Artificial Intelligence Applications in Waste Water Monitoring for Industrial Purposes

by

Nunthika Benjathapanun

A thesis submitted to City University for the Degree of Doctor of Philosophy
in Electrical, Electronic and Information Engineering

CITY University

Department of Electrical, Electronic and Information Engineering
Northampton Square, London EC1V 0HB, UK.

November 1995

To my mother and my father.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I wish to express my thanks to Professor K.T.V. Grattan for his advice and guidance throughout this work.

I am deeply indebted to Dr. W.J.O. Boyle for his supervision of this work and also for providing his valuable time to discuss problems of all sorts and also for his patience in reviewing this manuscript.

I wish to acknowledge the support provided by the Ministry of Science through King Mongkut's Institute of Technology, Ladkrabang Campus, for allowing me to study for this degree through a grant.

I gratefully appreciate the sharing of expertise and the many constructive comments made by Dr. Z. Mouaziz, and Mr. Brian Burns during the years of working with the Water & Environmental Instrumentation group at City University. Also, I would like to thank the secretarial and technical staff for their help.

Also, I grateful to Mrs. R. M. Simon for her sympathy and encouragement throughout the years of my studies. I also would like to thank Dr. G. Kearney for his suggestion as well as reviewing the manuscript.

Special thanks are given to Thai friends; Dr. P. Yupapin, Dr. R. Chitaree, and Dr. T. Wongcharoen, who have been with the author over this period of help and joy.

Finally, my deepest gratitude is devoted to my mother, father, sisters, and brothers for their loves, patience, and encouragements throughout my studies.

Abstract

This thesis reports on work carried out in the development of software for artificial intelligence sensing systems based on UV-Vis spectroscopy, designed for remote online and real-time analysis for monitoring of industrial effluent. A feasibility study on artificial intelligence methods and the design of an intelligent monitoring system has been researched. This system is capable of detecting the occurrences of chemical pollutants and the concentration of species involved.

The controlling software was developed in this work for the remote modem control of a computer controlled UV-Vis spectrometer system. This provides facilities for signal processing, data storage, and transfer of data to a host machine for real-time analysis. This gives significant advantages in term of automatically and instantly reporting of a pollution incident. This front end sensing system has been installed at industrial sites in order to demonstrate the apparatus in the real situations and to obtain data for qualitative analysis.

Difficulties in working with the above data pointed to the need for a laboratory-based evaluation and modelling of analysis methods. This evaluation and development of suitable methods forms a major part of the work. The samples prepared for a set of data were mixtures of nitrate, hypochlorite and ammonia in various concentrations representatives of that expected in real outflows. This data set presented several significant problems in data analysis, including an overlap of UV absorption bands and the interaction between ammonia and hypochlorite to form monochloramine which has its own specific spectral features.

In the evaluation, the spectral data obtained were analysed by two different methods. The first was Principle Component Analysis (PCA) which is based on linear multivariate analysis, and samples were investigated to compare the effects of interactions between components. The second method was Neural Network analysis, which is a non-linear analysis technique. After considerable effort, this approach resulted in a data analysis scheme where the Back-Propagation algorithm was used as a two-step process. In the first-step, the network inputs were derived by binary encoding segments of the second derivative of the absorption spectra according to their shape and the network outputs specified according to which species were likely to occur. As a result, the second-step network could then focus on a few inputs that strongly correlate with the presence of the expected species. Also the second-step provided a filter that compensated for false classification of species, at low concentration levels. The resulting data analysis scheme depends on a knowledge of the expected chemistry for implementation: however it gives a much better performance than PCA in this particular case.

The complete monitoring system has been integrated with a Graphical User Interface software to perform real time analysis at a host machine. A multi-task system for online monitoring of data transmitted from a remote site, has been developed, based on the neural network approach. Finally, the intelligent monitoring system is demonstrated and evaluated.

Declaration

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgements.

List of Symbols and Abbrevations

Symbols

A	Absorption
I	Intensity
l	Path length
[D]	data matrix
$[D]_N$	normalize data matrix
[Z]	covariance matrix
$[Z]_N$	correlation matrix
λ	Wavelength in nm
w	weight
ε	Extinction coefficient
c	concentration
δ	error parameter
η	learning rate
α	momentum coefficient
o_k	actual network output of kth node
t_k	target output of kth node
θ	neural node bias
Cl_2	chlorine
NH_3	ammonia
NO_3	nitrate
NH_2Cl	monochloramine
NO_3^-	nitrate ion
NH_4^+	ammonium ion
OCl^-	hypochlorite ion

Abbrevations

A/D	Analog to Digital
ANN	Artificial Neural Networks
BitBlt	Bit Block Transfer
BP	Backpropagation algorithm
BPNN	Backpropagation Neural Networks
CS	Code Segment register
DDE	Dynamic Data Exchange
DDEML	Dynamic Data Exchange Management Libraries
DLL	Dynamic Link Library
DS	Data Segment register
GDI	Graphical Device Interface
GUI	Graphical User Interface
ICA	Instrumentation, Control and Automation
IR	Infrared
LVQ	Learning Vector Quantization
MB	Mega Bytes (10^6 Byte)
MS-DOS	MicroSoft Disk Operating System
NIR	Near Infrared
NN	neural network
OLE	Object Linking and Embedding
OMNIS	Optimal minimal Neural network Interpretation of Spectra
PatBlt	Pattern block transfer
PC	Principal Component
PCA	Principal Component Analysis
PCR	Principal Component Regression
PDA	Photo Diode Array
pH	Logarithm of the concentration of hydronium ion
PLC	Programmable Logic Control
PLS	Partial Least Square
ppm	part per million

SS	Stack Segment register
TOC	Total Organic Carbon
TTFA	Target Transform Factor Analysis
UV	Ultraviolet
UV-Vis	Ultraviolet and visible
WAN	Wide Area Network
WWM	Waste Water Monitoring
XOR	Exclusive OR logic function

CHAPTER 1

Introduction

"Consider the past and you shall know the future"

Chinese Proverb

1.0 Introduction

Current analysis of water quality relies mainly on batch sampling and laboratory-based measurement. The result of chemical analysis of many samples is regularly used to determine whether there are possible harmful effects on humans, fish, and agriculture, in particular. However, where such methods are used, at the time when the pollution has been detected there is the risk that the contaminants might have already spread widely into the surrounding environment. Such a recent incident was reported in *The Times* newspaper on the 18th May 1994, when a quantity of caustic soda leaked from a tank at an industrial site into the river Ellen in Cumbria, causing the death of thousands of fish over a ten mile stretch of the river and illustrating the nature of the problem.

Water pollution can occur at any time. Unless pollution officers are swift to arrive at the scene, remedial measures may be too late and the pollution source may be difficult to identify. Effective monitoring is, by definition, continuous monitoring with instant and automatic reporting of incidents. On the other hand, the process industries have to be responsible for the production of waste by-products from their activities, some of which are in a liquid form that must be disposed, with or without prior treatment. The usual method of disposal of liquid waste, or effluents, is to discharge them into the local sewerage system or water course. With such a method of disposal there exists the possibility that undesirable substances could be discharged accidentally. However, the pollution incidents in a local system are easier to detect and less detrimental to the environment than those in a river stream.

There is a definite trend in the water industry to increase the use of instrumentation in the operation and monitoring of water courses, and in water treatment. These are several reasons for this:

- the growing consciousness of environmental issues: The various inspectorates and health and safety organizations have increased powers, the "green" lobbies have more voice and compliance with directives is becoming more closely inspected, for the release of toxic substances.

- the cost benefits of accurate process control: The privatization of the industry has resulted in more concern about accurate process control that would reduce the costs from raw material loss. The chemical industries are now required to show a higher degree of financial viability.

- the pushing of "High Tech": There is a "High Tech" push in industries with the prospect of cheap, remote sensor systems using microprocessor controllers and modern telemetry, to aim to decrease exposure to hazardous conditions. The remote location of instruments enables the monitoring of water systems in real time with a distributed processing system to enable there to be a more rapid response via control loops.

In summary, the aim of the work in this thesis is to develop an intelligent monitoring system that is capable of detecting the occurrences of chemical pollutants and to determine the concentration of each species involved. Once detected, action could then be taken to minimize the environmental impact of such an incident, ideally by tightening control of the processes involved and thus assuring for the prevention of the further discharge of contaminants from the site.

The initial task is to survey current instrumentation, as described in the rest of this chapter in the following, where in sections

1.1: the instrumentation status, existing sensors, support systems are surveyed,

1.2: the constraints of an intelligent monitoring system are summarized,

1.3: the relevant data analysis techniques are surveyed,

1.4: the aims and objectives of the work are discussed and

1.5: the structure of thesis is summarized.

1.1. Instrumentation Status

> *"you always have to have one eye open to the question: what can technology do? ... And one eye open to the question: what are people doing and how would this fit in? What would they do with it?"*
>
> Terry Winograd. Human-machine interaction[1]

The market for instruments to monitor water is vast and likely to grow under pressure of increasing environmental legislation. However, the performance of many systems has been widely reported as unsatisfactory, and the need for more research and development is pressing. In most applications, instrumentation, control and automation (ICA) equipment, especially front end sensors, perform less than satisfactorily. Faults can often be attributed to poor equipment technical quality and often to the type of personnel expected to use and maintain it. In most cases, there are insufficient qualified operators of systems. Also, systems can be incorrectly specified because senior management is not clear about how to use the acquired data. At the same time, the service and maintenance requirements of front end sensors are often beyond the capabilities of the available staff. As a result, some of the instruments developed for water monitoring have been withdrawn from service before proper assessment was made or allowed to deteriorate beyond repair, making matters worse.[2] In general, current instruments for water quality monitoring will not operate reliably without regular attention. Even if automatic cleaning and calibration procedures are provided, reagents must be used when required and regular maintenance and inspection carried out. Most instruments require weekly attention and the true cost of ownership this implies must be recognized and catered for.

One of the biggest drawbacks with existing technology is the need to transport bulky, heavy reagents from base to the field equipment. Although the interval between servicing can be extended, the possibility of reagents and calibration solutions deteriorating must be recognized. In general, most water monitoring sensors are either electrochemically or optically-based. In either case, problems of basic stability and adverse environment conditions have not always been recognized and catered for. An exception, but one which involves significantly increased cost, is the use of dual sensor systems in which two identical sensors are deployed, one being on-stream and one on stand-by, immersed in cleaning and calibrating fluid containing a biocide. For such systems, current readings are compared with past data trends and if data are found to be outside pre-determined limits, the sensors are transposed. If the reading is confirmed, a water quality alarm is raised and if not, the other sensor is left on-line and an instrument fault alarm is raised.[3]

1.2 Optical Based Sensors and Support Systems

The measurement of the composition of bulk matter is very important in various scientific and industrial fields. The quantitative and qualitative analysis required for this resides in a knowledge of the different chemical parameters forming the different constituents and their concentrations. There are now many methods for the measurement of the concentration of species and the instrumentation involved is in a state of continual redesign prompted by technological advances. This is in response to a more demanding need and thus a scientific framework on which analytical techniques are based.

Existing analytical methods rely on the transducing effect generated when a parameter is subjected to a physical, chemical or electrical disturbance or a combination of these. Each parameter possesses particular properties which in turn identifies it and thus makes it distinguishable among others. Consequently, instruments based on the properties of the specific species may, in principle, be developed to suit particular applications.

Most of the on-line instrumentation is designed for a major effort centred around calorimetric and electro-chemical techniques.[4] For example, calorimetry is currently the main method for determining total organic carbon (TOC)[5] and ion-selective liquid ion-exchange electrode[6] sensors have been developed for chlorine, nitrate and ammonium ions. Colorimetric techniques have proved to be accurate and reliable for the determination of various species in water.[7] In water, most species absorbing in the ultraviolet and visible parts of the spectrum have broad absorption peaks with peak widths of typically 20 nm or more. Such broad absorption peaks lead to difficulties in identifying different species in mixtures which may absorb at the same wavelength or swamp the absorption of a small signal at a nearby wavelength. To overcome these, chemical methods can be employed where the species involved is removed or modified chemically and a ratio measurement with and without the species of interest carried out. For example, the measurement of residual chlorine[8] may be carried out by adding NaOH to convert dissolved chlorine (Cl_2) and hypochlorous acid (HOCl) to hypochlorite ion (OCl^-). The measurement of the hypochlorite ion is made at 290 nm, which is far from the nitrate absorption peak. The measurement is taken again after adding Na_2SO_4 which converts the hypochlorite ion to Cl^-. The latter does not absorb in UV region. This method provides a specific sensor for a particular species but requires the addition of reagent which increases service and maintenance requirements.

An interesting system for remote spectrophotometric monitoring of industrial waste waters has been developed by Danigel *et al.* at Messtechnik und Automation, Ciba-Geigy AG, Switzerland.[9] This system consists of a personal computer linked to fibre-optical spectrophotometers with flow-through measuring probes and the appropriate control software. The system used a computer to regulate sampling and cleaning of the probe by flushing it with pure water over some period which is determined by reference measurement. Eighteen wavelengths are chosen between 250 and 1050 nm to characterize the spectra. The data define an '*alarm*' when the absorption measurement exceeds the predefined level. This does not need the addition of reagent, but it does not specifically identify the species or determine the level of contamination.

The work of this thesis is done under the constraint that the monitoring system must be capable of being operated automatically, with incidents reported "instantly" and no reagent adding for the determination of interferences in differential spectroscopy. A range of alternative techniques may be deployed to extend intervals between service visits and to eliminate the need for transporting reagents. These constraints cannot effectively be realized unless:

- an automatic system is employed, remote controlled by computer via, for example, a public telephone line,
- interference absorption bands are discriminated by mathematical analysis techniques or the use of artificial intelligent techniques and
- the incident is reported in real time via the system.

