dried.

Grey - An unattractive colour characteristic of black leaf that has undergone too much rubbing during sorting and cutting.

Gunpowder - A make of green tea, each leaf of which has been rolled into a pellet. The pellets resemble old fashioned gunpowder cartridges.

Heavy - Liquor that is thick, strong, coloury with only a little briskness.

High Fired - A slightly burnt tea, but not so badly fired as to be called burnt. Results from keeping tea in the drier too long or at too high a temperature.

Ichiban-cha - Japanese meaning first tea or first plucking.

Imperial - A type of rolled Ceylon, black tea.

Infusion - Properly speaking the solid leaf that is left after decanting the liquor. The colour and form of the infusion can reveal much about the manufacturing conditions.

Jasmine - A mild, delicately flavoured China tea that is scented after firing with white jasmine flowers.

Java - Teas grown in the island of Java now known as Indonesia. They are manufactured as in Ceylon and India, and are almost entirely of the black variety.

Keemun - A fine grade of black tea from central China, often imitated with considerable success. Keemun is a fine quality China Black, hand rolled and basket fired in contrast to the common types that are machine rolled and machine fired.

Lapsang Souchong - A fine grade of China black tea with a smokey or tarry flavour. Purposely introduced by firing over pine needles.

Leafy - Whole leaves found in broken grades.

Lie Tea - A Chinese mixture of willow and other spurious leaf with genuine tealeaf, fraudulently sold as tea. Adulteration of tea was a problem when it was heavily taxed in previous centuries.

Light - Liquor lacking body or thickness.

Liquor - The liquid that results from infusing the leaves with hot water.

Malty - A pleasent taste in the liquor, associated with Assams.

Mature - No flatness or rawness in the liquor.

Metallic - A harsh taste characteristic of some liquor. A metallic taste is often associated with poor pluck standards.

Muddy - Term used to describe dull liquor.

Mushy - A soft tea suggesting that it had been packed too moist.

Musty - Tea that has been attacked by mildew as a result of being packed too moist.

New - Term used to describe a tea, which has not had time to mature. Usually denotes some rawness in the infusion, which may disappear when the tea is kept.

Niban-cha - Japanese for second tea, or second plucking.

Nilgiri - A tea district in Southern India. The name means 'Blue Mountain.'

Nose - Tea tasters name for the aroma of tea

Oolong Tea - From the Chinese wu-lung, meaning 'black dragon'. A semi-fermented tea of fine quality, traditionally hand rolled and fired in baskets over pits containing red-hot charcoal. Originally from China, now also from Taiwan (Formosa).

Orthodox - The traditional rolling table method of tea production that maximizes large leaf grades and gives good aroma and flavour.

Pan Fired - A kind of green tea that is dried in iron pans over charcoal fires.

Pingsuey - A type of China green tea. The word 'pingsuey' actually means 'ice water'.

Pinheads - Small shotty gunpowder green teas.

Plain - Term used to describe dull liquor often with a rather sour taste.

Point - Attractive brightness and acidity of liquor.

Pungent - Pleasantly astringent in the mouth.

Ragged - Denotes uneven leaf in a grade.

Rasping - Coarse flavour in the cup. Also called 'harsh'.

Rawness - Harsh and bitter taste of immature tea.

Sambancha - Japanese for third tea, or third plucking.

Sappy - Full juicy liquor.

San-chu - Chinese for third spring, applied to the third plucking.

Scented Tea - Flower aroma teas traditionally made in China and Taiwan by introducing jasmine, gardenia, lychee or yulan blossoms during the packing process.

Sencha - Name given to the ordinary everyday teas of Japan. Exported Japanese teas are often Sencha types.

Silver Tip - Silver instead of golden tip. Visible buds in the made tea, coloured silver.

Smokey - Term used to describe a manufacturing defect i.e. liquor which tastes of smoke. Smokey can also refer to a tarry taste purposely introduced into 'Lapsang Souchong' and 'Caravan' blends.

Stalky - Term used to describe teas with stalk in them. Usually indicates presence of red stalk resulting from rolling that was too hard, or due to coarse leaf. A certain amount of stalk is removed by skilful sorting during manufacture.

Stewy - Soft liquor, lacking point. Often due to too long a fermentation, or to drying at too low a temperature.

Strength - Thick liquor, pungent and brisk.

Sumatra - Tea grown on the island of Sumatra. Grading and characteristics are similar to those of Java tea.

Sweet - A light and not undesirable characteristics in liquor. Nilgiri teas are particularly 'sweet'.

Tablet tea - Also 'brick tea'. Small compressed blocks made of fine tea dust of special quality in China. Were once used as a form of currency.

Tainted - Term used to describe tea with a strange flavour. May result from infection by microorganisms during manufacture or storage. Usually refers to a flavour entirely foreign to tea such as oil, petrol, onion etc.

Tarry - A smokey aroma or taste. See 'Lapsang Souchong'.

Tea - The tender leaves, and bud of the plant Camellia Sinesis, prepared and cured by recognized methods of manufacture. Used to prepare an infusion for drinking. Term also used with reference to the liquid drink infused from these leaves.

Tea Taster - An expert judge of the cup quality and leaf of teas. One who judges tea by tasting in the cup using organoleptic skills with special reference to sight, taste and smell.

Thick teas - Indian, Sri Lankan, Indonesian or China congou black teas possessed of full strength.

Tip - The bud leaf of the tea plant. Often covered in fine hairs it varies in colour in the made tea from silver to gold. The tip has the finest flavour.

Tippy Teas - Highly desirable teas abundant with silver or golden tips.

Uneven -Term used to describe tealeaf composed of irregular shaped pieces indicating bad sorting. When applied to the infused leaf the term means that it contains mixed red, green and black colours resulting from uneven withering, fermentation or rolling.

Uva - A high elevation quality tea district in Sri Lanka.

Weak - Denotes thin liquor often due to over withering or under fermenting.

Well Twisted - Leaf that is tightly rolled or twisted, which in orthodox manufacture indicates ideally withered tea.

Wiry - Term applied to well twisted, thin leaf orange pekoe. A good OP has long, very black even sized twisted leaf.

Woody - Denotes an undesirable hay flavour in tea, often due to long storage.

A1.2 Artificial Neural Networks

ADALINE - An acronym for a linear neuron: ADAptive LINear Element.

adaption - A training method that proceeds through the specified sequence of inputs, calculating the output, error and network adjustment for each input vector in the sequence as the inputs are presented.

adaptive learning rate - A learning rate that is adjusted according to an algorithm during training to minimize training time.

adaptive filter - A network that contains delays and whose weights are adjusted after each new input vector is presented. The network "adapts" to changes in the input signal properties if such occur. This kind of filter is used in long distance telephone lines to cancel echoes.

architecture - A description of the number of the layers in a neural network, each layer's transfer function, the number of neurons per layer, and the connections between layers.

backpropagation learning rule - A learning rule in which weights and biases are adjusted by error-derivative (delta) vectors backpropagated through the network. Backpropagation is commonly applied to feedforward multilayer networks. Sometimes this rule is called the generalized delta rule.

backtracking search - Linear search routine that begins with a step multiplier of 1 and then backtracks until an acceptable reduction in the performance is obtained.

batch - A matrix of input (or target) vectors applied to the network "simultaneously." Changes to the network weights and biases are made just once for the entire set of vectors in the input matrix. (This term is being replaced by the more descriptive expression "concurrent vectors.")

batching - The process of presenting a set of input vectors for simultaneous calculation of a matrix of output vectors and/or new weights and biases.

Bayesian framework - Assumes that the weights and biases of the network are random variables with specified distributions.

BFGS quasi-Newton algorithm - A variation of Newton's optimization algorithm, in which an approximation of the Hessian matrix is obtained from gradients computed at each iteration of the algorithm.

bias - A neuron parameter that is summed with the neuron's weighted inputs and passed through the neuron's transfer function to generate the neuron's output.

bias vector - A column vector of bias values for a layer of neurons.

Brent's search - A linear search that is a hybrid combination of the golden section search and a quadratic interpolation.

Charalambous' search - A hybrid line search that uses a cubic interpolation, together with a type of sectioning.

cascade forward network - A layered network in which each layer only receives inputs from previous layers.

classification - An association of an input vector with a particular target vector.

competitive layer - A layer of neurons in which only the neuron with maximum net input has an output of 1 and all other neurons have an output of 0. Neurons compete with each other for the right to respond to a given input vector.

competitive learning - The unsupervised training of a competitive layer with the instar rule or Kohonen rule. Individual neurons learn to become feature detectors. After training, the layer categorizes input vectors among its neurons.

competitive transfer function - Accepts a net input vector for a layer and returns neuron outputs of 0 for all neurons except for the "winner," the neuron associated with the most positive element of the net input n.

concurrent input vectors - Name given to a matrix of input vectors that are to be presented to a network "simultaneously." All the vectors in the matrix will be used in making just one set of changes in the weights and biases.

conjugate gradient algorithm - In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than a search along the steepest descent directions.

connection - A one-way link between neurons in a network.

connection strength - The strength of a link between two neurons in a network. The strength, often called weight, determines the effect that one neuron has on another.

cycle - A single presentation of an input vector, calculation of output, and new weights and biases.

dead neurons - A competitive layer neuron that never won any competition during training and so has not become a useful feature detector. Dead neurons do not respond to any of the training vectors.

decision boundary - A line, determined by the weight and bias vectors, for which the net input *n* is zero.

delta rule - See the Widrow-Hoff learning rule.

delta vector - The delta vector for a layer is the derivative of a network's output error with respect to that layer's net input vector.

distance - The distance between neurons, calculated from their positions with a distance function.

distance function - A particular way of calculating distance, such as the Euclidean distance between two vectors.

early stopping - A technique based on dividing the data into three subsets. The first subset is the training set used for computing the gradient and updating the network weights and biases. The second subset is the validation set. When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. The third subset is the test set. It is used to verify the network design.

epoch - The presentation of the set of training (input and/or target) vectors to a network and the calculation of new weights and biases. Note that training vectors can be presented one at a time or all together in a batch.

error jumping - A sudden increase in a network's sum-squared error during training. This is often due to too large a learning rate.

error ratio - A training parameter used with adaptive learning rate and momentum training of backpropagation networks.

error vector - The difference between a network's output vector in response to an input vector and an associated target output vector.

feedback network - A network with connections from a layer's output to that layer's input. The feedback connection can be direct or pass through several layers.

feedforward network - A layered network in which each layer only receives inputs from previous layers.

Fletcher-Reeves update - A method developed by Fletcher and Reeves for computing a set of conjugate directions. These directions are used as search directions as part of a conjugate gradient optimization procedure.

function approximation - The task performed by a network trained to respond to inputs with an approximation of a desired function.

generalization - An attribute of a network whose output for a new input vector tends to be close to outputs for similar input vectors in its training set.

generalized regression network - Approximates a continuous function to an arbitrary accuracy, given a sufficient number of hidden neurons.

global minimum - The lowest value of a function over the entire range of its input parameters. Gradient descent methods adjust weights and biases in order to find the global minimum of error for a network.

golden section search - A linear search that does not require the calculation of the slope. The interval containing the minimum of the performance is subdivided at each iteration of the search, and one subdivision is eliminated at each iteration.

gradient descent - The process of making changes to weights and biases, where the changes are proportional to the derivatives of network error with respect to those weights and biases. This is done to minimize network error.

hard-limit transfer function - A transfer that maps inputs greater-than or equal-to 0 to 1, and all other values to 0.

Hebb learning rule - Historically the first proposed learning rule for neurons. Weights are adjusted proportional to the product of the outputs of pre- and post-weight neurons.

hidden layer - A layer of a network that is not connected to the network output. (For instance, the first layer of a two-layer feedforward network.)

home neuron - A neuron at the center of a neighborhood.

hybrid bisection-cubicsearch - A line search that combines bisection and cubic interpolation.

input layer - A layer of neurons receiving inputs directly from outside the network.

initialization - The process of setting the network weights and biases to their original values.

input space - The range of all possible input vectors.

input vector - A vector presented to the network.

input weights - The weights connecting network inputs to layers.

input weight vector - The row vector of weights going to a neuron.

Jacobian matrix - Contains the first derivatives of the network errors with respect to the weights and biases.