In this study, a low resolution UV-Vis spectrometer is used as a front end sensor. Such a spectrometer is controlled by computer which is under remote control from a host machine via a modem. The data are gathered by the front end machine and then sent to the host machine for real time analysis and implementation. Two techniques for analyzing spectral data based on linear and non-linear methods are surveyed, as described in the next section.

1.3 Real-Time Monitoring

This approach described here utilizes a rugged diode array spectrophotometer, controlled by a single microprocessor board, which scans throughout the UV-Vis wavelengths and which is controlled remotely via a modem. Automatic calibration, warning, alarm and malfunction signals are provided for on-line monitoring. Also involved is the utilization of data analysis techniques in order to identify and quantify the concentrations of specific pollutants, in the presence of organic and suspended matter of the type found in rivers and effluent.

In this system, *"breakthrough"* of contaminants or other changes in water chemistry are registered as a change in the optical intensity and the observed spectral content. Many chemical species affect the UV-Vis spectrum directly: for example, chlorine in its various forms, ammonia, nitrate ion, carbonate ion, and aromatic organic matter. Turbidity can be monitored due to its broad-band effect on overall spectral intensity.

Advances in automatic sensing systems make possible the collection of large quantities of data. If the important information can be extracted easily and quickly, this will potentially improve the quality of products and reduce the waste of raw materials. Such a timely interpretation is possible but may be failured by such problems as undetected sensors failures, sensors going out of calibration, the integrity of data classification, the use of data reduction techniques and general human errors. It is no wonder that data analysis methods, in the face of these problems, often are seen to be inadequate. Without proper correlation of data with the different possible states of the sample water, the data bank can continue to grow without the garnering of any useful information.[10]

Spectroscopic methods generate spectral information which can be represented by an array of data. For each wavelength there is an associated datum value, which is referred to an independent variable. When multivariate data for a number of samples are available, the data may be arranged into a matrix with one row for each sample; each column containing the values of a given variable for all samples. This data matrix is the basis for all such multivariate data analysis methods.

There are several approaches to the general problem of interpreting sensor data including the use of statistical techniques and neural networks to determine a correlation between sensor data and the variables of interest.

One of the main expectations of the work is that the analysis of spectroscopic data using Principal Component Analysis (PCA) and/or Neural Network Analysis would yield methods that are directly applicable to the monitoring of any industrial waste water. These techniques are expected to replace the reagents normally used to remove chemical species which cause the overlapping of absorption peaks. These two techniques are investigated in different approaches considered: PCA is based on the linear multivariate analysis while Neural Network is based on non-linear analysis.

PCA is a multivariable, linear data analysis technique, that can be used to reduce the number of variables in a given data set without the loss of information. Most of the analytical methods explored have used PCA in order to reduce the n-dimensional-space representation of the data accumulated to a first few significant Principal Components (PCs). Consequently, these PCs are interpreted in their physical or chemical meaning by the other methods. The PCA technique can be used to interpret complicated data matrices to extract useful information related to the problem. For instance, PCA was used to investigate the origin of the effluents from Lake Saimaa, Finland[11] by analysing the water quality monitoring data. PCA has had wider application tackling problems such as the classification of authentic currency and other paper stock,[12] the grading of apples in classes of sugar content,[13] and the classification of alloys in terms of different metal contents.[14] For quantitative analysis, the significant factors obtained using PCA are then analyzed using *Target Transform Factor Analysis* (TTFA) which transforms these PCs into a data matrix with values close to the actual data matrix by mean of a least squares fitting. The calibration matrix from a training data set is then generated. The statistically identical technique is called *Principal Components Regression* (PCR). This technique has been widely used in biochemistry,[15] pharmaceutical,[16-19] and food industries[20] for quantitative determination of several components in a mixture under investigation.

The results from the above reported examples have been very good since the PCA analysis used strictly linear data obtained using precise chemical preparation of the samples and in ideal laboratory conditions. In many cases, neural networks are used to compensate the non-linearity such as was done by Gemperline *et al.*[21] who compared the PCR and Neural Network in pharmaceutical product analysis using UV-Vis Spectroscopic data. This study concluded that the PCR approach should be expected to give the best performance where a strict linear data set is observed and neural networks may be capable of giving superior performance when a nonlinear response due to solute interactions is present. There are also examples of successful experiments in analytical chemistry[22-37] and water pollution researches[38-40] using neural networks for the qualitative and quantitative analysis of multicomponent samples.

Most of the above examples which have application in the pharmaceutical products or food industries represent a more controlled situation than a typical waste water system. Waste water is a very complex mixture composed of the chemical substances used in a factory, chemicals in the water supply, and bio-organic material from sewage systems. There are certainly significant interactions between these components such as ammonia from sewage systems interacting with chlorine from a cleaning agent. There has been no research reported which was based on using the above analysis methods in such a complex system. However, there exists a wide variety of different process industries, all discharging effluents which are often of unique composition. These may be amenable to specific modeling by generating and observing a large number of samples of varying composition. In this study both techniques are investigated to obtain the best performance.

1.4 Aims and Objectives of this thesis

The current research is targeted at the potential use of optical methods for the on-line measurement of the chemical concentration of species dissolved in aqueous solution. The concentration of the chemical species is related to the variation of the light intensity of a particular sensing wavelength through the Beer-Lambert law. Existing instrumentation based on optical spectroscopy has mostly been directed towards laboratory use and has seen limited applications outside. From the research carried out in the water industry, it is clear that the state of the current instrumentation is not adequate to cope with the increasing demands. Indeed, there is a lack of instruments for the monitoring of species that are widely experienced and problems arising with the use of individual instruments for a particular species because of the use of different reagents. Nonetheless, there does not exist a monitor with systematic remote control with real-time analysis as far as is known at present.

Thus the specific aim of work in this thesis is to develop an intelligent monitoring system that is capable of detecting the occurrences of particular chemical pollutants and the determination of the concentration of each species. Once detected, action could then be taken to minimize the environmental impact of a pollution incident, ideally by controlling the process and preventing the further discharge of the contaminant from the site. There exists a wide variety of different process industries all discharging effluents which are often of unique composition. One common factor among many effluent discharges is that they are based upon water originating from various aspects of the process, which contains many known chemical species dissolved or suspended in it.

In addressing the need, sensing systems can be considered as consisting of several key components which, when combined, form a complete measurement system aimed at a specific measurement task. Many of the key components of such a system are developed in this work, for instance the computer interface to the sensing head, software drivers for control of electronic interfaces, data acquisition and storage, data transfer for remote analysis, data processing and decision making methods, and the user interface. This leaves only the optimization of the physical and chemical sensing elements/heads requiring a major development effort for a specific sensor system.

The aims of the work proposed for this project address the development of the key components which may be summarized as follows:

- Analysis of the basic chemical properties underlying the absorption of light by the molecules and establishing a quantitative relationship between them.

- Creating a software driver for an automatic control of a diode array spectrometer. The tasks here are initializing a spectrometer, gathering signals, converting to spectral data, their basic implementation, basic decision making, recording data, and communication with the host machine.

- Industrial evaluation. The system is installed at two different industrial sites, one with a treated outflow and the other with an untreated outflow. These were carried out and the two experiences accessed in term of the long term operation of the system in the industrial context.

- Design and implementation of a Graphical User Interface (GUI) software system coupled to a multitasking system for paralleled monitoring of data from remote sites. It is designed to gather raw data via communication lines, dynamic linking to another application for real-time analysis, indication of alarm situation, and providing a friendly user interface.

- Investigation of the chemometric techniques in absorption spectroscopy to determine the composition. Two techniques are used: Principal Component Analysis (PCA) and Artificial Neural Networks.

The work which has subsequently formed the overall content of the project is described in three parts outlined below.

- In the first part, a knowledge of spectroscopic theory was used to design the functional aspects of a remote monitoring machine. In this approach, a program was created to control a remote sensing system comprising a photodiode array (PDA) spectrometer and a programmable UV-Vis deuterium lamp. The software provides signal processing, the display of a spectral data plot, and data storage. It also checks a telephone link for transferring the data. In preliminary work this automated system was installed at two different industrial sites involving both treated and untreated outflow. These two field trials were carried out in order to assess the problems encountered during the long term operation of the instrument. The results of these trials gave valuable lessons to improve the system to simulate the real monitoring situations which differ from those in the laboratory.

- In the second part, measurements were carried in the chemical laboratory which involved known samples. These samples were prepared after consultation with the chief chemist on the industrial sites involved, after collecting a large number of spectra from these sites. The samples were the mixture of nitrate, chlorine and ammonia in various concentrations. These three components absorb UV with overlapping absorption peaks. In this part of work, the spectral data were analyzed by two methods. In the first, a conventional spectroscopic technique, PCA, was used to classify the chemical species in the mixture solution. The PCA results were then used together with Target Transformation

Factor Analysis to determine the concentration of each species. Another technique employed was the use of Neural Networks, which are now widely used in various research schemes. A range of alternative preprocessing techniques were investigated to determine their comparative effectiveness.

- In the third part, A GUI-software system was designed and implemented for a host computer. The approach taken for data analysis used Neural Networks which were found to have distinct advantage over the PCA technique. This software was created in Borland C^{++} and ran under Microsoft Windows with other two windows applications; a "*Dynacomm*" communication package and a "*Neural Desk*" neural network package. The whole system has been designed with the constraints of an intelligent controlled, friendly-user interface, and real-time analysis with dynamic linking. This resulted in a fully automated system in which Dynamic Data Exchange (DDE) causes the data to pass between the applications. The implementation is easy and lively with the animation graphics for real-time analysis.

1.5 Structure of the Thesis

The thesis is structured in the following way.

A brief introduction is given in Chapter 1, which reviews the necessity for intelligent monitoring instrumentation. This is carried out with an emphasis on the drawbacks of existing instruments as outlined in "*a guideline of the Water Industry Steering Group* *on ICA equipment*"[2] and the use of optically based sensors. A review of different techniques used in spectroscopic data analysis is presented. The aims and objectives of this work are also briefly previewed here to show how such techniques may fill the gap between real requirements and measurements as provided by current instrumentation.

The relevant theoretical background on the molecular absorption related to the qualitative and quantitative of chemical components is presented in chapter 2, together with samples of the typical spectrophotometrically-assessed chemistry experienced in the water industrial monitoring environment.

In Chapter 3, the front end sensing system is presented. A number of problems from experimental trials at industrial sites are discussed, implying an improvement in the front end equipment. The details of phenomena experienced in industrial process control are also surveyed. Other work that was carried on the same experimental data by a co-worker is reviewed and some conclusions relevant to this work reached.

The implementation of a Graphical User Interface (GUI) for Artificial Intelligent On-line Monitoring based on this feasibility study is demonstrated in Chapter 4. The design of a system for parallel monitoring to a remote site is described. This includes the use of Multi-tasking, Dynamic Data Exchange (DDE), Dynamic Link Libraries(DLL) and Memory Management between the windows applications, which run concurrently, which are briefly described. A User manual for the software is provided in this chapter, whereas the technical reference manual is provided in an appendix F.

In Chapter 5, a feasibility study on the use of PCA techniques for classification and quantitative analysis is performed. The ability of PCA to identify the presence and level of nitrate, ammonia and chlorine solutes in sample solutions is discussed. This feasibility study takes two approaches, one involving linearity of sample data and the other non-linear data. The mixture of nitrate with chlorine, and nitrate with ammonia are taken as examples of linear behaviour because there is no chemical interaction between each species. Non-linear data from mixtures of interacting chemicals, ie. between chlorine and ammonia, are discussed.

An investigation of the use of neural networks with various preprocessing techniques is described in Chapter 6. In investigating the networks a variety of network size, the number of data patterns, learning algorithms, and learning parameters are described. An analysis of the results from each technique is provided and error considerations are discussed. These are optimized for the best performance at predicting the constitution of unknown water samples. A conclusion on the use of such techniques is reached and delight of performance is given.

Finally, the general conclusions relating to the use of computer control in UV-Vis spectrometry at a remote site and the use of data analysis techniques are given in Chapter 7. The advantage of using Neural Networks to classify and to determine the level of the species of contaminant is discussed. The extension of this work for other applications in a water chemistry study is suggested.

CHAPTER 2

Theory of UV-Vis Spectrophotometry

"All instruction given or received by way of argument proceeds from pre-existent knowledge."

Aristotle, Posterior Analytics, Age of Intelligent Machine.[41]

2.0 Introduction

The interaction between light and molecules of a chemical species occurs through the process of absorption and scattering, ignoring refraction and reflection at the interface. The intensity of light that passes through a sample may be reduced with this being dependent on the type and concentration of chemical species dissolved in the sample. The relationship between the species under investigation and the changes of light intensity is mathematically determined using the Beer-Lambert law, under appropriate conditions. The Beer-Lambert law linearly relates the changes in intensity of the light beam travelling through the sample to the concentration of a chemical species in solution. This law is also valid for mixtures of absorbing species which can be treated as separate components providing that there are no mutual interactions.

This chapter discusses the theoretical aspects of UV-Vis spectroscopic instrumentation as applied to the measurement of contaminants in water. For the best performance, the various sources of error must be taken into account, and therefore an analysis of deviations introduced by the absorption law is described. The conventional spectrophotometric analysis and a summary of analyses appropriate to a study of water chemistry used are also given. These are described in the following in which in sections:

2.1 The basics of interaction between UV-Vis light with molecules of chemical species is described.

2.2 The relationship between absorption of light and chemical composition given by the Beer-Lambert law is shown in this section.

2.3 The limitations of the Beer-Lambert law are concerned.

2.4 The causes of instrumentation deviation are discussed.

2.5 The characteristics of an absorption peak is demonstrated.

2.6 The conventional methods for evaluation of the analyte concentration from spectrophotometric data are described.

2.7 The resolution of absorption peaks can be improved by differentiation of absorption spectra as demonstrated in this section where its application is described in 2.8.

2.8 Overlapping of absorption peaks can be identified or distinguished in the form of derivative spectra as described in this section.

2.9 Some procedures for water pollutant monitoring, based on spectrophotometry, are summarised.

2.1 Basis of UV-Vis Spectrophotometry

In this study, the monitoring of water contamination is based on the wavelength-dependent absorption of UV-Vis light by molecules of a specific species. In conventional absorption spectroscopy, the sample is placed in an optical cell between a light source and an entrance slit of a spectrometer. When this light impinges on the sample, part of the energy contained in the electromagnetic wave is acquired by the molecule in an absorption process. Such mechanisms of interaction are described in detail in several standard texts.[42-45] Some of the light may be absorbed which relates to the type and concentration of the contaminants. The relationship between absorption of light and concentration of chemical species is given by Beer-Lambert law. Since this plays such an important part in the work a thorough description will be given.

2.2 Beer-Lambert Law

The absorption of incident monochromatic radiation by a sample may be determined by

$$-dI = kI\,dl \tag{2.1}$$

when integrated over the path length*(l)* through which the light travelled, it gives

$$\int_{I_0}^{I} \frac{dI}{I} = k \int_{0}^{l} dl \tag{2.2}$$

or

$$\ln \frac{I_0}{I} = kl \tag{2.3}$$

where I_0, I are the intensities of the incident and transmitted light, and k is a constant that depends on the concentration. Eq.(2.3) can be written as

$$\ln\frac{I_0}{I} = Klc \qquad (2.4)$$

or

$$A = \log\frac{I_0}{I} = K_1 lc \qquad (2.5)$$

where $K_1 = K\ln 10 = 2.303K$ and A is the absorbance. Therefore, at a particular wavelength (λ), Eq.(2.5) can be rewritten as Eq.(2.6) and becomes the formulation form of the Beer-Lambert law.

$$A_\lambda = \log\frac{I_{0(\lambda)}}{I_\lambda} = \varepsilon_\lambda lc \qquad (2.6)$$

where ε_λ represents the extinction coefficient.

The Beer-Lambert law is valid for dilute analyte solutions, using strongly monochromatic light, and in optically homogeneous media. Luminescence and scattering of the solution should be absent. However, the Beer-Lambert law is valid also for heterogeneous media or turbid solutions having small particles (< 0.3 μm), but scattering of light will cause deviation from Beer's law. This law is also valid for mixtures of absorbing species providing that there are no mutual interactions which would contravene the principle of additivity of absorbances, i.e.

$$A_\lambda = \varepsilon_1 lc_1 + \varepsilon_2 lc_2 + \varepsilon_3 lc_3 + ... + \varepsilon_n lc_n \qquad (2.7)$$

where ε_n (n = 1,2,...) and c_n (n=1,2,...) are the extinction coefficients and the concentration of species 1,2,..,n respectively.

2.2.1 Coefficient of Extinction

The coefficient of extinction is the effective cross-section of one mol of absorbing species for radiation of a particular wavelength. This corresponds to the absorbance of a solution containing 1 mol l^{-1} of an absorbing species in a path length of 1 cm. It is easily calculated from

$$\varepsilon_\lambda = \frac{A}{lc} \qquad (\textit{in } l \; mol^{-1} \; cm^{-1}) \qquad (2.8)$$

This relationship is valid for highly monochromatic radiation, so ε will be dependent on the optical bandpass for a certain spectrometers, especially where the analyte has a sharp absorption peak. Hence, when a scan through the wavelength range is carried out and the concentration of the sample is kept constant, the variation of the absorption of the light through the sample at a fixed path length is mainly due to the coefficient of extinction. Therefore this coefficient, which is a function of the wavelength, can be used to determine the concentrations of molecules present in the sample. The most appropriate wavelength to use in an experimental spectroscopic system is where the first derivative of the extinction coefficient as a function of wavelength intercepts the origin.

2.3 Deviation from the Beer-Lambert Law

There are a number of considerable limitations to the applications of the Beer-Lambert law. For best performance, deviation effects must be taken into account. The main effects are discussed in relation to the following situations:

2.3.1 Stray Radiation

Stray light is radiation detected that is different from that selected by the monochromator. This is a problem because both the stray and the absorbing light may fall on the detector which can not differentiate between the two. The apparent absorbance(A') of the solution can be represented by

$$A' = \log\frac{I_0 + I_s}{I + I_s} \tag{2.9}$$

where I_s is the intensity of the stray light. If $I \to 0$ consequently A' →A:

$$A'_{I\to 0} = \log\frac{I_0 + I_s}{I_s} = \log\left(\frac{I_0}{I_s} + 1\right) \tag{2.10}$$

Hence the error caused by the stray light increases for large absorbance values and for a wavelength interval where the primary source has little radiation intensity. Additionally, the stray radiation can limit the maximum absorbance of the instrument to a low value and this maximum absorbance becomes independent of the signal to noise ratio of the detector. For example, with a 1% stray light radiation the maximum absorbance is limited to 2 units of absorption as can be seen by using Eq.(2.10) and replacing I_s by $0.01I_0$:

$$A' = \log\left(\frac{I_0}{I_s} + 1\right) \text{ and thus } A'_{\max} = \log\left(\frac{1}{0.01} + 1\right) \approx 2 \tag{2.11}$$

This result demonstrates the importance of the stray light effect which can, in some cases, limit the performance of a sensitive detector in a particular sensor system.

2.3.2 Changes in the Chemical Equilibrium

Changes caused by the dissociation, association or polymerization of the absorbing species affect the value of the coefficient of extinction. For example, this can be seen through the dilution process of the dichromate ion. In this particular case, the dichromate ion which absorbs light in the visible range of the spectrum (i.e. 450nm) is affected when the solution is diluted since the chemical reaction shifts towards the left of Eq.(2.12).

$$2CrO_4^{2-} + 2H^+ \Leftrightarrow 2HCrO_4^- \Leftrightarrow Cr_2O_7^{2-} + H_2O \qquad (2.12)$$

To avoid the shift of the equilibrium, all the chromium ions should be converted to CrO_4^{2-} by making the solution in 0.05M potassium hydroxide. In mixed solutions, such as those encountered in water monitoring, such changes may be observed with a consequent effect upon the avoidance of a shift of the equilibrium.

In this study, the two major effects of stray light and chemical equilibrium are concerned whereas other effects, which affect linearity, such as non-monochromaticity, the scattering effect, fluorescence effect, and refractive index changes can be found in several standard texts.[42-45]

2.4 Instrumentation Deviations

Such deviations can be caused mainly by the use of unstable light sources, especially discharge lamps such as deuterium lamp. Another source of such deviation is temperature drift as found in laser diodes, where the wavelength shifts as a function of temperature. Other effects, however, can be overcome by carefully choosing appropriate devices in the design of the measurement instrument. Some of the deviations can be compensated in the calibration of the instrument and hence can be included in the consideration of the overall instrument parameters.

2.5 Absorption spectra in the VIS and UV regions

Such spectra relate the absorbed radiation to the wavelength, λ. The electronic transitions which result from absorption are characterized by absorption peaks usually of a Gaussian character. Such spectra thus provide qualitative evidence of the analyte species, its stoichiometry and structure of chemical equilibria in solution. The representation most frequently employed in analytical practice is of A vs. λ, but it should be kept in mind that A is dependent on the concentration of the species and the path length, according to the Beer-Lambert law.

2.5.1 Peak Broadening

Absorption peak profiles in the UV-Vis part of the spectrum are rather wide for aqueous solutions for the species under consideration. Although the absorption in this part of the spectrum is of a discrete nature, it suffers from spectral broadening caused by the interactions with water molecules. Thus, only the envelope can be detected. Moreover, the superposition of the vibrational and rotational absorption leads to an enlargement of the absorption profile. The origin of this enlargement is related to the

Heisenberg Uncertainty Principle where the energy level is blurred, this effect being also known as lifetime broadening. An additional cause of peak broadening is the Doppler shift effect which results from the fact that molecules in solution and especially those in gaseous form are not static and move with high velocities. Thus, a measuring system could be used to detect the light emitted by these molecules as being different since the velocity of each molecule itself is different. Consequently, although these molecules are emitting light at the same wavelength, the light from latter is detected as being of different wavelengths, the total providing an envelope of the profile.

2.5.2 Characteristics of an absorption peak

The Gaussian shape of an absorption peak in the visible and ultraviolet region which is symmetric in terms of $A(\lambda)$ is defined by

(a) the peak height corresponding to $A_{\max}$;

(b) the peak width, δ, for the half-height of the peak maximum which reflects the peak broadening;

(c) $\lambda_{\max}$ for wavelength at the peak maximum is described by Eq.(2.13).

$$A(\lambda) = \frac{A_{\max}}{\sigma\sqrt{2\pi}} e^{\left\{\frac{-(x-\lambda_{\max})^2}{2\sigma^2}\right\}} \tag{2.13}$$

The curve is symmetrical about the wavelength of maximum absorption, $\lambda_{\max}$ and the greater the value of σ, the greater the spread of the curve, as shown in Fig. 2.1.

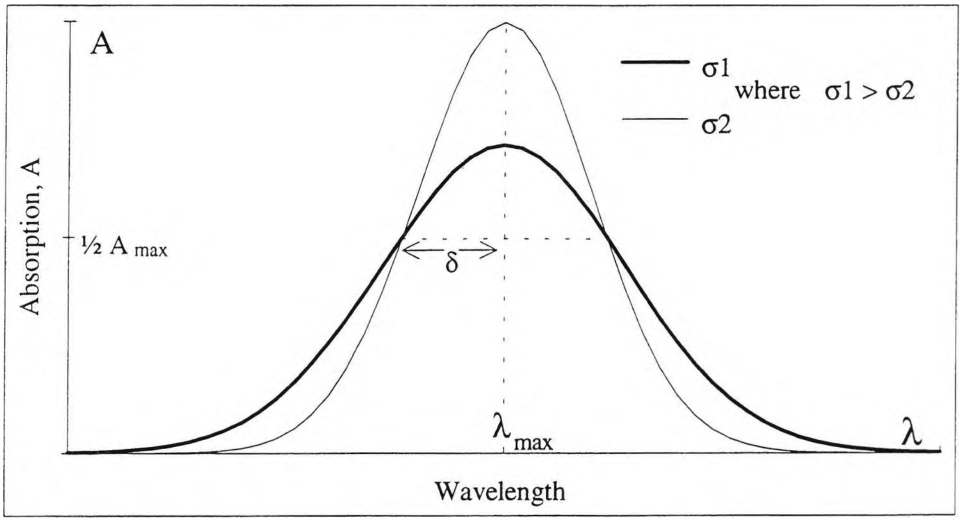


Fig. 2.1 Characteristics of elementary absorption peaks (Gaussian distribution)

λ_{max} and A_{max} are the most important optical parameters of the absorbing species. The correct position of A_{max} may be evaluated with highly monochromatic spectrophotometers at low or zero scanning speed, using measurement "by points" or with the aid of the first or second derivatives of the particular absorption band. The value of A_{max} is considerably diminished if an instrument with a large radiation bandpass is used to record narrow absorption peaks, and thus error can occur in the measurement.

The appearance of two or more absorption maxima on the absorption curve indicates that there is more than one electronic transition of the absorbing species or more than one absorbing species of the particular analyte. The resulting absorption spectrum is the envelope of all elementary absorption peaks in agreement with the principle of additivity of absorbances.

2.5.3 Effect of absorbing blank

A complicated effect is observed if the analyte solution is measured against a considerably absorbing blank, especially in the wavelength region where the sensitivity of the detector decreases. In such a case, the radiant energy impinging upon the detector is low and the proportion of stray light rapidly increases. The shape of the absorption curve is apparently modified and often false absorption maxima may be found.

2.6 Evaluation of the analyte concentration from spectrophotometric data

The determination of an unknown analyte concentration can be carried out with the aid of a particular spectrophotometric procedure under the same conditions, and a suitable treatment of the absorbance data at selected wavelengths. It is reasonable to assume that the Beer-Lambert law is obeyed and that the measurements are of sufficient precision and accuracy. Appropriate calibrations are prepared by use of the standard solutions.

The optimum concentration interval of the analyte corresponds to the strictly linear part of the calibration plot of absorbance *vs.* analyte concentration for which the Beer-Lambert law is strictly obeyed and the measurement precision is acceptable. Since the error of absorbance readings rapidly increases for A>1.8 absorption units (au), the most suitable analyte working interval is related to the range A=0.05-1.80 au.

The number of standard samples or replicate solutions used for the calibration plotting depends on the purposes for which the spectrophotometric procedure is employed and the kind of statistical treatment of the absorbance data. It is typically six points for routine spectrophotometric procedures.[45] A set of at least six standard analyte solutions is used covering the required concentration interval. Usually, the calibration plot is carried out by considering the value and position of A_{max}. The regression calculation of the linear part of the graph is undertaken according to a least squares procedure. The regression coefficient of the plot and the degree of fit of the straight line to the experimental points may be calculated to accomplish a fit to the Eq.(2.14):

$$y = \varepsilon x + A' \tag{2.14}$$

where $y = A$, ε is extinction coefficient and A' the absorbance of blank.