Kohonen learning rule - A learning rule that trains selected neuron's weight vectors to take on the values of the current input vector.

layer - A group of neurons having connections to the same inputs and sending outputs to the same destinations.

layer diagram - A network architecture figure showing the layers and the weight matrices connecting them. Each layer's transfer function is indicated with a symbol. Sizes of input, output, bias and weight matrices are shown. Individual neurons and connections are not shown. (See Chapter 2.)

layer weights - The weights connecting layers to other layers. Such weights need to have non-zero delays if they form a recurrent connection (i.e., a loop).

learning - The process by which weights and biases are adjusted to achieve some desired network behavior.

learning rate - A training parameter that controls the size of weight and bias changes during learning.

learning rules - Methods of deriving the next changes that might be made in a network OR a procedure for modifying the weights and biases of a network.

Levenberg-Marquardt - An algorithm that trains a neural network 10 to 100 faster than the usual gradient descent backpropagation method. It will always compute the approximate Hessian matrix, which has dimensions *n*-by-*n*.

line search function - Procedure for searching along a given search direction (line) to locate the minimum of the network performance.

linear transfer function - A transfer function that produces its input as its output.

link distance - The number of links, or steps, that must be taken to get to the neuron under consideration.

local minimum - The minimum of a function over a limited range of input values. A local minimum may not be the global minimum.

log-sigmoid transfer function - A squashing function that maps the input to the interval (0,1).

Manhattan distance - The Manhattan distance between two vectors x and y is calculated as:

```
D = sum(abs(x-y))
```

maximum performance increase - The maximum amount by which the performance is allowed to increase in one iteration of the variable learning rate training algorithm.

maximum step size - The maximum step size allowed during a linear search. The magnitude of the weight vector is not allowed to increase by more than this maximum step size in one iteration of a training algorithm.

mean square error function - The performance function that calculates the average squared error between the network outputs a and the target outputs t.

momentum - A technique often used to make it less likely for a backpropagation networks to get caught in a shallow minima.

momentum constant - A training parameter that controls how much "momentum" is used.

mu parameter - The initial value for the scalar μ.

neighborhood - A group of neurons within a specified distance of a particular neuron. The neighborhood is specified by the indices for all of the neurons that lie within a specified radius of the winning neuron.

net input vector - The combination, in a layer, of all the layer's weighted input vectors with its bias.

neuron - The basic processing element of a neural network. Includes weights and bias, a summing junction and an output transfer function. Artificial neurons, such as those simulated and trained with this toolbox, are abstractions of biological neurons.

neuron diagram - A network architecture figure showing the neurons and the weights connecting them. Each neuron's transfer function is indicated with a symbol.

ordering phase - Period of training during which neuron weights are expected to order themselves in the input space consistent with the associated neuron positions.

output layer - A layer whose output is passed to the world outside the network.

output vector - The output of a neural network. Each element of the output vector is the output of a neuron.

output weight vector - The column vector of weights coming from a neuron or input. (See outstar learning rule.)

outstar learning rule - A learning rule that trains a neuron's (or input's) output weight vector to take on the values of the current output vector of the post-weight layer. Changes in the weights are proportional to the neuron's output.

overfitting - A case in which the error on the training set is driven to a very small value, but when new data is presented to the network, the error is large.

pass - Each traverse through all of the training input and target vectors.

pattern - A vector.

pattern association - The task performed by a network trained to respond with the correct output vector for each presented input vector.

pattern recognition - The task performed by a network trained to respond when an input vector close to a learned vector is presented. The network "recognizes" the input as one of the original target vectors.

performance function - Commonly the mean squared error of the network outputs. However, the toolbox also considers other performance functions. Type nnets and look under performance functions.

perceptron - A single-layer network with a hard-limit transfer function. This network is often trained with the perceptron learning rule.

perceptron learning rule - A learning rule for training single-layer hard-limit networks. It is guaranteed to result in a perfectly functioning network in finite time, given that the network is capable of doing so.

performance - The behavior of a network.

Polak-Ribiére update - A method developed by Polak and Ribiére for computing a' set of conjugate directions. These directions are used as search directions as part of a conjugate gradient optimization procedure.

positive linear transfer function - A transfer function that produces an output of zero for negative inputs and an output equal to the input for positive inputs.

postprocessing - Converts normalized outputs back into the same units that were used for the original targets.

Powell-Beale restarts - A method developed by Powell and Beale for computing a set of conjugate directions. These directions are used as search directions as part of a conjugate gradient optimization procedure. This procedure also periodically resets the search direction to the negative of the gradient.

preprocessing - Perform some transformation of the input or target data before it is presented to the neural network.

principal component analysis - Orthogonalize the components of network input vectors. This procedure can also reduce the dimension of the input vectors by eliminating redundant components.

quasi-Newton algorithm - Class of optimization algorithm based on Newton's method. An approximate Hessian matrix is computed at each iteration of the algorithm based on the gradients.

radial basis networks - A neural network that can be designed directly by fitting special response elements where they will do the most good.

radial basis transfer function - The transfer function for a radial basis neuron.

regularization - Involves modifying the performance function, which is normally chosen to be the sum of squares of the network errors on the training set, by adding some fraction of the squares of the network weights.

resilient backpropagation - A training algorithm that eliminates the harmful effect of having a small slope at the extreme ends of the sigmoid "squashing" transfer functions.

saturating linear transfer function - A function that is linear in the interval (-1,+1) and saturates outside this interval to -1 or +1. (The toolbox function is satlin.)

scaled conjugate gradient algorithm - Avoids the time consuming line search of the standard conjugate gradient algorithm.

sequential input vectors - A set of vectors that are to be presented to a network "one after the other." The network weights and biases are adjusted on the presentation of each input vector.

sigma parameter - Determines the change in weight for the calculation of the approximate Hessian matrix in the scaled conjugate gradient algorithm.

sigmoid - Monotonic S-shaped function mapping numbers to a finite interval such as (-1,+1) or (0,1).

simulation - Takes the network input p, and the network object net, and returns the network outputs a.

spread constant - The distance an input vector must be from a neuron's weight vector to produce an output of 0.5.

squashing function - A monotonic increasing function that takes input values and returns values in a finite interval.

star learning rule - A learning rule that trains a neuron's weight vector to take on the values of the current input vector. Changes in the weights are proportional to the neuron's output.

sum-squared error - The sum of squared differences between the network targets and actual outputs for a given input vector or set of vectors.

supervised learning - A learning process in which changes in a network's weights and biases are due to the intervention of any external teacher. The teacher typically provides output targets.

symmetric hard-limit transfer function - A transfer that maps inputs greater-than or equal-to 0 to +1, and all other values to -1.

symmetric saturating linear transfer function - Produces the input as its output as long as the input i in the range -1 to 1. Outside that range the output is -1 and +1 respectively.

tan-sigmoid transfer function - A squashing function that maps the input to the interval (-1,1).

tapped delay line - A sequential set of delays with outputs available at each delay output.

target vector - The desired output vector for a given input vector.

test vectors - A set of input vectors (not used directly in training) that is used to test the trained network.

topology functions - Ways to arrange the neurons in a grid, box, hexagonal, or random topology.

training - A procedure whereby a network is adjusted to do a particular job. Commonly viewed as an "offline" job, as opposed to an adjustment made during each time interval as is done in adaptive training.

training vector - An input and/or target vector used to train a network.

transfer function - The function that maps a neuron's (or layer's) net output **n** to its actual output.

tuning phase - Period of SOFM training during which weights are expected to spread out relatively evenly over the input space while retaining their topological order found during the ordering phase.

underdetermined system - A system that has more variables than constraints.

unsupervised learning - A learning process in which changes in a network's weights and biases are not due to the intervention of any external teacher. Commonly changes are a function of the current network input vectors, output vectors, and previous weights and biases.

update - Make a change in weights and biases. The update can occur after presentation of a single input vector or after accumulating changes over several input vectors.

validation vectors - A set of input vectors (not used directly in training) that is used to monitor training progress so as to keep the network from overfitting.

weighted input vector - The result of applying a weight to a layer's input, whether it is a network input or the output of another layer.

weight function - Weight functions apply weights to an input to get weighted inputs as specified by a particular function.

weight matrix - A matrix containing connection strengths from a layer's inputs to its neurons. The element $w_{i,j}$ of a weight matrix W refers to the connection strength from input j to neuron i.

Widrow-Hoff learning rule - A learning rule used to trained single-layer linear networks. This rule is the predecessor of the backpropagation rule and is sometimes referred to as the delta rule.

A1.3 Fuzzy Logic

adaptive Neuro-Fuzzy Inference System - (ANFIS) A technique for automatically tuning Sugeno-type inference systems based on training data.

aggregation - The combination of the consequents of each rule in a Mamdani fuzzy inference system in preparation for defuzzification.

antecedent - The initial (or "if") part of a fuzzy rule.

consequent - The final (or "then") part of a fuzzy rule.

defuzzification - The process of transforming a fuzzy output of a fuzzy inference system into a crisp output.

degree of fulfillment - See firing strength

degree of membership - The output of a membership function, this value is always limited to between 0 and 1. Also known as a membership value or membership grade.

firing strength - The degree to which the antecedent part of a fuzzy rule is satisfied. The firing strength may be the result of an AND or an OR operation, and it shapes the output function for the rule. Also known as degree of fulfillment.

fuzzification - The process of generating membership values for a fuzzy variable using membership functions.

fuzzy c-means clustering - A data clustering technique wherein each data point belongs to a cluster to a degree specified by a membership grade.

fuzzy inference system (FIS) - The overall name for a system that uses fuzzy reasoning to map an input space to an output space.

fuzzy operators - AND, OR, and NOT operators. These are also known as logical connectives.

fuzzy set - A set that can contain elements with only a partial degree of membership.

fuzzy singleton - A fuzzy set with a membership function that is unity at a one particular point and zero everywhere else.

implication - The process of shaping the fuzzy set in the consequent based on the results of the antecedent in a Mamdani-type FIS.

Mamdani-type inference - A type of fuzzy inference in which the fuzzy sets from the consequent of each rule are combined through the aggregation operator and the resulting fuzzy set is defuzzified to yield the output of the system.

membership function (MF) - A function that specifies the degree to which a given input belongs to a set or is related to a concept.

singleton output function - An output function that is given by a spike at a single number rather than a continuous curve. In the Fuzzy Logic Toolbox it is only supported as part of a zero-order Sugeno model.

subtractive clustering - A technique for automatically generating fuzzy inference systems by detecting clusters in input-output training data.

Sugeno-type inference - A type of fuzzy inference in which the consequent of each rule is a linear combination of the inputs. The output is a weighted linear combination of the consequents.

T-conorm - (also known as S-norm) — A two-input function that describes a superset of fuzzy union (OR) operators, including maximum, algebraic sum, and any of several parameterized T-conorms.

T-norm - A two-input function that describes a superset of fuzzy intersection (AND) operators, including minimum, algebraic product, and any of several parameterized T-norms.

Appendix 2

Tea Flavour Wheel

A.2.1 Tea Flavour Wheel

Tea industries all over the world presently use certain standard terminology of tea flavour, however, there is no mention about a quantitative description or score on these flavour terms. The Tocklai Tea Research Association, Assam, (India) has adopted standard terminology. These tea notes, however, overlap in certain parts. Twenty-five non-overlapping flavour terms have been identified (Bhuyan & Borah, 2001) as shown in the figure A2.1. There are about 40 generally used flavour notes. An E-nose may provide a more objective platform to augment the conventional methods for tea tasting and quality monitoring during production process with the help of these terms on the flavour wheel. The inner part of the wheel has mainly two areas, the first for the taste and the second for the aroma. The taste is more concern with the Electronic-tong and flavour is concern with the E-nose. This research is focused only on the discrimination of the flavour and thus flavour terms are described.