2.6.1 Direct calculation of concentration from the calibration plot

The unknown concentration (c_x) of the analyte may be obtained directly from

$$c_x = \frac{A_x}{\varepsilon l} \tag{2.15}$$

where A_x is absorbance of unknown. The conditional absorption coefficient must be previously determined on the spectrophotometer later used for the sample analysis and under the same conditions. The use of tabulated values of extinction coefficients is not recommended because the conditions under which they were determined are usually not sufficiently well defined.

The linear combination of absorbances for several wavelengths, λ_i for which the Beer-Lambert law is valid is also proportional to the analyte concentration. The evaluation of c_x from the measurements at several wavelengths can be obtained from

$$\begin{aligned} A_{\lambda_1} &= \varepsilon_{\lambda_1} c_x l \qquad \Delta A_{\lambda_1} = \varepsilon_{\lambda_1} \Delta c_x l \\ A_{\lambda_2} &= \varepsilon_{\lambda_2} c_x l \qquad \Delta A_{\lambda_2} = \varepsilon_{\lambda_2} \Delta c_x l \end{aligned} \tag{2.16}$$

Influences of the solution turbidity or of some accompanying absorbing species may be removed in this way.

2.6.2 Multicomponent spectrophotometric analysis

The absorbance exhibited by a system containing several absorbing components is the sum of all individual absorbances at the particular wavelength. If no mutual interactions take place, the principle of additivity of absorbances is maintained

$$A_j = \sum_{i=1}^{n} \varepsilon_{ij} c_i l \quad \text{at } \lambda_j \tag{2.17}$$

where A_j is the absorption of the mixture at the wavelength *j*, ε_{ij} is the extinction coefficient of the component *i* at the wavelength *j* and *n* is number of components. The absorption curve is the envelope of the absorption curves of the individual components, e.g., for two components, nitrate ion and monochloramine, as shown in Fig. 2.2. In such cases, maxima or minima, shoulders and inflections, or mixed absorption maxima may appear on the absorption curves. The additivity of absorbance is easily tested by calculation of the absorbances for mixtures of standard solutions, containing known analyte concentrations. Even if the Beer-Lambert law is not obeyed for one component of the mixture, the sum of the absorbance will still be correct for mixture with a constant concentration of such a component.

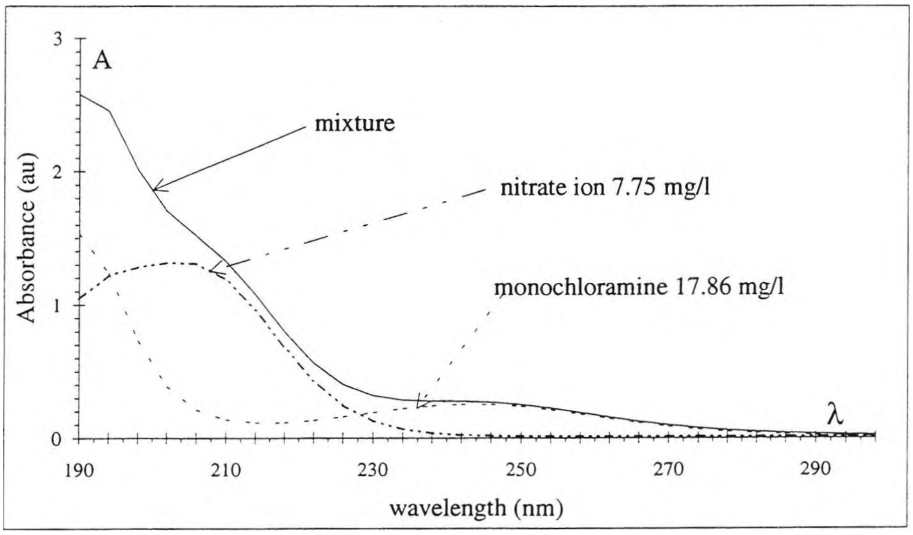


Fig. 2.2 Absorption spectra of the mixture of nitrate ion and monochloramine

The determination of a mixture comprising several absorbing components in solution is carried out by measuring the absorbance at several wavelengths and solving a set of simultaneous equations of the type

$$A_j = \varepsilon_1 l c_1 + \varepsilon_2 l c_2 + ...+ \varepsilon_n l c_n \quad \textit{at wavelength } \lambda_j \tag{2.18}$$

The accuracy and precision of the calculations are dependent on the number of components present and the selection of suitable wavelengths. The absorption coefficients of all the components at the selected wavelengths may be previously determined from solutions of the pure components under the same conditions as for the mixture. In principle, there are two ways to choose the number of wavelengths:

(a) one optimum wavelength is selected for each absorbing component (e.g.,λ_{max}), the number of wavelengths as well as that of the equations for the total absorbance being the same as the number of absorbing components;

(b) the number of wavelengths is so large that the system of equations is overdetermined. The set of equations containing i unknowns may be solved by various numerical methods (determinants, cracowians, iteration procedures, and elimination according to Gauss, etc.). Measurements at a larger number of wavelengths than the number of components involved can be used to advantage. The number of equations for the additivity of absorbances, m, exceeds the number of unknown concentrations of the components, n

$$\begin{array}{l} \varepsilon_{11}c_1 + \varepsilon_{21}c_2 + ... + \varepsilon_{n1}c_n = lA_1 \\ \varepsilon_{12}c_1 + \varepsilon_{22}c_2 + ... + \varepsilon_{n2}c_n = lA_2 \\ \vdots \quad\quad \vdots \quad\quad\quad\quad \vdots \quad\quad \vdots \\ \varepsilon_{1m}c_1 + \varepsilon_{2m}c_2 + ... + \varepsilon_{n,m}c_n = lA_m \end{array} \tag{2.19}$$

where m = the number of measurement at different wavelength,

n = the number of components,

l = pathlength (cm.),

$\varepsilon_{i\,j}$ = extinction coefficient of component *i* at wavelength λ_j, and

A_j = absorbance of the mixture measured at wavelength λ_j.

This overdetermined system of equations is reduced to a set of "normal" equations by means of a least squares procedure or identical statistics method *as "Multiple linear regression*" (MLR).

$$\sum_{j=1}^{m} \left(A_{\exp,j} - A_{cal,j} \right)^2 = \min \tag{2.20}$$

where $A_{\exp,j}$ is a measured absorbance at wavelength number *j* and $A_{cal,j}$ is a calculated absorbance at wavelength number *j*. Wavelengths with low information contents with respect to the components of the system should not be used for the evaluation.[45]

During the multicomponent analysis the concentrations of components may be evaluated even without preliminary knowledge of the numerical values of the extinction coefficients for the components and wavelength used. This can be carried out by using the inverse matrix equation. Additionally, various multicomponent analysis procedures may be used, ascertaining previously the extinction coefficients for all components to be determined at selected wavelengths. For these calculation procedures the necessary ε values are usually evaluated from calibration graphs prepared with pure component standard solutions. The additivity of absorbances for a mixture of absorbing components is often disturbed by mutual interactions which are not always clear in systems containing ions or charged complexes. Thus, increased losses of accuracy are observed for the calculated component concentrations.

However, analysis by means of ordinary multiple regression becomes impossible when the stage is reached where there are only slight differences between the single component spectra. In such a case, a principal component analysis is suitable for modelling the absorbance matrix that results from multivariate calibration measurements into principal components (PCs). These PCs are then fitted to the concentrations of the components by the common regression methods and thus is called Principal Component Regression (PCR). PCR is also often referred to as Target Transform Factor Analysis (TTFA). Sometimes the partial least squares (PLS) analyses are preferable to conventional principal component analysis. The concentration matrix is additionally described by the principal components of the absorbance matrix. Information from the calibration solution can be better used since it reflects a criterion of the similarity of the sample to the calibration set.

2.7 Derivative Spectrophotometry

The differentiation of absorption spectra has considerable advantages for spectrophotometry in the UV-Vis regions. It is the key for the potential enhancement of resolution of overlapping bands. It facilitates the detection of poorly resolved absorption peaks arising from mixtures or impurities in solution. It also enables the exact determination of $\lambda_{\max}$ of the particular analyte species and increases the sensitivity of the spectrophotometric procedures. The shapes of the first four derivatives of the unperturbed Gaussian maximum are shown in Fig.2.3.

The first and each odd derivative pass through zero at the wavelength of the absorption maximum or minimum. A similar shape of the derivative function also appears for spectra where the absorbance of the analyte is overlapped.

The function $\frac{d^2A}{d\lambda^2}$ has a strong minimum for the inflection point of the descending part of the absorbance peak. For a single absorption band, the minimum in the derivative indicates the maximum absorption of the band.

In a composite absorption curve the points of negative inflection of the absorption peaks often coincide with the maxima of the individual bands of the components. The shapes of higher derivatives become increasingly more complex. The strong narrow maximum of the fourth derivative indicates the position of the maximum of the original peak. The use of the 1st or 2nd order derivative can eliminate unwanted effects

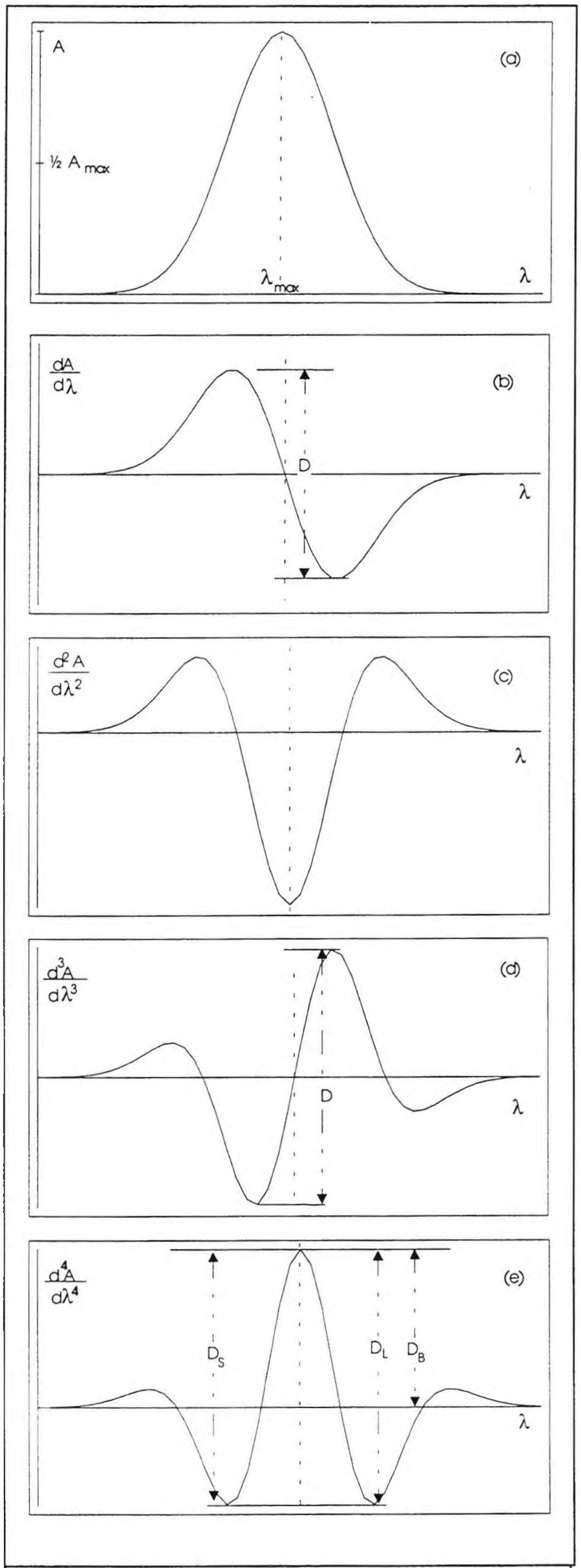


Fig. 2.3 A Gaussian peak and the 1st - 4th derivative, as a function of wavelength, λ.

such as various types of scattering, instrumental effects, differences caused by replacing cells or baseline shifts caused by a continuous background absorption- which all allow a more accurate quantitative evaluation of data. This is especially true for biological samples. In principle, both the peak-to-valley amplitude, D_S, D_L and the baseline-to-valley distance, D_B, are proportional to the analyte concentration, as shown in Fig. 2.3.

In fact, the situation is more complicated for higher derivatives, but the proportionality of the derived signal to the analyte concentration is often maintained. However there are several disadvantages with respect to the evaluation of derivative spectra. The signal-to-noise ratio decreases progressively at higher derivative orders. Low noise amplifiers and high quality differentiators must be used, with some degree of low-pass filtering and smoothing being included to control the increase of noise, especially that of high frequency. The random errors may also be aggravated in the derivative modes.

2.8 Effect of overlapping of absorption peaks

A considerable peak sharpening results upon multiple differentiation such as using $\frac{d^2A}{d\lambda^2} or \frac{d^4A}{d\lambda^4}$. The precise value of λ_{max} is obtained from the point $\frac{d^nA}{d\lambda^n} = 0$, where *n* is an odd number, or from the position of the narrow maximum of the derivative plot (*n* is even) against wavelength for unperturbed absorption peak evaluation.

The form of the derivatives is radically changed if a component peak is not resolved from the main absorption peak in the zero spectrum. In this way, even spectral features or weak shoulders on the main absorption peak can be identified or very similar spectra distinguished. The number, position and form of satellite maxima or minima of the derivative peaks for the main absorbing component are changed in the presence of small concentrations of accompanying components which absorb over the same wavelength range. For example Fig. 2.4 shows the absorbance of mixtures between a fixed value of 7.75 mg/l of nitrate and various concentrations of monochloramine. The concentration of the analyte components is obscured by the large overlapping absorption peaks but this can be solved by using the second order derivative. For the evaluation of a small analyte peak in the presence of a large overlapping band of an interfering species, the first derivative spectra will be more useful and subject to smaller systematic errors.[45]

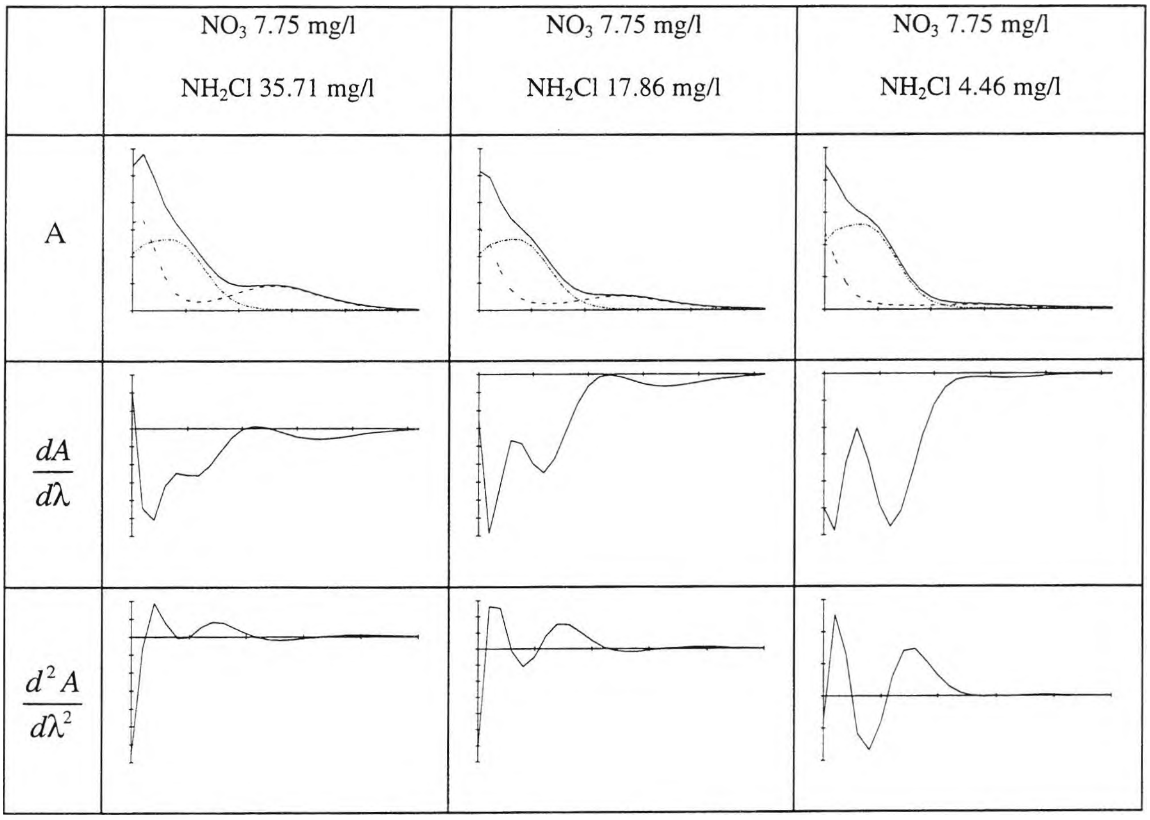


Fig. 2.4 Overlapping Gaussian band pairs and their 1st and 2nd derivatives, as a function of wavelength, λ.

2.9 Spectrophotometry in Water Environmental Chemistry

UV-Vis Spectrophotometry is still widely used for the determination of some harmful inorganic or organic species which are water pollutants. Various important species in water can be determined spectrophotometrically. Examples include F^-, NO_2^-, NO_3^-, NH_3, CN^-, SCN^-, phosphate, phenol, crude oil and products, pesticides and components of industrial waste, and water from domestic sewages. Also advantage can be taken of spectrophotometry in automated interval analysis. Some procedures for water pollutant monitoring may be summarised in Table 2.1.

Table 2.1 Spectrophotometric determination of some pollutants in water[45]

Pollutant	Reagent / Procedure
F^-	Formation of a blue ternary species with Alizarine Complexane and lanthanum[III] in slightly acidic media; measured at 620 nm.
NH_3, NH_4^+	(a) Formation of purple dye with pyrazolone-pyridine reagent; the product is developed in the presence of Chloramine T at pH value of 3.7 and extracted into carbon tetrachloride. (b) Ammonia absorbs in the UV region of the spectrum at 180 nm. The products of reaction of ammonia with chlorine such as monochloramine (NH_2Cl) absorbs UV at a peak centred at 244 nm while di- and tri- chloramine ($NHCl_2$ and NCl_3) do not absorb in the UV. Adding NaOCl to form NH_2Cl and then the measurement of NH_2Cl instead.[46]
NO_2^-	(a) NO_2^- absorbs radiation at 355 nm (ε=23.3 l mol^{-1} cm^{-1}), 302 nm (ε=9.121 l mol^{-1} cm^{-1}) at pH 6 211 nm (ε=5800 l mol^{-1} cm^{-1}) in 0.01 mol l^{-1} OH^- , HNO_2 at pH < 5 absorption peak at 357 nm, twice that of NO_2^- at 355nm

Table 2.1 continued on →

Table 2.1 (continued)

NO_3^-	(a) NO_3^- has λ_{max} 193.6 nm (ε=10^4 l mol^{-1} cm^{-1}), it also absorbs at 198, 203 or 210 nm where interferences from stray light are lower, or Cl^{-1} does not interfere at 210 nm. Corrections of absorbance values against organics are at 275 nm. Solution acidity ($HClO_4$) slightly influences the absorption maxima. In conc. H_2SO_4 λ_{max} is 227-230 nm, where other substances absorb less intensely. For simultaneous determination of nitrite and nitrate in aqueous media measure at 302 and 305 nm. Nitrate does not absorb at 355 nm; For nitrite A_{355}/A_{302} = 2.50. Thus, the absorbance at 355 nm is divided by 2.5 and the value subtracted from that at 302 nm. Nitrate is evaluated in this way. (b) λ_{max} at 200 nm. where sodium nitrate ($NaNO_3$) was mixed with distilled water to provide a nitrate ion concentration of 2 g/l.[46]

Table 2.1 continued on →

Table 2.1 (continued)

Cl_2, HOCl, OCl^-	When chlorine is added to water it reacts to produce dissolved chlorine(Cl_2), hypochlorous acid (HOCl), and hypochlorite ion(OCl^-). The peak absorption of dissolved Cl_2 at 229 nm, HOCL at 233 nm and OCl^- at 290 nm. Convert Cl_2 and HOCl into OCl^- by adding NaOH to modify pH at > 9 Convert OCl ion to be Cl^- which does not absorb in UV region by adding Na_2SO_4.[47]
Phenol	Interaction of 4-aminoantipyrine (4-aminophenazone) with phenol and $Fe(CN)_6^{3-}$ to red-orange quinone-imine in slightly alkaline media, extraction into chloroform, measured at 500 nm.

2.10 Summary

The interaction of light with matter results in the potential for the characterisation of the type of chemical species studied. This property is then used in the analysis of matter by means of absorption of the UV-Vis light travelling through the analytical samples, employing the Beer-Lambert law which is used to relate the concentration of the absorbing species to the detected light intensity. However, this law is affected by both instrumental and "real" factors such as stray light, chemical equilibrium and absorbing blank. Thus all of these deviations are considered in the data analysis for the best performance, as will be described appropriately in Chapter 5 and Chapter 6.

Additionally, overviews of conventional analysis methods which are used to estimate the concentration of involved species, are introduced to familiarise the general ideas of spectroscopic analysis and also to compare with the two methods, PCA and Neural networks, used in this study.

The addition technique, derivative spectrophotometry, is used to increase the resolution of the absorption spectra which is caused by the overlapping of involved species. This technique is used to pre-process the data, and then analysed by Neural Networks. This results in much more improvement of Neural Networks' performance which will be described appropriately in Chapter 6.

Some examples of spectrophotometry in water chemistry are used to understand the chemical properties of the species interest in this study. These are also used in the design of remote sensing system, after consulting with the senior managers at the industrial sites as described in next chapter.

CHAPTER 3

Remote Front End Sensing

"Most industrial accidents today are blamed on human error. We blame the person. But if most accidents are caused by humans then maybe it's not the humans, maybe it's that we aren't designing things appropriately for people to use. We should redesign it with error correcting codes so that it could correct them itself. We should do the same for people. We know that people make errors, so we should design systems so that either people no longer make errors, or the systems are intensive to the errors."

Donald Norman, Human-Computer Interaction[1]

3.0 Introduction

The aim of this aspect of the work is to develop and characterize an intelligent monitoring system that is capable of detecting the occurrences of several important chemical pollutants. Once detected, action could be taken to minimise the environmental impact of such an incident, ideally by controlling the process and preventing the further discharge of the contaminant from an industrial site to a stream. There exists a wide variety of different process industries all discharging effluents, often of unique composition. One common factor among many effluent discharges is that they are often based upon water originating from various aspects of the process under study, which contain chemical species dissolved or suspended in it. In this work, a system using remote spectrophotometric monitoring of the industrial waste water is presented which relies on the fact that contaminants in water alter their spectral absorption. This can then be used to determine quantitatively the concentration of chemical parameters dissolved in aqueous solutions. The relationship between the species under investigation and the spectral absorption is determined using the Beer-Lambert law, as discussed earlier. The spectral absorption in the ultraviolet and visible region is used to investigate the theoretical and practical feasibility of an effective measuring instrument based on the optical properties of the chemical parameter under consideration.

In this Chapter, a "front end" system for the monitoring system that is automatically controlled by computer and also operated under remote control via a modem is demonstrated. The equipment consists of Jobin Yvon diode array spectrophotometer with a photo diode array (PDA) driver, and a deuterium lamp operated with a programmable power supply. This equipment was constructed at City University where the controlling software was developed as part of this programme. In this approach, the automated system was installed at two different industrial sites and operated in collaboration with the Department of Fluid Engineering at Cranfield University. The installation at the industrial sites was carried out in cooperation with a research student at Cranfield University. The two experiments reported were carried out in order to assess the problems encountered during the long term operation of the instrument. In commissioning, several problems were confronted that give a number of valuable lessons to improve the equipment to enable a system to be designed to suit a range of real industrial situations. The cause and the solution suggested to each of these problems is discussed. At the end, a conclusion based on the results of the several experiments undertaken is presented.

3.1 Experimental trials of Optical based sensor systems for water pollution monitoring

The low resolution UV-Vis spectrophotometer is used for the on-line monitoring of treated industrial wastewater. The instrument consists of a Jobin Yvon Spectraview 1D spectrophotometer under the control of an IBM-PC and a Cathodeon deuterium lamp, also under computer control. A ten centimetre path length flow cell is placed between the deuterium lamp and the entrance slit of spectrometer. Initially, the equipment had been tested in continuous use for two weeks at City University. After the system had been seen to operate satisfactorily during the two weeks testing period, it was installed at the industrial sites in order to assess the problems encountered during a long term operation of the instrument in the "real" environment. In collaboration with the Department of Fluid Engineering at Cranfield University, the monitoring system was installed at the outflow of a conventional waste water treatment plant at Colworth House, Unilever's research establishment, at Sharnbrook, Bedfordshire. The main waste products at Colworth house result from food manufacturing pilot process including manufacture of ice-cream and the use of ammonia in cleaning process. The discharge is routinely monitored on-line for a range of parameters including dissolved oxygen, turbidity, and ammonia concentration. The installation of the UV-Vis monitor system at this site provided an opportunity to evaluate the ability of the system to detect pollution event ‘break through’ from the changes in spectra and also to evaluate and further develop its long term performance. The experimental set-up is illustrated in the schematic shown in Fig. 3.1.

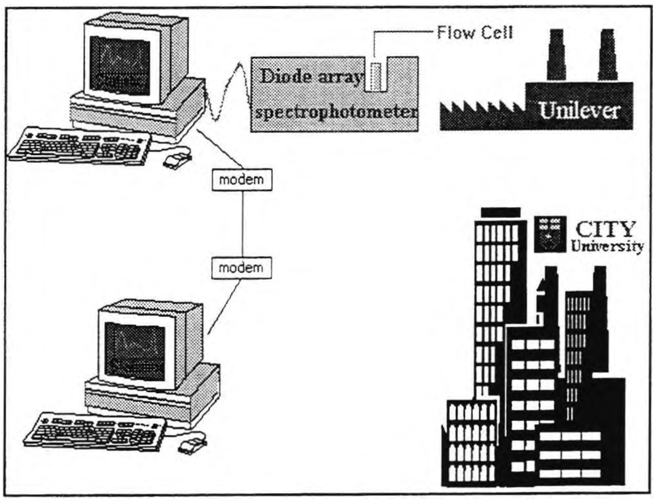


Fig.3.1 The instrumentation set-up arrangement

The instrument is controlled and monitored by software which is developed in Borland C^{++} and run under MS-DOS. The controlling software for the remote instrument is shown as a flow chart in Fig. 3.2.

In the first step, the photo-diode-array (PDA) software driver, provided with the Jobin Yvon spectrophotometer,[48] is loaded as a resident program, thus initialising the spectrophotometer. In the next step, the commercial communication package 'Odyssey'[49] is launched, and within this a script language program loop which runs a proprietary program SCAN.C within a DOS shell.

The script program in Odyssey also services modem inquiries. If a phone call signal occurs, the program scan will be interrupted and Odyssey takes control and attends to the phone inquiry. This allows the transfer of data and the control of the remote computer via the modem.