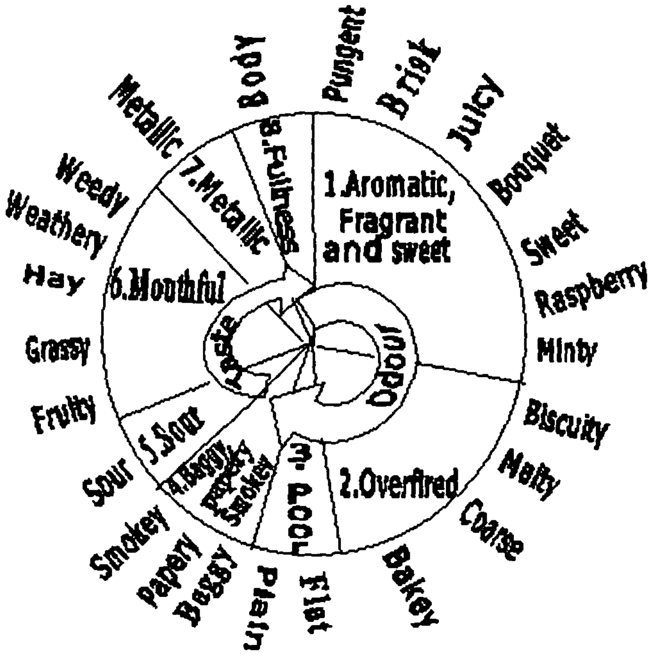


Figure A2.1: Flavour wheel used to illustrate the international flavour terminology for tea. There are about 40 flavour terms out of which only 24 non-overlapping terms have been used.

Appendix 3

Chemical Composition of Tea

The Table A3.1 is explained with the ratios of chemical composition in tea. More than 500 chemical constituents are present in tea. These are flavonoids, amino acids, vitamins, caffeine and polysaccharides. Most of the factories produce orthodox black tea, where the contents of *linalool* and *geraniol* in the black tea are balanced and the aroma is mild and fine. Both volatile and non-volatile compounds are found in tea. The optimum fermentation process of the tea is utmost important for formation of these constituents in good amount. Tea also contains many essential micronutrients, such as vitamins C, B, E and K. The main chemical constituents with their ratios of presence are listed in the Table A3.1. Different varieties of the tea from Sri Lanka, India and Japan are included in Table A3.1

Table A3.1: The main chemical constituents of the tea and their ratios of composition to the total chemical composition present in tea.

Compound	**Sri Lanka**		**India**					**Japan**
			Var. Assamica	**Hybrid of Assamica Sinesis Darjeeling**				**Benih o mare**
	Rt^b	**Uva**	**Assam Dimbula**	**(1)**	**(2)**	**(1)**	**(2)**	
t-2-hexenal	0.40	3.10	2.60	4.90	3.10	0.70	0.30	1.50
cis-3-hexenal	0.53	2.80	4.30	0.20	3.80	1.40	0.10	6.10
t-2-hexenyl formate	0.65	9.50	11.80	11.90	5.00	5.70	3.10	5.20
Linalool oxide (furanoid-cis)	0.77	3.40	3.20	3.50	3.60	8.20	4.70	3.80
Linalool oxide (furanoid-trans)	0.83	10.30	8.80	8.00	12.00	16.70	12.00	12.00
Linalool	1.00	24.00	15.50	18.30	32.80	15.60	13.70	9.30
Phenylacetaldehyde	1.20	0.20	0.50	4.00	5.00	1.10	1.80	1.00
Linalool oxide pyranoid-cis	1.40	0.30	0.40	0.40	Trace	Trace	1.00	6.00
Methysalcylate	1.50	18.60	18.80	9.00	13.2	9.80	5.30	4.90
Geraniol	1.67	1.3	2.20	3.30	1.60	7.30	15.90	21.7
Benzylalcohol	1.71	1.00	1.90	4.30	1.00	1.70	2.00	2.60
2-phenylethanol	1.78	0.20	0.90	4.30	1.00	2.00	6.70	7.50
Cis-jasmone+β-ionone	1.83	0.20	0.10	7.4	1.50	0.50	4.40	0.30

Appendix 4

E-nose Manufacturing List

A list of the leading manufacturers of the sensors used in the odour detection is enclosed in this appendix for the future reference. There are number of manufacturers for these sensors for wide range of odours. Selections of the sensors for application area demands knowledge of characteristics responses and technology used. Manufacturers normally enlist the application areas, limitations of sensors and user instructions.

Table A4.1: Leading Manufacturers of E-nose sensors

Sl. No.	Manufacturer	Website / Product	Technology	Applications
1.	Agilent Technologies	www.agilent.com HP 440A	MS or GC/MS	▪ Compound libraries ▪ Natural & fuel gases ▪ Toxicology ▪ Medical
2.	Airsense Analytics	www.airsense.com PEN, i-PEN	MOS and MS	▪ Food ▪ Materials ▪ Environment & Safety ▪ Medicine
3.	Alpha MOS	www.alpha-moscom Fox 5000, αPrometheus	Sensor array (MOS, CP, QMB) and MS	▪ Food ▪ Cosmetic ▪ Packaging ▪ Process / quality control
4.	Applied Sensors	www.appliedsensor.com VOCcheck / meter, 3320	Field Effect MOS, MOS, QMB	▪ Food quality & control ▪ Chemical and biochemical ▪ Solvents, leaks & off-odours
5.	Bloodhound Sensors Ltd	www.bloodhound.co.ukBloodhound	CP	▪ Food Processing ▪ Microbiology ▪ Environment ▪ Fragrances
6.	Cyano Sciences Inc.	www.cyanoscience.com Cyranose 320	CP	▪ Food quality ▪ Medical ▪ Process Control ▪ Chemical and biochemical ▪ Environmental Monitoring
7.	Environics Oy	www.environics.fi ChemPro, M90, MGD-1	Ion mobility cell	▪ Military warfare ▪ Industry chemicals ▪ Space systems
8.	Electronics Sensor Technology	www.estcal.com zNose moel 4100/7100	GC and SAW	▪ Food beverages ▪ Cosmetics ▪ Explosive & narcotics ▪ Nerve Agent ▪ Medical
9.	HKR Sensorsysteme	www.hkr-sensor.de QMB 6, MS-Sensor	QMB and MS	▪ Food beverages ▪ Cosmetics ▪ Organic materials ▪ Nerve Agent ▪ Pharmaceutical industry
10.	Illumina Inc.	www.illumina.com SNP genotyping services	Bead Array fibre optic	▪ Chemical detection ▪ Solvent vapours ▪ Pharmaceutical
11.	Lennartz Electronics	www.illumina.com SNP genotyping services	Bead Array fibre optic	▪ Chemical detectin ▪ Solvent vapours ▪ Pharmaceutical
12.	Marconi Applied	www.marconitech.com /	CP, MOS, QMB,	▪ Food processing

	Technologies	chemsens	SAW and other gas sensors	▪ Contaminant detection
13.	Microsensor Systems Inc.	www.microsensorsystems.com Hazmatcak, VaporLab, Eacgle	SAW or GC	▪ Chemical warfare ▪ Environment ▪ Medical ▪ Food, beverages, cosmetics
14.	OligoSense	Sch-www.uia.ac.be/struct / oligosense	Organic semiconductor sensor modules	▪ Automotive ▪ Food processing ▪ Contaminant detection
15.	Osmetech plc	www.osmetech.plc.uk OMA, Point of Care	CP	▪ Health care ▪ microbilogy
16.	RST Rostock	www.rst-rostock.de Sam System	QMB, SAW, MOS modules	▪ Security technology ▪ Environmental monitoring
17.	Smart Nose	www.smartnose.com Smart Nose 300	MS	▪ Food & process control ▪ Contaminant detection ▪ Environment ▪ Health
18.	Figaro Inc. Japan	www.figaro.com	MOS	▪ Food VOCs ▪ Gas leak detection ▪ Breath alcohol

Table A.4.1 concluded

Appendix 5

PCL -208 Card Specifications

A5.1 PCL – 208: 12 Bit Low Cost A/D - D/A Card

PCL-208 is a half size low cost AD/DA card for the low-end data acquisition needs. Though being a low cost card, it has very good accuracy & a reasonably good conversion speed, making it cost effective alternative to the costlier AD/DA cards. The PCL-208 uses successive approximation method of A to D conversion and offers 12-bit resolution with 25-microsecond conversion time. 8 single ended input channels available to the user, w/o sample & hold facility. This can be provided externally by user if required. In addition to the 8 channels of ADC, the card also provides a single channel 12-bit DAC output.

A5.2 Specifications

A5.2.1 Analog Input

- ADC : AD574 or equivalent, successive approximation type.

- Channels : 8 Single -ended input channels with 12-bit resolution.
- Range : Bipolar ± 5 V.
- Conversion Speed : 25 ì S.
- Accuracy : 0.015 % of reading ± 1 bit.
- Linearity : ± 1 bit.
- Trigger Modes : Software trigger.
- Data transfer : By program control.

A5.2.1 Analog Output

- **DAC channel:** with 12-bit resolution.
- **DAC :** AD7541 or equivalent, monolithic multiplying type.
- **Ranges :** 0 to + 5 V, 0 to +10V.
- **Setting time :** 30 mS.
- **Linearity :** ± ½ bit.
- **Output drive :** ± 5 mA max.

Appendix 6

TGS Sensor Specifications

A.6.1 TGS Sensors

The sensors used for E-nose system for experimentation on tea and spice flavour and aroma discrimination are MOS based gas sensors, TGS-2611, TGS-842, TGS-822, TGS-813-J01 and from FIGARO INC, Japan. These sensors are known to have good response to the food related VOCs and thus a natural choice for choosing them for this research work. Manufacturers specifications for these sensors are described in this appendix. It includes the circuit diagrams, power ratings, response characteristic plots and application areas of these sensors.

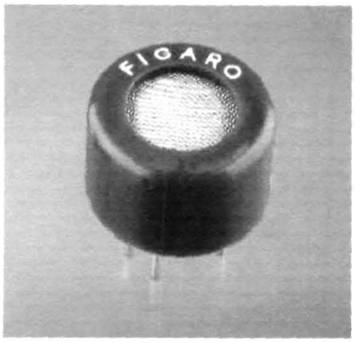
TGS 813 - for the detection of Combustible Gases

Features:

- * General purpose sensor with sensitivity to a wide range of combustible gases
- * High sensitivity to methane, propane, and butane
- * Long life and low cost
- * Uses simple electrical circuit

Applications:

- * Domestic gas leak detectors and alarms
- * Portable gas detectors



TGS - 813

The sensing element of Figaro gas sensors is a tin dioxide (SnO_2) semiconductor which has low conductivity in clean air. In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. A simple electrical circuit can convert the change in conductivity to an output signal which corresponds to the gas concentration.

The TGS 813 has high sensitivity to methane, propane, and butane, making it ideal for natural gas and LPG monitoring. The sensor can detect a wide range of gases, making it an excellent, low cost sensor for a wide variety of applications. Also available with a ceramic base which is highly resistant to severe environments up to 200°C (model# TGS 816).

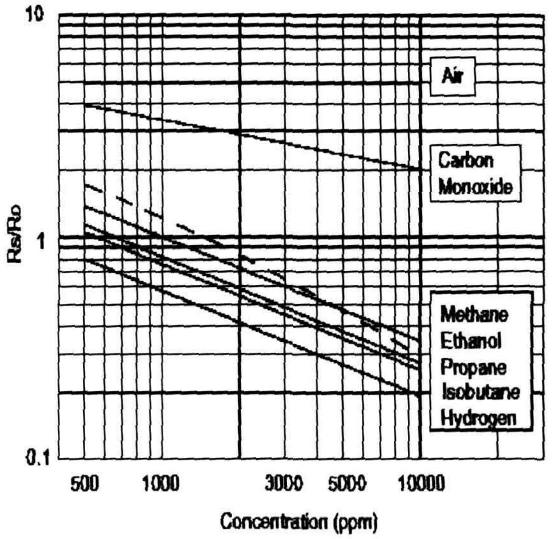
The figure below represents typical sensitivity char-acteristics, all data having been gathered at standard test conditions (see reverse side of this sheet). The Y-axis is indicated as *sensor resistance ratio* (Rs/Ro) which is defined as follows:

Rs = Sensor resistance of displayed gases at various concentrations
Ro = Sensor resistance in 1000ppm methane

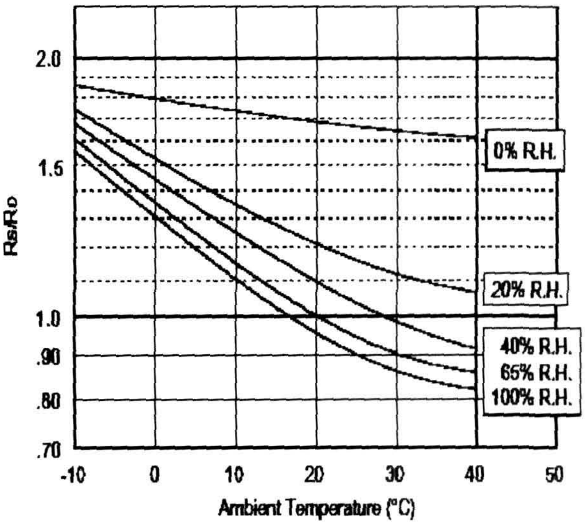
The figure below represents typical temperature and humidity dependency characteristics. Again, the Y-axis is indicated as *sensor resistance ratio* (Rs/Ro), defined as follows:

Rs = Sensor resistance at 1000ppm of methane at various temperatures/humidities
Ro = Sensor resistance at 1000ppm of methane at 20°C and 65% R.H.