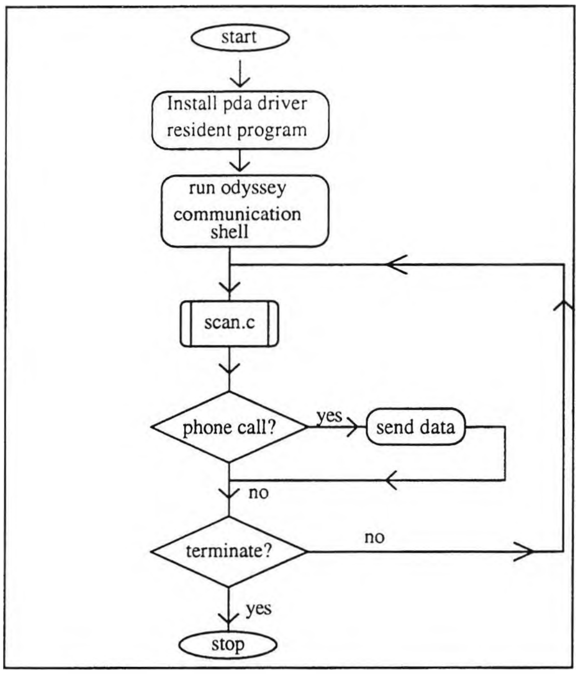

start
Install pda driver resident program
run odyssey communication shell
scan.c
phone call?
yes
send data
no
terminate?
no
yes
stop

Fig. 3.2 A flow chart for the software controlling the remote instrument.

The SCAN program is used to gather the data containing spectra over the range 200-800 nm from the spectrometer and to save the resulting data to the hard disk. This procedure is carried out according to the flow chart shown in Fig. 3.3. In a step by step way, it can be described as follows:

1. the controlled parameters is retrieved from the "Setdata" file. These parameters are: temperature setting point to the diode array, exposure time, duration time between each scan, number of files to gather and record data in 60 seconds, etc.
2. the spectrophotometer is initialised with parameters from step1,
3. the background intensity is gathered during the time no light is emitted from the deuterium lamp,

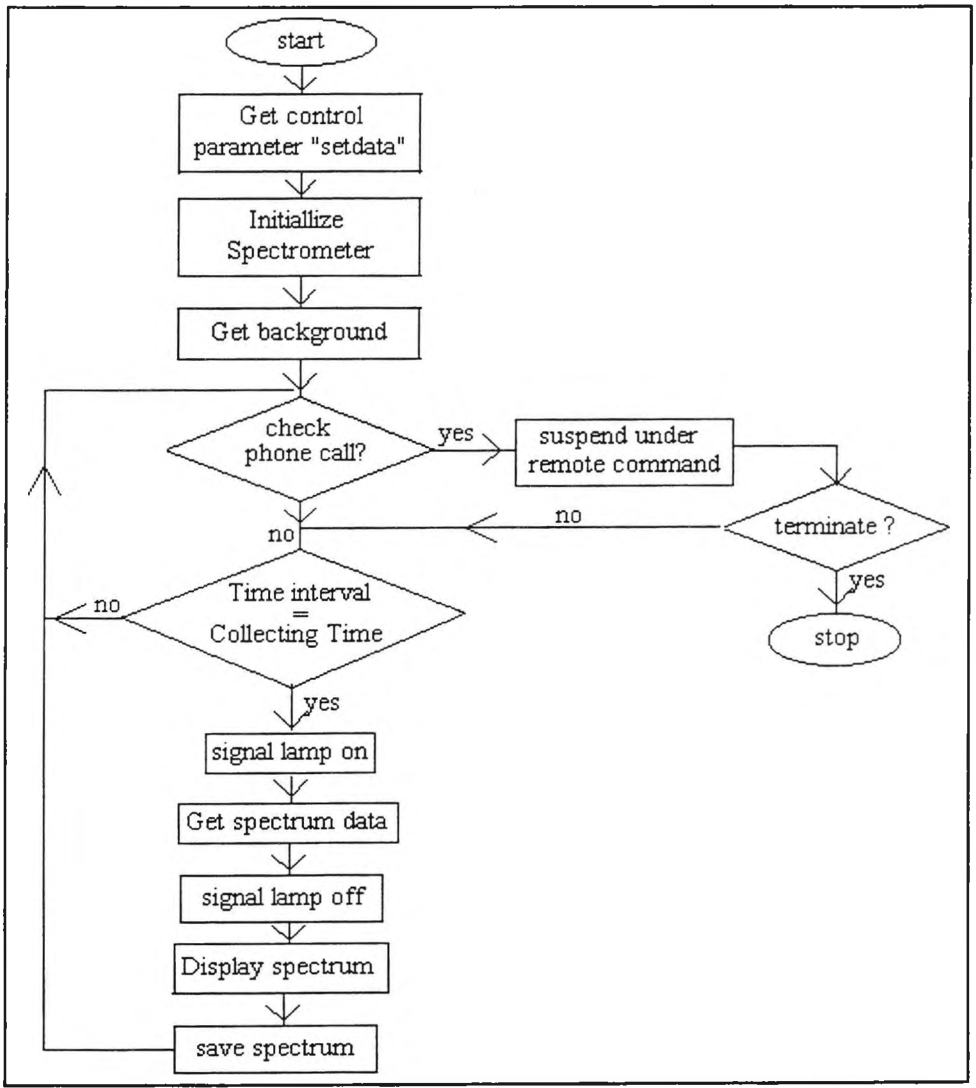


Fig. 3.3 Flow chart for the controlling software (SCAN.C).

4. the controller sends a "heating the filament" signal to lamp power supply, and then deuterium filament is heated up,
5. the program runs the loop for checking a phone call signal and a clock. The phone call interrupt will suspend this loop until the interrupt procedure has returned. When the interval time equals the cycle duration, as defined in step 1, the program carries out the following commands:
   - send the "lamp on" signal, which produces a 'strike' voltage between filament and anode of deuterium lamp from the power supply,

- read the intensity signal of the 200-800 nm spectrum from the spectrometer,
- send the “lamp off” signal, which stops the deuterium filament and return it to the heat up filament stage.
- plot the intensity spectrum vs. wavelength, and
- save the resulting data to hard disk.

This loop runs continually until the terminated signal occurs by pressing a key on the keyboard or by the host machine.

The user-defined parameters are saved in an editable "SETDATA" file, which file layout is shown in Appendix A. These parameters involved are the temperature of the diode array, the exposure time, and time duration for gathering the spectral data and recording to hard disk. The spectral intensity data have been recorded as an ASCII text file with filename "s_data"(for a 5 second cycle), "m_data" (for 10 minute cycle), and "h_data" (for 1 hour cycles) as set as default in ‘SETDATA’ file. Each filename has an extension filename which is generated by incrementing the number of files. These text files then require subsequent processing in order to convert them to either absorption or transmission data.

Although the system has operated satisfactorily during a period of two week testing at City University, the installation at industrial sites of the system has drawn back several problems encountered during a long term operation of the system in the real environment. These experiments at industrial sites will be described in the next section.

3.2 Experiments at the Industrial sites

In this approach, the automated system was installed at two different industrial sites. One was at a treated outflow from the main laboratory of Unilever Research and the other was at an untreated outflow from a major beer brewery. The system confronted a number of problems, that gave several valuable lessons to improve the system to suit operation in real situations.

3.2.1 Unilever experiments(1)

The equipment was established at Colworth House, on Unilever premises to monitor the treated outflow from the Research Laboratories. The instrument was installed for a period of two months. During this period only a few useful data sets were gathered because the system faced several problems, such as the impractical levels of intensity spectra, several apparently negative values of intensities, and halting of the computer. The following section will discuss the cause of the problems and how they were solved.

3.2.1.1 Problems encountered in trials of the system

Although the system was found to have operated in a satisfactory way during two weeks of testing at City University, the system faced several problems during installation at the industrial sites. These problems and the associated causes are discussed in the following.

The first problem was that the spectral intensity that appeared in an intensity plot over a period of time dropped, at one particular wavelength. This was due to the presence of optical flow cell fouling that had built up after a period of three to four days. This resulted from a quick build up of biological material with bacteria, and muddy water forming a gelatinous layer with a green or brown colour on the glass windows. The early stage of such fouling can be removed by the use of jets of biocide, and then by flushing with clean water. However, it requires manual cleaning schedules over a longer term to deal with severe build-up of fouling.

The second problem experienced was that the spectral data were collected and stored in 1024 values of intensities in the range 180 nm. to 820 nm. Some of these values appeared as negative numbers, that were caused by two sources. Firstly, an individual transmission signal was less than its background signal value where the intensity spectra are the result of subtracting of the transmission and background signal values. This occurrence was caused by the failure of a deuterium lamp, or by very high levels of absorption. These can be noticed in a spectral intensity vs. wavelength plot. When the whole intensity spectrum drops down to the background level, this means no light is emitted from the deuterium lamp at that wavelength. The dropping of intensities in the UV range may be caused by a very high concentration of chemical species, whereas the high turbidity often experienced drops in the intensities in the visible range. Secondly, because the photodiode array uses sixteen bits for data acquisition, where the last bit is used to express the sign of the number, therefore the data for which the value is greater than 2^{15} will be represented by a negative value.

The final problem experienced was due to the physical design of the PDA driver card and the A/D converter card whose installation blocked the ventilation in the computer housing. These boards overheated, resulting in the computer terminating the gathering of spectral data. The equipment also suffered from power fluctuations, as well as the frequently break-down of the electrical supply.

A number of important lessons have been learned for future designs, even though a small amount of useful data were received during this period, and these were taken into account for the next trials, as discussed in the following section.

3.2.1.2 Solving the problems encountered

Three major problems highlighted, i.e. impractical data, the fouling of windows, and unauthorized system termination were solved, as shown in the following section.

Impractical data

To solve the impractical data problem, the background and deuterium lamp stability were observed. The variation of exposure times was taken into account to find the optimum fit with a limit of sixteen bit data (0-65535). The stability of the signal relative to the background was observed by using a 90 millisecond exposure time every 15 seconds, for 5 minutes. This observation was due to the experience that some intensity values less than dark current signal were observed.

The background signal was represented by:

$$\left.\begin{array}{ll} \textit{1666.82 cps} \pm \textit{0.56\%} & \textit{in odd no.} \\ \textit{2121.92 cps} \pm \textit{0.46\%} & \textit{in even no.} \end{array}\right\} \textit{of diode array}$$

This background signal fluctuated between an odd and even number of the diode array as shown in Fig. 3.4. From this result, it was determined that the future experiments would gather only the even diode signals, which are more stable.

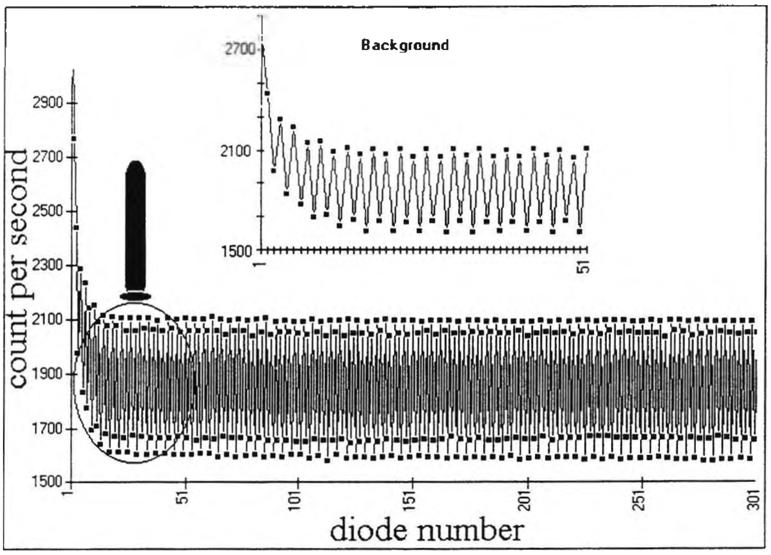


Fig. 3.4 The background stability plot obtained

The power supply to the deuterium lamp was controlled by a programmable logic control (PLC) which receives the trigger signals through the program SCAN.C. When the system starts, the SCAN.C program sends a "heating filament" signal to the PLC. In each data collection cycle, the program sends a "high voltage" signal to strike the filament, and sends a signal to return to the "heating filament" stage, after obtaining the required spectral data.

The use of discharge lamps such as the deuterium lamp with an "on-off" high voltage signal to strike the lamp may cause the emission of an unstable light signal. It will need time to reach a steady state. The measurement of a lamp stability can be carried out by varying the delay time between sending the "high voltage" signal and gathering the data. The measurement is carried out every 15 seconds for 20 times for each delay time at 1, 10 and 100 milliseconds. The signal observed would be smaller when the delay time is increased. The results are illustrated in Fig. 3.5 and 3.6. The observed signal is most stable at 100 milliseconds delay time which gives the average and standard deviation as 5160.6±0.69% whereas it is 5949.8±2.28% and 5833.0±2.60% for 1 and 10 ms delay times, respectively.

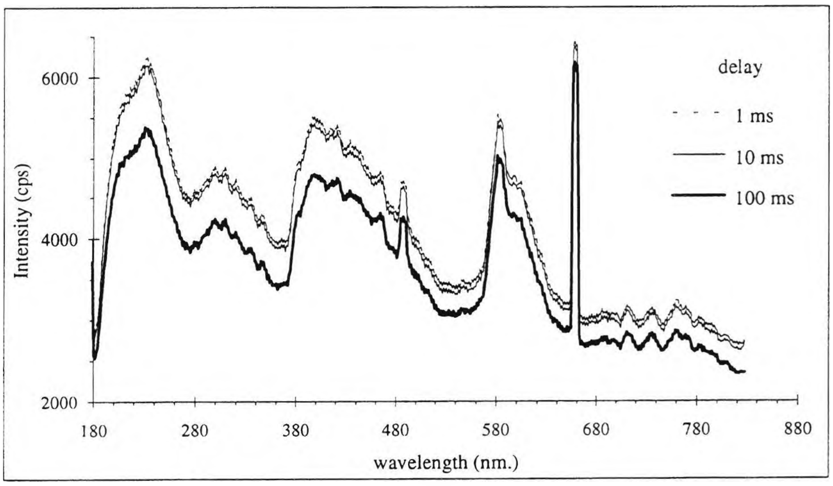


Fig. 3.5 A plot of the variety of the delay times.