Sensitivity Characteristics:

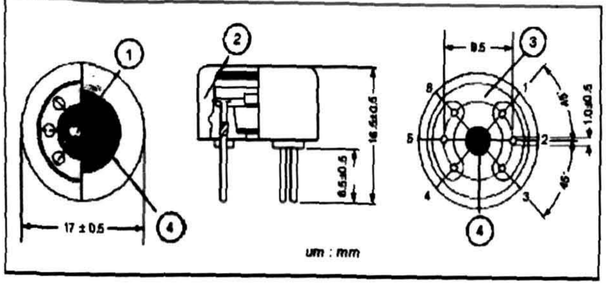


Temperature/Humidity Dependency:



IMPORTANT NOTE: OPERATING CONDITIONS IN WHICH FIGARO SENSORS ARE USED WILL VARY WITH EACH CUSTOMER'S SPECIFIC APPLICATIONS. FIGARO STRONGLY RECOMMENDS CONSULTING OUR TECHNICAL STAFF BEFORE DEPLOYING FIGARO SENSORS IN YOUR APPLICATION AND, IN PARTICULAR, WHEN CUSTOMER'S TARGET GASES ARE NOT LISTED HEREIN. FIGARO CANNOT ASSUME ANY RESPONSIBILITY FOR ANY USE OF ITS SENSORS IN A PRODUCT OR APPLICATION FOR WHICH SENSOR HAS NOT BEEN SPECIFICALLY TESTED BY FIGARO.

Structure and Dimensions:



1 Sensing Element:
SnO₂ is sintered to form a thick film on the surface of an alumina ceramic tube which contains an internal heater.

2 Cap:
Nylon 66

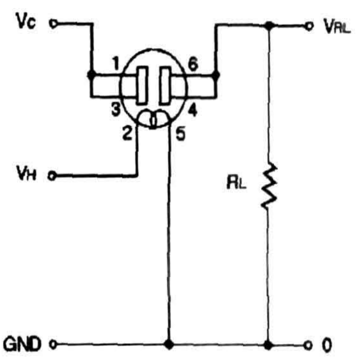
3 Sensor Base:
Nylon 66

4 Flame Arrestor:
100 mesh SUS 316 double gauze

Pin Connection and Basic Measuring Circuit:

The numbers shown around the sensor symbol in the circuit diagram at the right correspond with the pin numbers shown in the sensor's structure drawing (*above*). When the sensor is connected as shown in the basic circuit, output across the Load Resistor (V_{RL}) increases as the sensor's resistance (Rs) decreases, depending on gas concentration.

Basic Measuring Circuit:



Standard Circuit Conditions:

Item	Symbol	Rated Values	Remarks
Heater Voltage	V_H	5.0±0.2V	AC or DC
Circuit Voltage	V_C	Max. 24V	DC only Ps≤15mW
Load Resistance	R_L	Variable	0.45kΩ min.

Electrical Characteristics:

Item	Symbol	Condition	Specification
Sensor Resistance	Rs	Methane at 1000ppm/air	5kΩ ~ 15kΩ
Change Ratio of Sensor Resistance	Rs/Ro	$\frac{\text{Rs (Methane at 3000ppm/air)}}{\text{Rs (Methane at 1000ppm/air)}}$	0.60 ± 0.05
Heater Resistance	R_H	Room temperature	30.0 ± 3.0Ω
Heater Power Consumption	P_H	V_H=5.0V	835mW (typical)

Standard Test Conditions:

TGS 813 complies with the above electrical characteristics when the sensor is tested in standard conditions as specified below:

Test Gas Conditions: 20°±2°C, 65±5%R.H.
Circuit Conditions: V_C = 10.0±0.1V (AC or DC),
V_H = 5.0±0.05V (AC or DC),
R_L = 4.0kΩ±1%
Preheating period before testing: More than 7 days

Sensor Resistance (Rs) is calculated by the following formula:

$$Rs = \left(\frac{V_C}{V_{RL}} - 1\right) \times R_L$$

Power dissipation across sensor electrodes (Ps) is calculated by the following formula:

$$Ps = \frac{V_C^2 \times Rs}{(Rs + R_L)^2}$$

FIGARO USA, INC.
3703 West Lake Ave. Suite 203
Glenview, Illinois 60025
Phone: (847)-832-1701
Fax: (847)-832-1705
email: figarousa@figarosensor.com

For information on warranty, please refer to Standard Terms and Conditions of Sale of Figaro USA Inc.

REV: 9/02

TGS 822 - for the detection of Organic Solvent Vapors

Features:

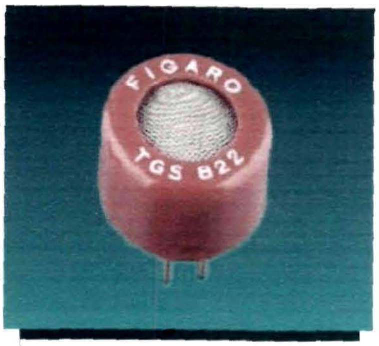
- *** High sensitivity to organic solvent vapors such as ethanol**
- *** High stability and reliability over a long period**
- *** Long life and low cost**
- *** Uses simple electrical circuit**

Applications:

- *** Breath alcohol detectors**
- *** Gas leak detectors/alarms**
- *** Solvent detectors for factories, dry cleaners, and semiconductor industries**

The sensing element of Figaro gas sensors is a tin dioxide (SnO_2) semiconductor which has low conductivity in clean air. In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. A simple electrical circuit can convert the change in conductivity to an output signal which corresponds to the gas concentration.

The **TGS 822** has high sensitivity to the vapors of organic solvents as well as other volatile vapors. It also has sensitivity to a variety of combustible gases such as carbon monoxide, making it a good general purpose sensor. Also available with a ceramic base which is highly resistant to severe environments as high as 200°C (model# TGS 823).



TGS - 822

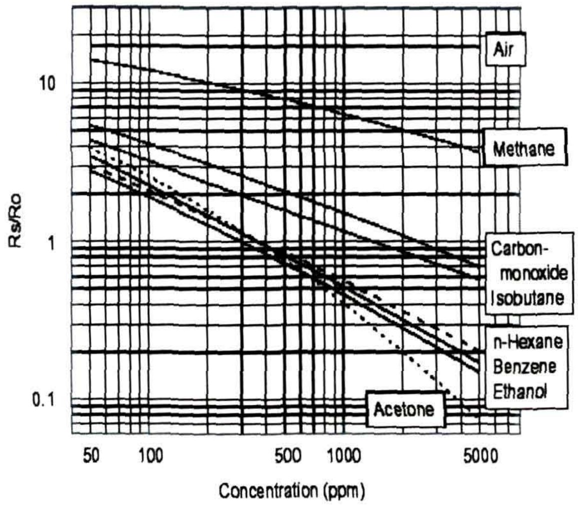
The figure below represents typical sensitivity char-acteristics, all data having been gathered at standard test conditions (see reverse side of this sheet). The Y-axis is indicated as *sensor resistance ratio* (Rs/Ro) which is defined as follows:

Rs = Sensor resistance of displayed gases at various concentrations
Ro = Sensor resistance in 300ppm ethanol

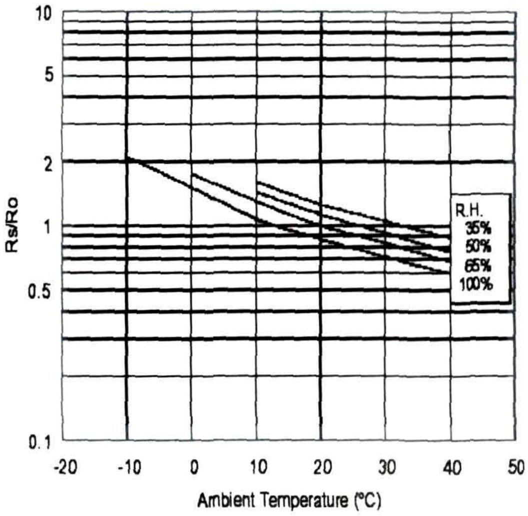
The figure below represents typical temperature and humidity dependency characteristics. Again, the Y-axis is indicated as *sensor resistance ratio* (Rs/Ro), defined as follows:

Rs = Sensor resistance at 300ppm of ethanol at various temperatures/humidities
Ro = Sensor resistance at 300ppm of ethanol at 20°C and 65% R.H.

Sensitivity Characteristics:

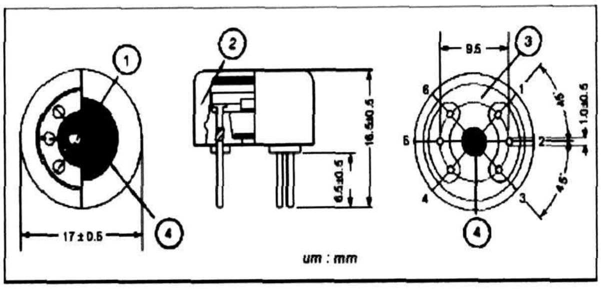


Temperature/Humidity Dependency:



IMPORTANT NOTE: OPERATING CONDITIONS IN WHICH FIGARO SENSORS ARE USED WILL VARY WITH EACH CUSTOMER'S SPECIFIC APPLICATIONS. FIGARO STRONGLY RECOMMENDS CONSULTING OUR TECHNICAL STAFF BEFORE DEPLOYING FIGARO SENSORS IN YOUR APPLICATION AND, IN PARTICULAR, WHEN CUSTOMER'S TARGET GASES ARE NOT LISTED HEREIN. FIGARO CANNOT ASSUME ANY RESPONSIBILITY FOR ANY USE OF ITS SENSORS IN A PRODUCT OR APPLICATION FOR WHICH SENSOR HAS NOT BEEN SPECIFICALLY TESTED BY FIGARO.

Structure and Dimensions:



① Sensing Element:
SnO₂ is sintered to form a thick film on the surface of an alumina ceramic tube which contains an internal heater.

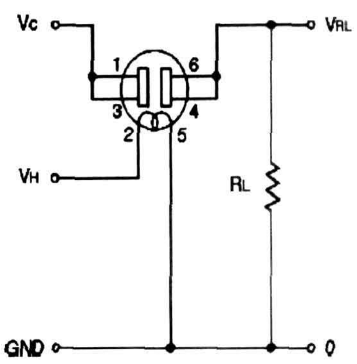
② Cap:
Nylon 66

③ Sensor Base:
Nylon 66

④ Flame Arrestor:
100 mesh SUS 316 double gauze

Pin Connection and Basic Measuring Circuit:
The numbers shown around the sensor symbol in the circuit diagram at the right correspond with the pin numbers shown in the sensor's structure drawing (*above*). When the sensor is connected as shown in the basic circuit, output across the Load Resistor (V_{RL}) increases as the sensor's resistance (Rs) decreases, depending on gas concentration.

Basic Measuring Circuit



Standard Circuit Conditions:

Item	Symbol	Rated Values	Remarks
Heater Voltage	V_H	5.0±0.2V	AC or DC
Circuit Voltage	V_C	Max. 24V	DC only Ps≤15mW
Load Resistance	R_L	Variable	0.45kΩ min.

Electrical Characteristics:

Item	Symbol	Condition	Specification
Sensor Resistance	Rs	Ethanol at 300ppm/air	1kΩ ~ 10kΩ
Change Ratio of Sensor Resistance	Rs/Ro	$\frac{\text{Rs(Ethanol at 300ppm/air)}}{\text{Rs(Ethanol at 50ppm/air)}}$	0.40 ± 0.10
Heater Resistance	R_H	Room temperature	38.0 ± 3.0Ω
Heater Power Consumption	P_H	V_H=5.0V	660mW (typical)

Standard Test Conditions:
TGS 822 complies with the above electrical characteristics when the sensor is tested in standard conditions as specified below:

Test Gas Conditions: 20°±2°C, 65±5%R.H.
Circuit Conditions: V_C = 10.0±0.1V (AC or DC),
V_H = 5.0±0.05V (AC or DC),
R_L = 10.0kΩ±1%
Preheating period before testing: More than 7 days

Sensor Resistance (Rs) is calculated by the following formula:

$$Rs = \left(\frac{V_C}{V_{RL}} - 1\right) \times R_L$$

Power dissipation across sensor electrodes (Ps) is calculated by the following formula:

$$Ps = \frac{Vc^2 \times Rs}{(Rs + R_L)^2}$$

FIGARO USA, INC.
3703 West Lake Ave. Suite 203
Glenview, Illinois 60025
Phone: (847)-832-1701
Fax: (847)-832-1705
email: figarousa@figarosensor.com

For information on warranty, please refer to Standard Terms and Conditions of Sale of Figaro USA Inc.

REV: 09/02

TGS 842 - for the detection of Methane

Features:

* High sensitivity to Methane
* Long-term stability
* Low sensitivity to alcohol vapors
* Uses simple electrical circuit

Applications:

* Domestic gas alarms for the detection of methane

* Portable gas detectors

The sensing element of Figaro gas sensors is a tin dioxide (SnO_2) semiconductor which has low conductivity in clean air. In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. A simple electrical circuit can convert the change in conductivity to an output signal which corresponds to the gas concentration.

The **TGS 842** has high sensitivity and selectivity to methane. Due to its low sensitivity to alcohol vapors and its low temperature/humidity dependency, the sensor can achieve good reproducibility, making it ideal for domestic gas alarms.

TGS - 842

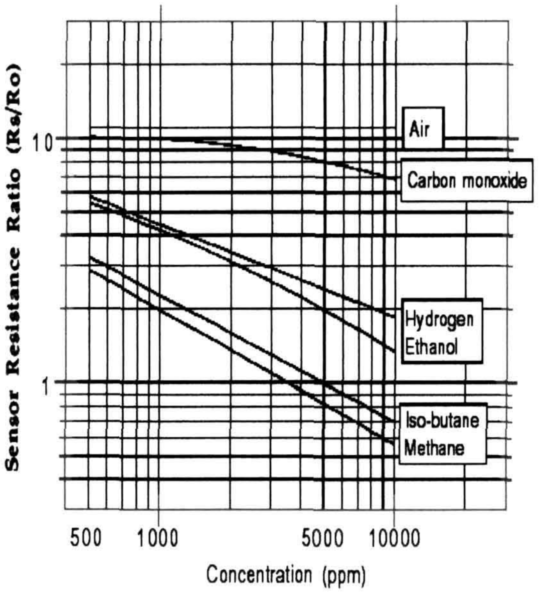
The figure below represents typical sensitivity char-acteristics, all data having been gathered at standard test conditions (see reverse side of this sheet). The Y-axis is indicated as *sensor resistance ratio* (Rs/Ro) which is defined as follows:

Rs = Sensor resistance of displayed gases at various concentrations
Ro = Sensor resistance in 3500ppm methane

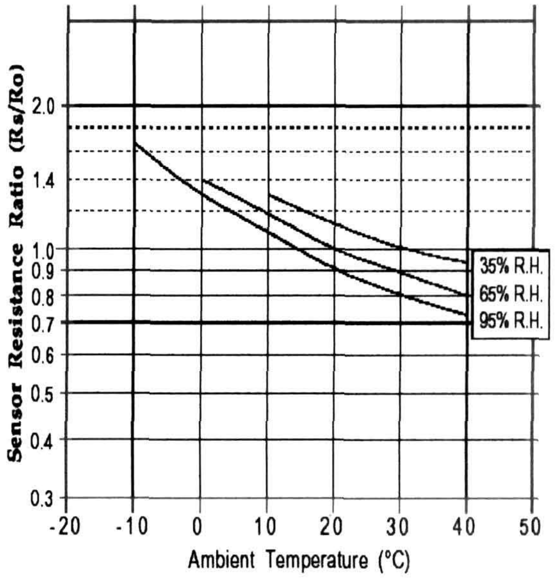
The figure below represents typical temperature and humidity dependency characteristics. Again, the Y-axis is indicated as *sensor resistance ratio* (Rs/Ro), defined as follows:

Rs = Sensor resistance at 3500ppm of methane at various temperatures/humidities
Ro = Sensor resistance at 3500ppm of methane at 20°C and 65% R.H.

Sensitivity Characteristics:

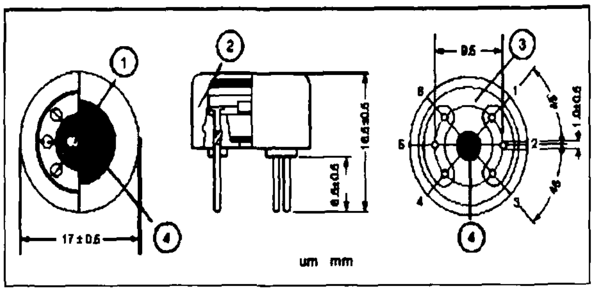


Temperature/Humidity Dependency:



IMPORTANT NOTE: OPERATING CONDITIONS IN WHICH FIGARO SENSORS ARE USED WILL VARY WITH EACH CUSTOMER'S SPECIFIC APPLICATIONS. FIGARO STRONGLY RECOMMENDS CONSULTING OUR TECHNICAL STAFF BEFORE DEPLOYING FIGARO SENSORS IN YOUR APPLICATION AND, IN PARTICULAR, WHEN CUSTOMER'S TARGET GASES ARE NOT LISTED HEREIN. FIGARO CANNOT ASSUME ANY RESPONSIBILITY FOR ANY USE OF ITS SENSORS IN A PRODUCT OR APPLICATION FOR WHICH SENSOR HAS NOT BEEN SPECIFICALLY TESTED BY FIGARO.

Structure and Dimensions:



(1) Sensing Element
SnO2 is sintered to form a thick film on the surface of an alumina ceramic tube which contains an internal heater

(2) Cap:
Nylon 66

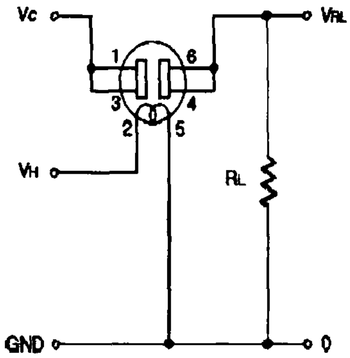
(3) Sensor Base.
Nylon 66

(4) Flame Arrestor:
100 mesh SUS316 double gauze

Pin Connection and Basic Measuring Circuit:

The numbers shown around the sensor symbol in the circuit diagram at the right correspond with the pin numbers shown in the sensor's structure drawing (*above*). When the sensor is connected as shown in the basic circuit, output across the Load Resistor (V_{RL}) *increases as the sensor's resistance (Rs) decreases, depending on* gas concentration

Basic Measuring Circuit:



Standard Circuit Conditions:

Item	Symbol	Rated Values	Remarks
Heater Voltage	V_H	5 0±0 2V	AC or DC
Circuit Voltage	V_C	Max 24V	DC only Ps≤15mW
Load Resistance	R_L	Variable	0 45kΩ min

Electrical Characteristics:

Item	Symbol	Condition	Specification
Sensor Resistance	Rs	Methane at 1000ppm/Air	3kΩ ~ 15kΩ
Change Ratio of Sensor Resistance	Rs/Ro	$\frac{\text{Rs (Methane at 3000ppm/air)}}{\text{Rs (Methane at 1000ppm/air)}}$	0 55 ± 0 05
Heater Resistance	R_H	Room temperature	30 0 ± 3 0Ω
Heater Power Consumption	P_H	V_H=5 0V	835mW (typical)

Standard Test Conditions:

TGS 842 complies with the above electrical characteristics when the sensor is tested in standard conditions as specified below:

Test Gas Conditions	20°±2°C, 65±5%R H
Circuit Conditions:	Vc = 10.0±0 1V (AC or DC), V_H = 5 0±0 05V (AC or DC), R_L = 4 0kΩ±1%
Preheating period before testing	More than 7 days

Sensor Resistance (Rs) is calculated by the following formula:

$$Rs = \left(\frac{Vc}{V_{RL}} - 1\right) \times R_L$$

Power dissipation across sensor electrodes (Ps) is calculated by the following formula:

$$Ps = \frac{Vc^2 \times Rs}{(Rs + R_L)^2}$$

FIGARO USA, INC.
3703 West Lake Ave Suite 203
Glenview, Illinois 60025
Phone (847)-832-1701
Fax (847)-832-1705
email figarousa@figarosensor com

For information on warranty, please refer to Standard Terms and Conditions of Sale of Figaro USA Inc.

REV 09/02

TGS 2611 - for the detection of Methane

Features:

- * Low power consumption
- * High sensitivity to methane
- * Long life and low cost
- * Uses simple electrical circuit

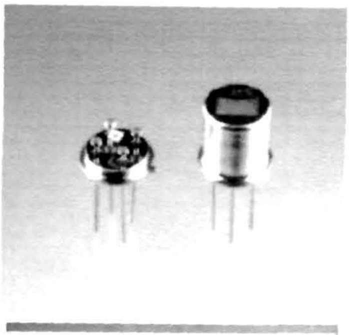
Applications:

- * Domestic gas alarms
- * Portable gas detectors
- * Gas leak detector for gas appliances

The sensing element is comprised of a metal oxide semiconductor layer formed on an alumina substrate of a sensing chip together with an integrated heater. In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. A simple electrical circuit can convert the change in conductivity to an output signal which corresponds to the gas concentration.

The **TGS 2611** has high sensitivity and selectivity to methane gas. Due to its low sensitivity to alcohol vapors (a typical interference gas in the domestic environment), the sensor is ideal for domestic gas alarms.

Due to miniaturization of the sensing chip, TGS 2611 requires a heater current of only 56mA and the device is housed in a standard TO-5 package.

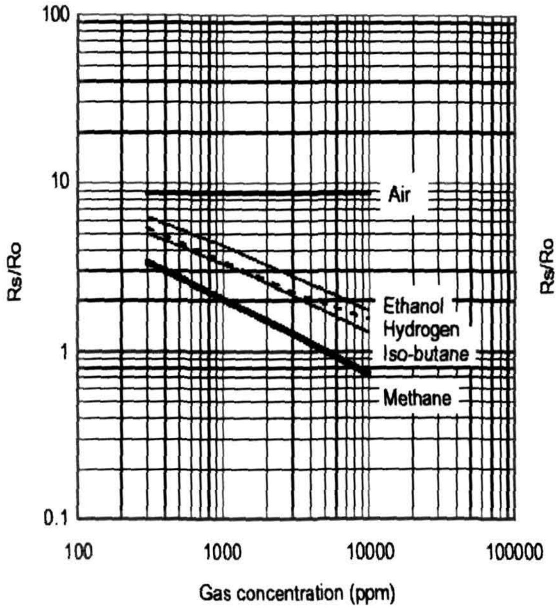


TGS - 2611

The figure below represents typical sensitivity characteristics, all data having been gathered at standard test conditions (see reverse side of this sheet). The Y-axis is indicated as *sensor resistance ratio* (Rs/Ro) which is defined as follows:

Rs = Sensor resistance in displayed gases at
various concentrations
Ro = Sensor resistance in 5000ppm of methane

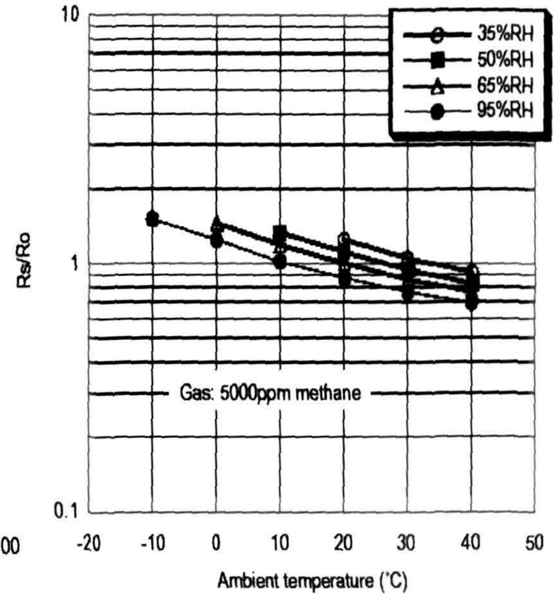
Sensitivity Characteristics:



The figure below represents typical temperature and humidity dependency characteristics. Again, the Y-axis is indicated as *sensor resistance ratio* (Rs/Ro), defined as follows:

Rs = Sensor resistance in 5000ppm of methane
at various temperatures/humidities
Ro = Sensor resistance in 5000ppm of methane
at 20°C and 65% R.H.