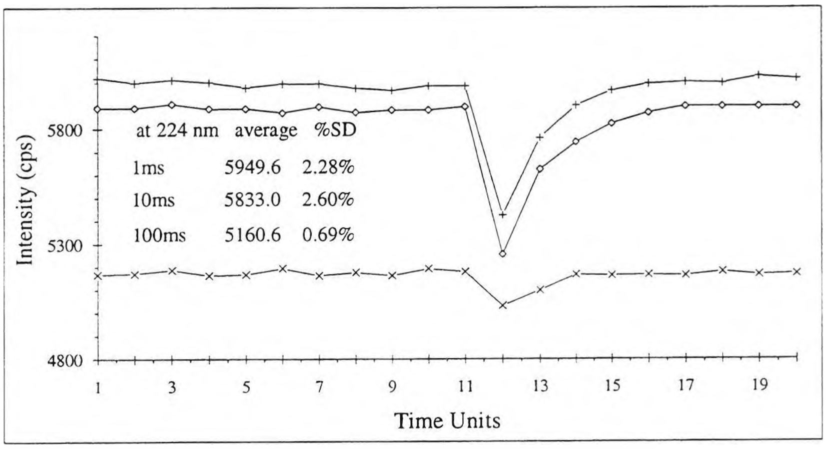


Fig. 3.6 Signal stability at 240 nm by different delay time.

The measurement was also repeated more than 300 times in every 15 seconds. By observation, it was found that the first 50 times, the lamp worked in a satisfactory way, but after that on some occasions there was no light emitted from the lamp. The solution was to replace the lamp power supply, and this was found to be a satisfactory solution.

The problem of the intensity dropping in the UV range is illustrated by Fig. 3.7. The comparison between the spectra of distilled water, Unilever waste water, detergent and oil is plotted. This shows some intensities observed in the UV range were very low. This probably was caused by using too high a concentration of a chemical present or too long a path length. The absorption is, of course, a function of the concentration and path length, as stated by Beer's law.

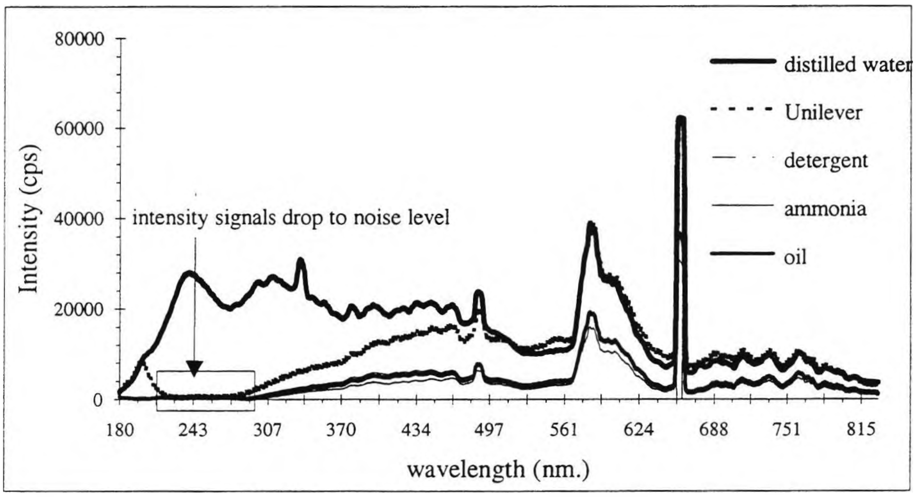


Fig. 3.7 Intensity plot due to chemical contaminants

This raises to the question "What is the threshold concentration of pollutant to obtain useful data and suited to the waste water monitoring system under development?" To attempt to answer this, a series of laboratory-prepared samples were investigated in advance of further industrial testing. The samples were prepared in the laboratory by adding various amounts of ammonia, sodium hypochlorite, sodium nitrate, and phenol in two different "backgrounds" of water. These two background waters were distilled water and "Unilever water," i.e. that from which the pollutants were removed by passing through resin. The concentration of chemicals was varied in three ranges defined approximately as corresponding to "low", "legal warning" and "legal limits".

The intensity spectra obtained are illustrated in Fig. 3.8, in which it can be seen that the intensity values of ammonia and chlorine in the spectra are still very low. It is difficult to separate them from the background due to the noise level present in the instrument. This need to reduce the path length of the flow cell means that the number of absorption particles is reduced. Otherwise the sample has to be diluted in order to increase the transmitted intensity level, apart from the presence of the background noise level.

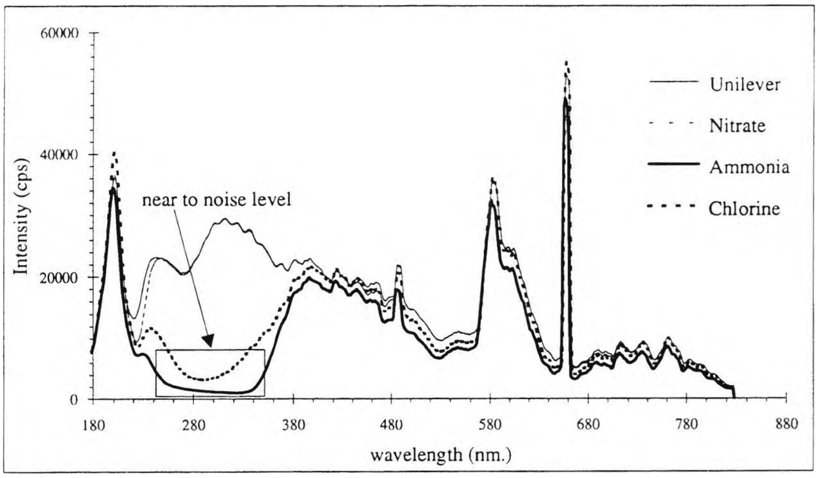


Fig. 3.8 Intensity spectra due to chemicals in "Unilever water" that was passed through resin.

Fouling of the windows

In the first trial, the flow cell was made using quartz windows at either end, which showed up the problem of fouling of the window. The quartz windows were very fragile and could easy be damaged by manual cleaning. To try to overcome this difficulty, a jet nozzle can be fabricated from a piece of 50 mm diameter plastic pipe to give a 7 x 1 cm rectangular shape, and is connected to a water supply to produce a "water jet" illustrated in Fig. 3.9. This method can also then be used to carry out the high absorption studies by reduce the optical path length from 10 cm to 1 cm. The spectral data obtained from the water jet are shown in Fig. 3.10. The intensity values of the spectra in UV range after chemical additives are higher than those of 10 cm flow cell shown in Fig. 3.7.

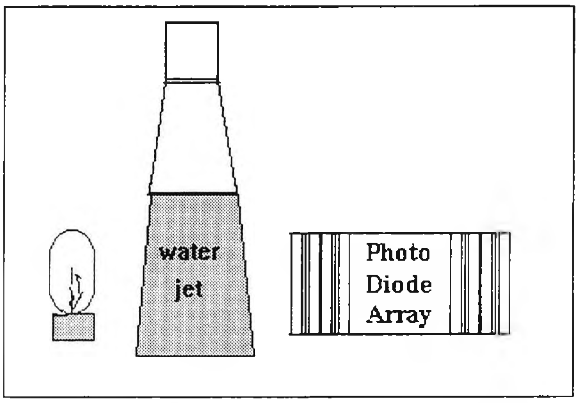


Fig. 3.9 A jet nozzle set up.

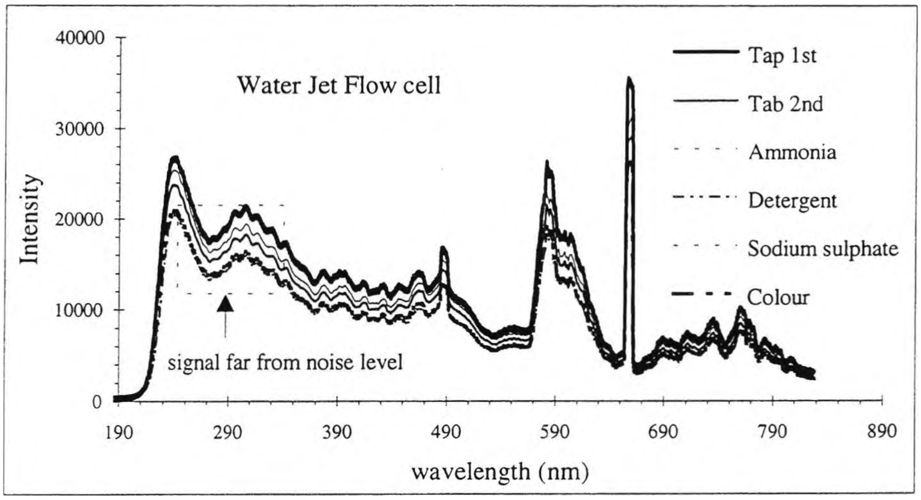


Fig. 3.10 intensity spectra through water jet

The optical flow cell is also changed, as shown in Fig. 3.11. A new flow cell was designed and consists of a 6 x 6 x 6 cm^3 cubic cell with the walls made of 1 cm thick perspex. The input flow enters centrally from the top where it is spilt and directed toward surface of the two windows, which are mounted vertically. The flow exists by passing upward through a hole in the top of the cell. The shape of the cell was changed so that cleaning of windows could be carried out without disconnection the cell from the sample supply. The design is such as to allow cleaning of the windows and to reduce the path length without the need for dismantling the entire cell, as was the case for the first design.

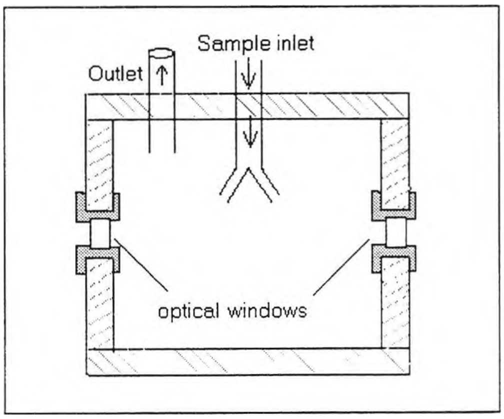


Fig. 3.11 optical flow cell.

Unauthorized System Halt

The sources of this problem are both in the software and the hardware. In the previous trial, sometimes invalid data occurred that caused the program to be left hanging with “error” questions such as "overflow divided by zero," or "invalid logarithm of negative value," or "write fault error writing device COM1: Abort, Retry, Ignore, Fail?", etc. The controlling program was modified to check for the presence of invalid data, communications problems and any possible events that might happen to interrupt the system.

The halting caused by the overheating of the two spectrometer control boards was solved by simply fitting a fan directly above these boards in the computer. An opto-isolated relay is also fitted to the reset jumper on the computer mother board that allows the re-booting of the computer, by getting an interrupt signal from the other computer.

In the original plan, the data would be downloaded on a daily basis across the modem from City University or Cranfield University. However, due to the unreliability of the computer, it was decided to connect a serial cable from the remote computer to another computer in the Unilever control office.

This PC-486 computer runs the DYNACOMM communication package. DYNACOMM receives spectral data from remote machine via a RS-232 link and checks the waiting time. It will send a reset signal to a remote computer if it has waited more than 5 minutes without any data coming through.

The new system, in which the problems discussed from the first trial were solved, was reinstalled at two different industrial sites. A number of data are collected and analyzed as discussed in the next sections.

3.2.2 Second set of experiments at Unilever

The new system was reinstalled at Unilever. The system successfully operated and was able to store the required data for a two weeks period. Figs. 3.12 and 3.13 illustrate the variation of the intensity of the transmitted light for the effluent examined. During the period of the recording of the spectra, no pollution events were reported, therefore a 50 ppm solution of ammonia is measured to represent an artificial pollution event. However the ammonia spectrum appears between the normal effluent range as shown in Fig. 3.12 and also appears in the control chart range, as shown in Fig. 3.13.

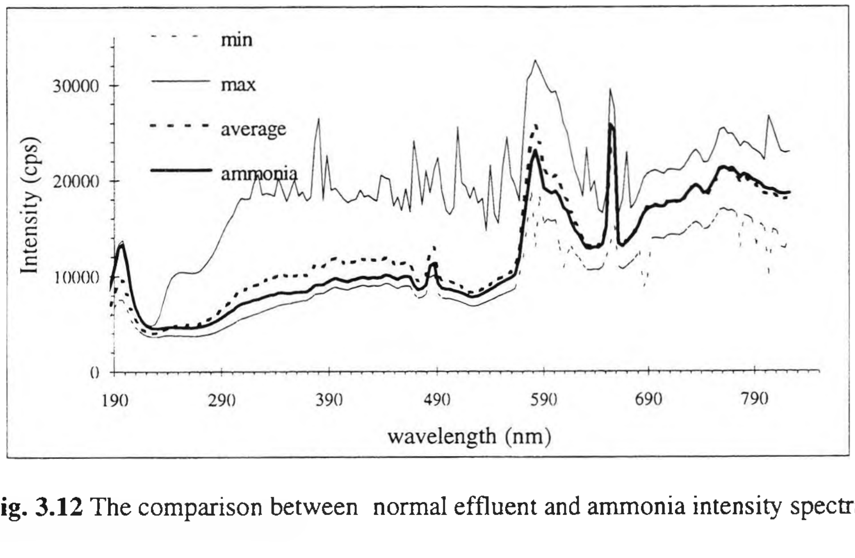


Fig. 3.12 The comparison between normal effluent and ammonia intensity spectra.