Temperature/Humidity Dependency:

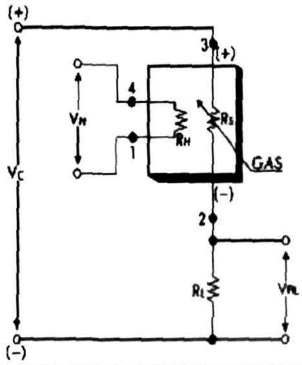


IMPORTANT NOTE: OPERATING CONDITIONS IN WHICH FIGARO SENSORS ARE USED WILL VARY WITH EACH CUSTOMER'S SPECIFIC APPLICATIONS. FIGARO STRONGLY RECOMMENDS CONSULTING OUR TECHNICAL STAFF BEFORE DEPLOYING FIGARO SENSORS IN YOUR APPLICATION AND, IN PARTICULAR, WHEN CUSTOMER'S TARGET GASES ARE NOT LISTED HEREIN. FIGARO CANNOT ASSUME ANY RESPONSIBILITY FOR ANY USE OF ITS SENSORS IN A PRODUCT OR APPLICATION FOR WHICH SENSOR HAS NOT BEEN SPECIFICALLY TESTED BY FIGARO.

Basic Measuring Circuit:

The sensor requires two voltage inputs: heater voltage (V_H) and circuit voltage (V_C). The heater voltage (V_H) is applied to the integrated heater in order to maintain the sensing element at a specific temperature which is optimal for sensing. Circuit voltage (V_C) is applied to allow measurement of voltage (V_{RL}) across a load resistor (R_L) which is connected in series with the sensor.

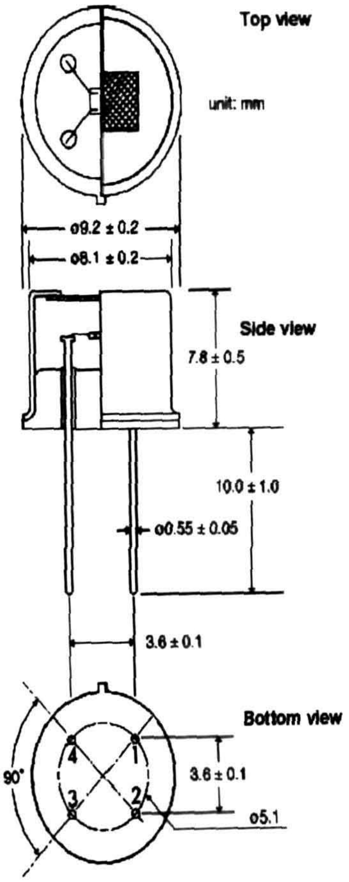
A common power supply circuit can be used for both V_C and V_H to fulfill the sensor's electrical requirements. The value of the load resistor (R_L) should be chosen to optimize the alarm threshold value, keeping power dissipation (P_S) of the semiconductor below a limit of 15mW. Power dissipation (P_S) will be highest when the value of Rs is equal to R_L on exposure to gas.



Specifications:

Model number			TGS 2611	
Sensing element type			D1	
Standard package			TO-5 metal can	
Target gases			Methane, Natural Gas	
Typical detection range			500 ~ 10,000 ppm	
Standard circuit conditions	Heater Voltage	V_H	5.0±0.2V DC/AC	
	Circuit voltage	V_C	5.0±0.2V DC	P_S ≤ 15mW
	Load resistance	R_L	Variable	0.45kΩ min.
Electrical characteristics under standard test conditions	Heater resistance	R_H	59Ω at room temp. (typical)	
	Heater current	I_H	56 ± 5mA	
	Heater power consumption	P_H	280±25mW	
	Sensor resistance	Rs	0.68~6.8 kΩ in 5000ppm methane	
	Sensitivity (change ratio of Rs)		0.60 ± 0.06	$\frac{\text{Rs (9000ppm)}}{\text{Rs (3000ppm)}}$
Standard test conditions	Test gas conditions		Methane in air at 20±2°C, 65±5%RH	
	Circuit conditions		V_C = 5.0±0.01V DC V_H = 5.0±0.05V DC	
	Conditioning period before test		7 days	

Structure and Dimensions:



Pin connection:
1 : Heater
2 : Sensor electrode (-)
3 : Sensor electrode (+)
4 : Heater

The value of power dissipation (P_S) can be calculated by utilizing the following formula:

$$P_S = \frac{(V_C - V_{RL})^2}{R_S}$$

Sensor resistance (Rs) is calculated with a measured value of V_{RL} by using the following formula:

$$R_S = \frac{V_C - V_{RL}}{V_{RL}} \times R_L$$

For information on warranty, please refer to Standard Terms and Conditions of Sale of Figaro USA Inc. All sensor characteristics shown in this brochure represent typical characteristics. Actual characteristics vary from sensor to sensor. The only characteristics warranted are those in the Specification table above.

FIGARO USA, INC.
3703 West Lake Ave. Suite 203
Glenview, Illinois 60025
Phone: (847)-832-1701
Fax: (847)-832-1705
email: figarousa@figarosensor.com

REV: 01/00

Appendix 7

MATLAB Programming Codes

PCA Analysis

```
% MATLAB Source CODE for PCA
% analysis of ten Tea flavour
% qualities (Bakey, Musty, Poor,
% Sour, Sweet, Woody, Smokey,
% Minty, Baggy and Papery).
%
%=====================
% close all the running programs
% and fig windows
close all;
% Clear work space and command
% windows
clear all;
% Load the data file
load Tea_data.mat;
% Load the data for the first Tea
% Quality (Bakey)
data=bakey_data';
% Find out Covariance of the raw
% data
covdata=cov(data);

% Find Principle Components for the %
bakey data
% from covariance data vector
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle
% Components for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector % by
using only
% first three Principal Components
finaldata=PC123'*bakey_data;
% Plot three dimensional PCA results % in
scatter diagram
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*','r');

% Clear work space and command
% windows and all the running
%  programs
close all;
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
% Load the data file
load Tea_data.mat;
```

```
% Load the data for the Second Tea %
Quality (Musty)
data=musty_data';
% Find out Covariance of the raw
% data
covdata=cov(data);
% Find Principle Components for the %
musty data from covdata
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle
% Components for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*musty_data;
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'+','g');
% close all the running programs,
% Clear work space and command
% windows
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
% Load the data file
load Tea_data.mat;
% Load the data for the Third Tea Quality
(poor)
 data=poor_data';
% Find out Covariance of the raw
% data
covdata=cov(data);
% Find Principle Components for the %
poor data from covdata
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*poor_data;
% Plot three dimensional PCA resultss
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*','b')

% close all the running programs, Clear work
space and command windows
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
```

```
% Load the data file
load Tea_data.mat;
% Load the data for the Fouth Tea Quality
(sour)
data=sour_data';
% Find out Covariance of the raw data
covdata=cov(data);
% Find Principle Components for the sour tea
data from covdata
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*sour_data;
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*','c')

% close all the running programs%, Clear
work space and command windows
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
% Load the data file
load Tea_data.mat;
% Load the data for the Fifth Tea Quality
(Sweet)
data=sweet_data';
% Find out Covariance of the raw data
covdata=cov(data);
% Find Principle Components for the sweet
tea data
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*sweet_data;
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*','k')
% close all the running programs%, Clear
work space and command windows
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
% Load the data file
```

```
load Tea_data.mat;
% Load the data for the Fifth Tea Quality
(Sweet)
data=woody_data';
% Find out Covariance of the raw data
covdata=cov(data);
% Find Principle Components for the sweet
tea data
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*woody_data;
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*','m')

% close all the running programs%, Clear
work space and command windows
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
% Load the data file
load Tea_data.mat;
% Load the data for the Fifth Tea Quality
(Sweet)
data=smokey_data';
% Find out Covariance of the raw data
covdata=cov(data);
% Find Principle Components for the sweet
tea data
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*smokey_data;
c= [0.8, 0.8, 0];
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*',c)

% close all the running programs%, Clear
work space and command windows
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
% Load the data file
load Tea_data.mat;
% Load the data for the Fifth Tea Quality
(Sweet)
data=minty_data';
% Find out Covariance of the raw data
covdata=cov(data);
% Find Principle Components for the sweet
tea data
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*minty_data;
c1= [1, 0.6, 0.5];
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*',c1)
% close all the running programs%, Clear
work space and command windows
clear;

% wait for key board input signal
pause;
% don not erase the previous graph
hold on;
% Load the data file
load Tea_data.mat;
% Load the data for the Fifth Tea Quality
(Sweet)
data=baggy_data';
% Find out Covariance of the raw data
covdata=cov(data);
% Find Principle Components for the sweet
tea data
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*baggy_data;
c2= [0.7, 0.1, 1];
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*', c2)
% close all the running programs%, Clear
work space and command windows
clear;
% wait for key board input signal
pause;
% don not erase the previous graph
```

```
hold on;
% Load the data file
load Tea_data.mat;
% Load the data for the Fifth Tea Quality
(Sweet)
data=papery_data';
% Find out Covariance of the raw data
covdata=cov(data);
% Find Principle Components for the sweet
tea data
[PC, LATENT, EXPLAINED] =
PCACOV(covdata);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
finaldata=PC123'*papery_data;
c3= [0.7, 0.6, 0.6];
% Plot three dimensional PCA results
scatter3(finaldata(1,:),finaldata(2,:),finaldata(3,
:),'*',c3)
h = legend('1. Bakey','2. Musty', '3. Poor', '4.
Sour', '5. Sweet','6. Woody', '7. Smokey','8.
Minty', '9. Baggy', '10. Papery');
```

```
% CODES FOR PCA analysis of five common
Indian % Spices. The CODES are written in
MATLAB VERSION 6
% Written by K R Kashwan in Department of
Electronics
% School of Science & Technology, Tezpur
University Assam (India)-784028
% close all the running programs
close all;
% Clear work space and command windows
clear all;
% Load the data file
 load Spice_data.mat;

% Load the data for the first spice (Chili)
padata=ChiliC_data';
% Find out Covariance of the raw data
pdat=cov(padata);
% Find Principle Components for the chili
data
% from covariance data vector
[PC, LATENT, EXPLAINED] =
PCACOV(pdat);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);

% Find out the Feature Data Vector by using
only
% first three Principal Components
fd=PC123'*ChiliC_data;
% Plot three dimensional PCA results in catter
diagram
scatter3(fd(1,:),fd(2,:),fd(3,:),'*','r');

% Clear work space and command windows,
close all the running programs
clear;
% wait for key board input signal
pause;
% do not erase the previous graph
hold on;
% Load the data file
load Spice_data.mat;
% Load the data for the Second spice (Garlic)
padata=GarlicC_data';
% Find out Covariance of the raw data
pdat=cov(padata);
% Find Principle Components for the garlic
data
[PC, LATENT, EXPLAINED] =
PCACOV(pdat);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
fd=PC123'*GarlicC_data;
% Plot three dimensional PCA results
scatter3(fd(1,:),fd(2,:),fd(3,:),'+','g');

% close all the running programs, Clear work
space and command windows
clear;
% wait for key board input signal
pause;
% Load the data file
load Spice_data.mat;
% Load the data for the Third spice (Ginger)
padata=GingerC_data';
% Find out Covariance of the raw data
pdat=cov(padata);
% Find Principle Components for the ginger
data
[PC, LATENT, EXPLAINED] =
PCACOV(pdat);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
fd=PC123'*GingerC_data;
% Plot three dimensional PCA results
scatter3(fd(1,:),fd(2,:),fd(3,:),'o','b')
```

```
% close all the running programs, Clear work
space and command windows
clear;
% wait for key board input signal
pause;
% Load the data file
load Spice_data.mat;
% Load the data for the Fourth spice (Onion)
padata=OnionC_data';
% Find out Covariance of the raw data
pdat=cov(padata);
% Find Principle Components for the Onion
data
[PC, LATENT, EXPLAINED] =
PCACOV(pdat);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
fd=PC123'*OnionC_data;
% Plot three dimensional PCA results
scatter3(fd(1,:),fd(2,:),fd(3,:),'.','c')

% close all the running programs %, Clear
work space and command windows
clear;

% wait for key board input signal
pause;
% Load the data file
load Spice_data.mat;
% Load the data for the Fifth spice
(Turmeric)
padata=TurmericC_data';
% Find out Covariance of the raw data
pdat=cov(padata);
% Find Principle Components for the
turmeric data
[PC, LATENT, EXPLAINED] =
PCACOV(pdat);
% Select the first three Principle Components
for the
% dimensional reduction
PC123=PC(:,1:3);
% Find out the Feature Data Vector by using
only
% first three Principal Components
fd=PC123'*TurmericC_data;
% Plot three dimensional PCA results
scatter3(fd(1,:),fd(2,:),fd(3,:),'o','k')
h = legend('Chili',' Garlic', 'Ginger', 'Onion',
'Turmeric');
```

MLP

```
clear all;close all;

%Define a 4-6-10 MLP with Sigmoidal
Neurons at the Hidden
%and output layers; train using gradient
descent backprop

net=newff([0 15; 0 15; 0 15; 0
15;],[6,10],{'logsig','logsig'},'traingd');

%Show training progress every 50 epochs
net.trainParam.show=200;

%Set learning rate to 0.5
net.trainParam.lr=0.9;

%Set maximum epochs to 15000
net.trainParam.epochs=10000;

net.trainparam.goal=1e-3;

%data of baggy sample of tea

load baggy.mat;
p(1,1:10)=baggy1;
p(2,1:10)=baggy2;
p(3,1:10)=baggy3;
p(4,1:10)=baggy4;

clear baggy1 baggy2 baggy3 baggy4;

%data of bakey sample of tea
load bakey.mat;
p(1,11:20)=bakey1;
p(2,11:20)=bakey2;
p(3,11:20)=bakey3;
p(4,11:20)=bakey4;

clear bakey1 bakey2 bakey3 bakey4;

%data of minty sample of tea
load minty.mat;
p(1,21:30)=minty1;
p(2,21:30)=minty2;
p(3,21:30)=minty3;
p(4,21:30)=minty4;

clear minty1 minty2 minty3 minty4;

%data of musty sample of tea
load musty.mat;
p(1,31:40)=musty1;
p(2,31:40)=musty2;
p(3,31:40)=musty3;
p(4,31:40)=musty4;

clear musty1 musty2 musty3 musty4;

%data of papery sample of tea
load papery.mat;
p(1,41:50)=papery1;
```

```
p(2,41:50)=papery2;
p(3,41:50)=papery3;
p(4,41:50)=papery4;

clear papery1 papery2 papery3 papery4;

%data of poor sample of tea
load poor.mat;
p(1,51:60)=poor1;
p(2,51:60)=poor2;
p(3,51:60)=poor3;
p(4,51:60)=poor4;

clear poor1 poor2 poor3 poor4;

%data of smokey sample of tea
load smokey.mat;
p(1,61:70)=smokey1;
p(2,61:70)=smokey2;
p(3,61:70)=smokey3;
p(4,61:70)=smokey4;

clear smokey1 smokey2 smokey3 smokey4;

%data of sour flavour sample of tea
load sour.mat;
p(1,71:80)=sour1;
p(2,71:80)=sour2;
p(3,71:80)=sour3;
p(4,71:80)=sour4;

clear sour1 sour2 sour3 sour4;

%data of sweet sample of tea
load sweet.mat;
p(1,81:90)=sweet1;
p(2,81:90)=sweet2;
p(3,81:90)=sweet3;
p(4,81:90)=sweet4;

clear sweet1 sweet2 sweet3 sweet4;

%data of woody sample of tea
load woody.mat;
p(1,91:100)=woody1;
p(2,91:100)=woody2;
p(3,91:100)=woody3;
p(4,91:100)=woody4;

clear woody1 woody2 woody3 woody4;

%Define training input and target data vectors

tone=ones(1,10);
t(1,1:10)=tone;
t(2,11:20)=tone;
t(3,21:30)=tone;
t(4,31:40)=tone;
t(5,41:50)=tone;
t(6,51:60)=tone;
t(7,61:70)=tone;
t(8,71:80)=tone;
t(9,81:90)=tone;
t(10,91:100)=tone;

clear tone;
%train the network
net=train(net,p,t);

%Validate Training
output=sim(net,p);

%Test network wth new data
load baggy.mat;
p1(1,1:10)=baggy1;
p1(2,1:10)=baggy2;
p1(3,1:10)=baggy3;
p1(4,1:10)=baggy4;

clear baggy1 baggy2 baggy3 baggy4;

%p1=baggy         %[[ 0.313; 0.205;0.193;
1.907],[0.435;0.322;0.349;2.634]];
t1=t;
clear t1;
output1=sim(net,p1);
```

LVQ

```
clear all;close all;

%Define a 4-10-0.8 LVQ with learnlv1
Neurons at the Hidden
%and output layers; train using learnr

net=newlvq([0 5; 0 5; 0 5; 0 5;],4, [0.1 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1 0.1 ], 0.8,
'learnlv1');

%Show training progress every 200 epochs
net.trainParam.show=200;

%Set learning rate to 0.5
net.trainParam.lr=0.9;

%Set maximum epochs to 15000
net.trainParam.epochs=1000;

net.trainparam.goal=1e-3;

%data of baggy sample of tea

load baggy.mat;
p(1,1:10)=baggy1;
p(2,1:10)=baggy2;
p(3,1:10)=baggy3;
p(4,1:10)=baggy4;
```

```
clear baggy1 baggy2 baggy3 baggy4;

%data of bakey sample of tea
load bakey.mat;
p(1,11:20)=bakey1;
p(2,11:20)=bakey2;
p(3,11:20)=bakey3;
p(4,11:20)=bakey4;

clear bakey1 bakey2 bakey3 bakey4;

%data of minty sample of tea
load minty.mat;
p(1,21:30)=minty1;
p(2,21:30)=minty2;
p(3,21:30)=minty3;
p(4,21:30)=minty4;

clear minty1 minty2 minty3 minty4;

%data of musty sample of tea
load musty.mat;
p(1,31:40)=musty1;
p(2,31:40)=musty2;
p(3,31:40)=musty3;
p(4,31:40)=musty4;

clear musty1 musty2 musty3 musty4;

%data of papery sample of tea
load papery.mat;
p(1,41:50)=papery1;
p(2,41:50)=papery2;
p(3,41:50)=papery3;
p(4,41:50)=papery4;

clear papery1 papery2 papery3 papery4;

%data of poor sample of tea
load poor.mat;
p(1,51:60)=poor1;
p(2,51:60)=poor2;
p(3,51:60)=poor3;
p(4,51:60)=poor4;

clear poor1 poor2 poor3 poor4;
%data of smokey sample of tea
load smokey.mat;
p(1,61:70)=smokey1;
p(2,61:70)=smokey2;
p(3,61:70)=smokey3;
p(4,61:70)=smokey4;

clear smokey1 smokey2 smokey3 smokey4;

%data of sour flavour sample of tea
load sour.mat;
p(1,71:80)=sour1;
p(2,71:80)=sour2;
p(3,71:80)=sour3;
p(4,71:80)=sour4;

clear sour1 sour2 sour3 sour4;


%data of sweet sample of tea
load sweet.mat;
p(1,81:90)=sweet1;
p(2,81:90)=sweet2;
p(3,81:90)=sweet3;
p(4,81:90)=sweet4;

clear sweet1 sweet2 sweet3 sweet4;

%data of woody sample of tea
load woody.mat;
p(1,91:100)=woody1;
p(2,91:100)=woody2;
p(3,91:100)=woody3;
p(4,91:100)=woody4;

clear woody1 woody2 woody3 woody4;

%Define training input and target data vectors

tone=ones(1,10);
t(1,1:10)=tone;
t(2,11:20)=tone;
t(3,21:30)=tone;
t(4,31:40)=tone;
t(5,41:50)=tone;
t(6,51:60)=tone;
t(7,61:70)=tone;
t(8,71:80)=tone;
t(9,81:90)=tone;
t(10,91:100)=tone;

%train the network
net=train(net,p,t);

%Validate Training
output=sim(net,p);

%Test network wth new data
p1=[[ 0.313; 0.205;0.193;
1.907],[0.435;0.322;0.349;2.634]];
%t1=[0 1];
output1=sim(net,p1);
```

PNN

```
function net=newpnn(p,t,spread)
NEWPNN Designs a probabilistic neural
network.
net = newpnn
net = newpnn(P,T,SPREAD)
NET = NEWPNN creates a new network with
a dialog box.
```

NET = NEWPNN(P,T,SPREAD) takes two or three arguments,
P - RxQ matrix of Q input vectors.
T - SxQ matrix of Q target class vectors.
SPREAD - Spread of radial basis functions, 0.1 and returns a new probabilistic neural network.

Here a classification problem is defined with a set of input data P and class indices Tc.

```
load teadata;
load teaflavour;
P = teadata;
Tc = teaflavour;
```

Here the teaflavours are converted to target vectors,
and a PNN is designed and tested.

```
T = ind2vec(Tc)
 net = newpnn(P,T);
Y = sim(net,P)
Yc = vec2ind(Y)
```

Algorithm

```
if nargin < 2
  net = newnet('newpnn');
  return
end

% Defaults
if nargin < 3, spread = 0.1; end

% Error checks
if (~isa(p,'double')) | (~isreal(p)) | (length(p)
== 0)
  error('Inputs are not a non-empty real
matrix.')
end
if (~isa(t,'double')) | (~isreal(t)) | (length(t) ==
0)
  error('Targets are not a non-empty real
matrix.')
end
if (size(p,2) ~= size(t,2))
  error('Inputs and Targets have different
numbers of columns.')
end
if (~isa(spread,'double')) | ~isreal(spread) |
any(size(spread) ~= 1) | (spread < 0)
  error('Spread is not a positive or zero real
value.')
end

% Dimensions
[R,Q] = size(p);
[S,Q] = size(t);

% Architecture
net = network(1,2,[1;0],[1;0],[0 0;1 0],[0 1]);

% Simulation
net.inputs{1}.size = R;
net.inputWeights{1,1}.weightFcn = 'dist';
net.layers{1}.netInputFcn = 'netprod';
net.layers{1}.transferFcn = 'radbas';
net.layers{1}.size = Q;
net.layers{2}.size = S;
net.layers{2}.transferFcn = 'compet';

% Weight and Bias Values
net.b{1} = zeros(Q,1)+sqrt(-log(.5))/spread;
net.iw{1,1} = p';
net.lw{2,1} = t;
```

RBF

```
function [net,tr]=newrb(p,t,goal,spread,mn,df)
NEWRB Design a radial basis network.

        net = newrb
        [net,tr] =
newrb(P,T,GOAL,SPREAD,MN,DF)

% NET = NEWRB creates a new network
with a dialog box.
%
% NEWRB(PR,T,GOAL,SPREAD,MN,DF)
takes these arguments,
%   P      - RxQ matrix of Q input vectors.
%   T      - SxQ matrix of Q target class
vectors.
%   GOAL   - Mean squared error goal,
%   SPREAD - Spread of radial basis
functions,
%   MN     - Maximum number of neurons,.
%   DF     - Number of neurons to add
between displays,
% and returns a new radial basis network.
%
```

Here a radial basis network is designed for inputs P teadata,
and targets T teaflavour.

```
load teadata;
load teaflavour
P =teadata;
T = teaflavour;
net = newrb(P,T);
```

Here the network is simulated for a new input.

```
Y = sim(net,P)
```

Algorithm

```
if nargin < 2
  net = newnet('newrb');
  tr = [];
  return
```

```
end

% Defaults
if nargin < 3, goal = 0; end
if nargin < 4, spread = 1; end
if nargin < 6, df = 25; end

% Error checks
if (~isa(p,'double')) | (~isreal(p)) | (length(p)
== 0)
  error('Inputs are not a non-empty real
matrix.')
end
if (~isa(t,'double')) | (~isreal(t)) | (length(t) ==
0)
  error('Targets are not a non-empty real
matrix.')
end
if (size(p,2) ~= size(t,2))
  error('Inputs and Targets have different
numbers of columns.')
end
if (~isa(goal,'double')) | ~isreal(goal) |
any(size(goal) ~= 1) | (goal < 0)
  error('Performance goal is not a positive or
zero real value.')
end
if (~isa(spread,'double')) | ~isreal(spread) |
any(size(spread) ~= 1) | (spread < 0)
  error('Spread is not a positive or zero real
value.')
end
if (~isa(df,'double')) | ~isreal(df) | any(size(df)
~= 1) | (df < 1) | (round(df) ~= df)
  error('Display frequency is not a positive
integer.')
end

% More defaults
Q = size(p,2);
if nargin < 5, mn = Q; end

% More error checking
if (~isa(mn,'double')) | ~isreal(mn) |
any(size(mn) ~= 1) | (mn < 1) | (round(mn) ~=
mn)
  error('Maximum neurons is not a positive
integer.')
end

% Dimensions
R = size(p,1);
S2 = size(t,1);

% Architecture
net = network(1,2,[1;1],[1; 0],[0 0;1 0],[0 1]);

% Simulation
net.inputs{1}.size = R;
net.layers{1}.size = 0;
net.inputWeights{1,1}.weightFcn = 'dist';
net.layers{1}.netInputFcn = 'netprod';
net.layers{1}.transferFcn = 'radbas';
net.layers{2}.size = S2;

% Performance
net.performFcn = 'mse';

% Design Weights and Bias Values
[w1,b1,w2,b2,tr] =
designrb(p,t,goal,spread,mn,df);

net.layers{1}.size = length(b1);
net.b{1} = b1;
net.iw{1,1} = w1;
net.b{2} = b2;
net.lw{2,1} = w2;

function [w1,b1,w2,b2,tr] =
designrb(p,t,eg,sp,mn,df)

[r,q] = size(p);
[s2,q] = size(t);
b = sqrt(-log(.5))/sp;

% RADIAL BASIS LAYER OUTPUTS
P = radbas(dist(p',p)*b);
PP = sum(P.*P)';
d = t';
dd = sum(d.*d)';

% CALCULATE "ERRORS" ASSOCIATED
WITH VECTORS
e = ((P' * d)' .^ 2) ./ (dd * PP');

% PICK VECTOR WITH MOST "ERROR"
pick = findLargeColumn(e);
used = [];
left = 1:q;
W = P(:,pick);
P(:,pick) = []; PP(pick,:) = [];
e(:,pick) = [];
used = [used left(pick)];
left(pick) = [];

% CALCULATE ACTUAL ERROR
w1 = p(:,used)';
a1 = radbas(dist(w1,p)*b);
[w2,b2] = solvelin2(a1,t);
a2 = w2*a1 + b2*ones(1,q);
sse = sumsqr(t-a2);

% Start
tr = newtr(mn,'perf');
tr.perf(1) = sumsqr(t);
tr.perf(2) = sse;
if isfinite(df)
  fprintf('NEWRB, neurons = 0, SSE =
%g\n',sse);
end
```

```
flag_stop = 0;

for k = 2:mn

  % CALCULATE "ERRORS" ASSOCIATED
WITH VECTORS
  wj = W(:,k-1);
  a = wj' * P / (wj'*wj);
  P = P - wj * a;
  PP = sum(P.*P)';
  e = ((P' * d)' .^ 2) ./ (dd * PP');

  % PICK VECTOR WITH MOST "ERROR"
  pick = findLargeColumn(e);
  W = [W, P(:,pick)];
  P(:,pick) = []; PP(pick,:) = [];
  e(:,pick) = [];
  used = [used left(pick)];
  left(pick) = [];

  % CALCULATE ACTUAL ERROR
  w1 = p(:,used)';
  a1 = radbas(dist(w1,p)*b);
  [w2,b2] = solvelin2(a1,t);
  a2 = w2*a1 + b2*ones(1,q);
  sse = sumsqr(t-a2);

  % PROGRESS
  tr.perf(k+1) = sse;

  % DISPLAY
  if isfinite(df) & (~rem(k,df))
    fprintf('NEWRB, neurons = %g, SSE =
%g\n',k,sse);

flag_stop=plotperf(tr,eg,'NEWRB',k);
  end

  % CHECK ERROR
  if (sse < eg), break, end
  if (flag_stop), break, end

end

[S1,R] = size(w1);
b1 = ones(S1,1)*b;

% Finish
tr = cliptr(tr,k);

function i = findLargeColumn(m)

replace = find(isnan(m));
m(replace) = zeros(size(replace));

m = sum(m .^ 2,1);
i = find(m == max(m));
i = i(1);

function [w,b] = solvelin2(p,t)

if nargout <= 1
  w= t/p;
else
  [pr,pc] = size(p);
  x = t/[p; ones(1,pc)];
  w = x(:,1:pr);
  b = x(:,pr+1);
end
```

Regression Analysis

```
function [b,bint,r,rint,stats] =
regress(y,X,alpha)
%REGRESS Multiple linear regression using
least squares.
%   b = REGRESS(y,X) returns the vector of
regression coefficients, b,
%   in the linear model  y = Xb, (X is an nxp
matrix, y is the nx1
%   vector of observations).
%
%   [B,BINT,R,RINT,STATS] =
REGRESS(y,X,alpha) uses the input, ALPHA
%   to calculate 100(1 - ALPHA) confidence
intervals for B and the
%   residual vector, R, in BINT and RINT
respectively.  The vector
%   STATS contains the R-square statistic
along with the F and p
%   values for the regression.
%
%   The F and p values are computed under
%   the assumption that the model contains a
constant term, and they
%   are not correct for models without a
constant.  The R-square
%   value is the ratio of the regression sum of
squares to the
%   total sum of squares.

if  nargin < 2,
   error('REGRESS requires at least two input
arguments.');
end

if nargin == 2,
   alpha = 0.05;
end

% Check that matrix (X) and left hand side (y)
have compatible dimensions
[n,p] = size(X);
[n1,collhs] = size(y);
if n ~= n1,
   error('The number of rows in Y must equal
the number of rows in X.');
end
```

```
if collhs ~= 1,
   error('Y must be a vector, not a matrix');
end

% Remove missing values, if any
wasnan = (isnan(y) | any(isnan(X),2));
if (any(wasnan))
  y(wasnan) = [];
  X(wasnan,:) = [];
  n = length(y);
end

% Find the least squares solution.
[Q, R]=qr(X,0);
b = R\(Q'*y);

% Find a confidence interval for each
component of x

RI = R\eye(p);
xdiag=sqrt(sum((RI .* RI)',1))';
nu = n-p;                    % Residual degrees of
freedom
yhat = X*b;                  % Predicted
responses at each data point.
r = y-yhat;                  % Residuals.
if nu ~= 0
  rmse = norm(r)/sqrt(nu);       % Root mean
square error.
else
  rmse = Inf;
end
s2 = rmse^2;                 % Estimator of error
variance.
tval = tinv((1-alpha/2),nu);
bint = [b-tval*xdiag*rmse,
b+tval*xdiag*rmse];

% Calculate R-squared.
if nargout==5,
  RSS = norm(yhat-mean(y))^2;  %
Regression sum of squares.
  TSS = norm(y-mean(y))^2;     % Total sum
of squares.
  r2 = RSS/TSS;              % R-square statistic.
  if (p>1)
    F = (RSS/(p-1))/s2;      % F statistic for
regression
  else
    F = NaN;
  end
  prob = 1 - fcdf(F,p-1,nu);  % Significance
probability for regression
  stats = [r2 F prob];

  % All that requires a constant.  Do we have
one?
  if (~any(all(X==1)))
    % Apparently not, but look for an implied
constant.
    b0 = R\(Q'*ones(n,1));
    if (sum(abs(1-X*b0))>n*sqrt(eps))
      warning(sprintf(['R-square is not well
defined unless X has' ...
                 ' a column of ones.\nType "help
regress" for' ...
                 ' more information.']));
    end
  end
end

% Find the standard errors of the residuals.
% Get the diagonal elements of the "Hat"
matrix.
% Calculate the variance estimate obtained by
removing each case (i.e. sigmai)
T = X*RI;
hatdiag=sum((T .* T)',1)';
ok = ((1-hatdiag) > sqrt(eps));
hatdiag(~ok) = 1;
if nu < 1,
  ser=rmse*ones(length(y),1);
elseif nu > 1
  denom = (nu-1) .* (1-hatdiag);
  sigmai = zeros(length(denom),1);
  sigmai(ok) = sqrt((nu*s2/(nu-1)) - (r(ok) .^2
./ denom(ok)));
  ser = sqrt(1-hatdiag) .* sigmai;
  ser(~ok) = Inf;
elseif nu == 1
  ser = sqrt(1-hatdiag) .* rmse;
  ser(~ok) = Inf;
end

% Create confidence intervals for residuals.
Z=[(r-tval*ser) (r+tval*ser)]';
rint=Z';

% Restore NaN so inputs and outputs conform
if (nargout>2 & any(wasnan))
  tmp = ones(size(wasnan));
  tmp(:) = NaN;
  tmp(~wasnan) = r;
  r = tmp;
end
if (nargout>3 & any(wasnan))
  tmp = ones(length(wasnan),2);
  tmp(:) = NaN;
  tmp(~wasnan,:) = rint;
  rint = tmp;
end
```

AUTHOR'S PUBLICATIONS

List of Publications

[1] K. R. Kashwan, M. Bhuyan, *Robust Electronic-Nose System with Temperature and Humidity Drift Compensation for Tea and Spice Flavour Discrimination,* IEEE Conference, AsiaSense-2005, Asian Conference on Sensors And International Conference on New Techniques in Pharmaceutical and Biomedical Research, ISBN: 0-7803-9371-6, September 5-7, 2005, Kuala Lumpur, Malaysia.

[2] K. R. Kashwan, M. Bhuyan, *Tea Flavour Discrimination by using Electronic-Nose Sensors and Artificial Neural Network Pattern Recognition Techniques,* BIOMED 2005, National Conference with ISBN Proceedings on Emerging Trends in Biomedical Instrumentation, April 20-30, 2005, Birla Institute of Technology, Mesra, Ranchi, 835215 (India)

[3] K. R. Kashwan, M. Bhuyan, *Determination of Drift due to Temperature, Humidity and Pressure in the Samples for MOS based Electronic-Nose Sensors* BIOMED 2005, National Conference with ISBN Proceedings on Emerging Trends in Biomedical Instrumentation, April 20-30, 20052005, Birla Inst. of Tech., Mesra, Ranchi, (India)

[4] Ritaban Dutta, E. L. Hines, J. W. Gardner, K. R. Kashwan, M. Bhuyan. *Tea quality prediction using a tin oxide based electronic nose:* Elsevier Science Journal: Sensors and Actuators B 94(2003) Volume 94, Issue 2 (1 September 2003) Pages: 228-237. Elsevier Science Ltd. Oxford, UK

[5] Ritaban Dutta, K. R. Kashwan, M. Bhuyan, E. L. Hines, J. W. Gardner *Electronic Nose based tea quality standardization,* Neural Networks Special issue, Volume 16, Issue 5-6 (2003) 847-853, Elsevier Science Ltd, UK.

[6] Ritaban Dutta, E. L. Hines, J. W. Gardner, K. R. Kashwan, M. Bhuyan. *Determination of tea quality by using a Neural Network based Electronic Nose*, International Joint Conference on Neural Network (IJCNN 2003), Portland, Oregon, USA. IEEE Conference sponsored by IEEE Neural Network Society and International Neural network Society, July 20-24 Vol. 1, pp. 404-409, 2003, USA

[7] K R Kashwan, M. Bhuyan, *Electronic-Nose Sensors and Artificial Neural Network Pattern Recognition Techniques to Discriminate Non-Overlapping Tea Flavours* Elsevier Science Journal: Food Science and Technology, LWT, Submission No.: Y/FSTL_332, UK (Communicated)

[8] K R Kashwan, M. Bhuyan, *Spice Aroma Discrimination by using Electronic-Nose and Artificial Neural Network based Pattern Recognition* Elsevier Science Journal: Journal of Food Engineering, Manuscript No.: O-05-1470 UK (Communicated)

[9] K. R. Kashwan, M. Bhuyan, *Determination of Drift in the Samples for MOS based Electronic-Nose Sensors,* Elsevier Science Journal: Sensors and Actuators B Chemical, Manuscript No.: SNB-D-05-00078, UK (Communicated)