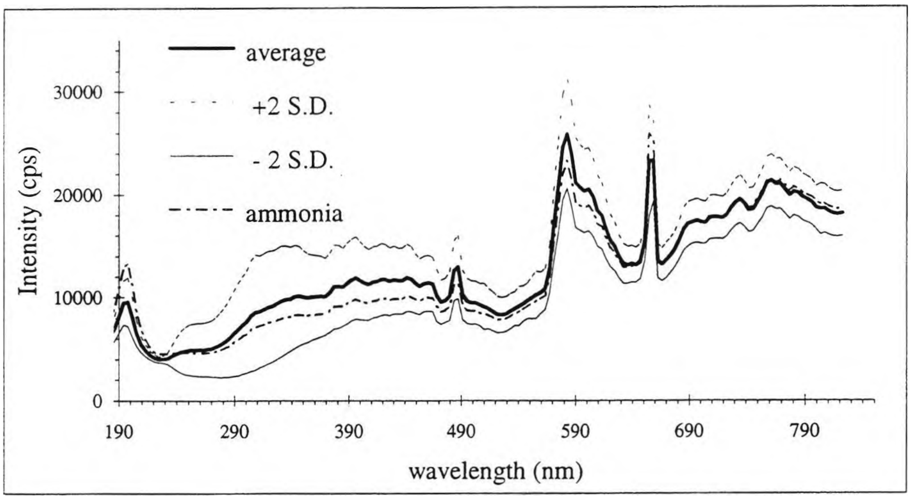


Fig. 3.13 The comparison of intensity spectra between ammonia and the control chart range.

The Fig. 3.14 shows the variation of the intensity of the effluent signal over a 3 day period. It is a plot of the intensity for three wavelengths, these being 190 nm, 210 nm, and 500 nm which correspond to the absorption peaks of ammonia, nitrate, and the measurement of turbidity respectively.

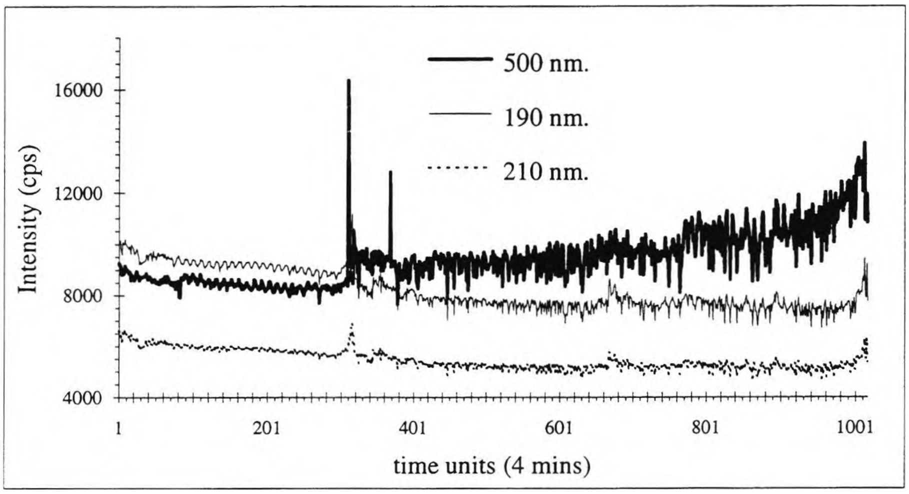


Fig. 3.14 Plot of the variation of intensity for 190 nm., 210 nm, and 500 nm. over a 3 days period.

The backpropagation neural network is then used to determine these spectral data in two states as "polluted" or "unpolluted."[50] However, it was unable to identify the pollution state in the test spectra. This inability is due to the fact that the spectra denoting the polluted samples lie between the features of the normal effluent spectra.

The equipment was then relocated to monitor the untreated outflow at a local brewery in order to test it further in the industrial environment. An experiment was set up to determine whether the pollution breakthrough monitor could function in such a different industrial environment.

3.2.3 Brewery experiment

In this experiment, the equipment was installed to monitor the untreated outflow at a local brewery. After consultation with the chief chemist on the site, the discharge of hypochlorite, which biased from cleaning agents was identified as a specific pollutant for monitoring in terms of accidental discharges as may be expected. The spectral data obtained from such an outflow were recorded over three week period. Fig. 3.15 shows a plot of the variation of intensity over 24 hours at 290 nm, which corresponds to the hypochlorite ion peak.

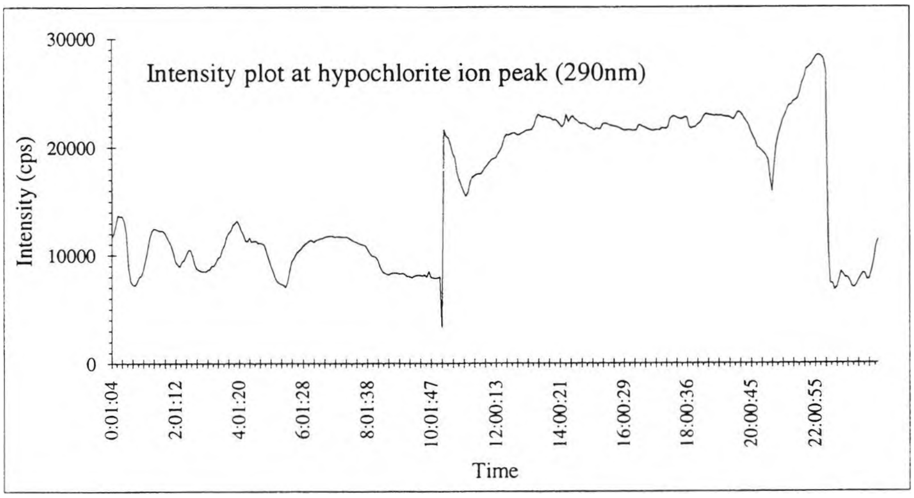


Fig. 3.15 Plot of the variation of intensity at 290 nm over a 24 hour period.

In the test approximately 1000 spectral data were recorded. The spectral data were manually sorted and classified into those representing normal effluent and those having a stronger absorption in the UV region of the spectrum, indicating the presence of absorbing species which represented the polluted effluent.

In this work, even though the general shape of the polluted spectra[50] did correspond to the changes in the UV absorption due to the presence of hypochlorite and other causes for this change are possible. This spectra for "normal" and "polluted" water are shown in Fig. 3.16. A 50 ppm hypochlorite ions solution was prepared to be a test sample for the investigation. The result indicated that the trained network can successfully classify the network output into "polluted" and "normal," but unfortunately the results could not be confirmed because the sample solutions were not collected for determining in parallel to the recording of the data.

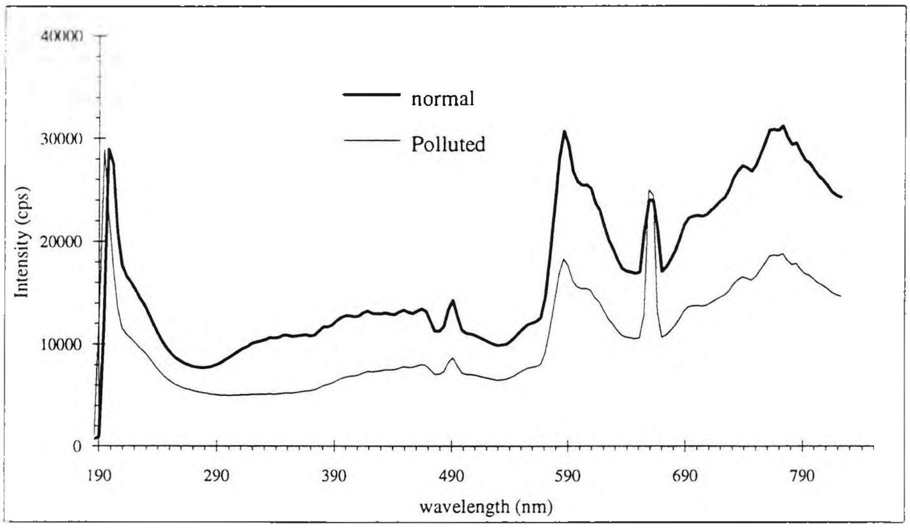


Fig. 3.16 Plot illustrating the shape of the outflow, over the range 180-800 nm of normal and polluted spectra.

3.3 Summary

A remote controlled UV-Vis spectrometer system has been designed, constructed and employed for real-time monitoring of industrial outflows. This system operated successfully over extended periods in laboratory conditions. In industrial trials, however, the system confronted several problems by which gave valuable information which the performance of the system could be improved. The first problem concerned the occurrence of intermittent faults in the monitoring system caused by interrupted power supplies, the instability in the optical source and the fouling of the optical flow cell by organic growth. These problems were all overcome by a range of solutions including the redesign of software, the optical source electronics and the use of a new optical arrangement that overcame the fouling problem. These are outsite the remit of this thesis.

The second situation experienced is more problematical because it concerns the significance of the data produced by the monitoring system. It was found to be impractical to classify the spectral data from the monitor by hand to obtain a data set for statistical and/or artificial intelligent analysis. More than 200 MB of data was obtained but there is no obvious way to classify these data either by hand or automatically. It was considered that in all but the most simplest of cases would it be difficult to estimate the error of the system fully without prior knowledge of the chemical compositions expected in the samples. This is chiefly because collecting and evaluating the necessary number of solution samples would entail an enormous cost. The alternative of simulating the expected chemical pollutants in the laboratory at City University provided a way of investigating the problem of data collection and analysis in a more controlled manner.

With the chemical preparation and the composition of the samples known, more information can be discerned from the absorption spectra and from changes in spectral detail than would be in an industrial-based trial. With well defined data sets from laboratory based simulation, mathematics and artificial intelligence methods can be evaluated with confidence in the source data. Thus, two feasibility studies are carried out and reported in the subsequent chapters which evaluate the use of Principal Component Analysis (PCA) and Artificial Neural Networks (ANN). These stages result in the data pre-processing technique for applying Neural Networks successfully.

To complete the on-line waste water monitoring system, the “front end” sensing system as described in this chapter was complemented by employing Graphical User Interface (GUI) software. This GUI system was specially developed for integrating the front end with data analysis and communication software which will be the subject of the next chapter.

CHAPTER 4
GUI-Software for On-line Monitoring

"Make things as simple as possible -- but no simpler."

Albert Einstein, Human-Computer Interaction[1]

4.0 Introduction

Computer graphics today encompass more than just the quantitative charts and graphs generated by a high level computer language like FORTRAN, Pascal, COBOL or C. While it would be possible to write complex graphics application programs using for example Line, Circle, Point to, or Paint commands, this would be very wasteful. The reason for this is that there is currently a proliferation of graphics displays, computers and operating systems in use such as Windows based programs, so it is now imperative that application programmers should be able to design graphics-based applications without any specific computer hardware system in mind. The graphical user interface, embedded in operating systems like MS-Windows, OS-2, and Xwindows, solves this problem by providing the programmer with the functionality required to implement a graphical interface tasks in a manner easily understood by the user. The emergence of such Graphical Interfaces has revolutionized the methods of man-machine interaction used in modern computers. In fact, this is the area of computing which now has the greatest impact and effect on the largest number of users as exemplified by the following quotation:

> *Technology needs to be usable as well as functional:*
>
> *"Except for special things like computer games, people don't use computers because they want to use computers because they want to write papers; they want to communicate with people; they want to design bridges and so on. Whatever they're doing, the computer is an enabling device that can help them to do it."*
>
> *"you always have to have one eye open to the question: what can the technology do? ... And one eye open to the question: what are people doing and how would this fit in? What would they do with it?"*
>
> *Terry Winograd,* Human-Computer Interaction[1]

Dramatic advances in technology have revolutionised the way that people now interact with computers. Higher performance and faster machines, in combination with the Wide Area Network (WAN) make it possible not only to transmit data but also to process and interact with these data. In practice, a wide range of experience in end-users can be accommodated using GUIs.

Some important concepts of GUIs as used in this work are introduced to those who are interested in developing a Windows based application are given in Appendix B. For more details, there are several standard textbooks on Windows programming.[51-53]

In this study, GUI software is developed for on-line waste water monitoring system which can be used by a wide range of target users, as mentioned by Winograd that:

> *"If you build something you need to consider not just 'I'm building something because I need to build it,' but 'what effect is it going to have on the way people work and the way people live?' So when you are looking at the human side,' it's not just one person, it's looking at the whole social structure of what's going on and how technology can both make that better and help solve problems."*
>
> *Terry Winograd,* Human-Computer Interaction[1]

In this study Borland C++ is used to write an application integrating the graphical tools from Windows with a high level language C++ code. In this environment the author can create a proprietary acquisition and instrument control application with C++ which can communicate and share information with other windows applications (multi-tasking) by using a Dynamic Data Exchange (DDE) protocol. In this work, this results in a software system which combines two commercial packages, DynaComm -- a communication package, and Neural Desk -- a neural network analysis package, with an 'in house' developed program called 'Waste Water Monitoring' (WWM). This development of WWM software uses the successful applications of neural network to multivariable sensing system discussed in Chapter 6. This results in a real-time analysis which is a fully automatic sensor control with a user friendly interface

This chapter discusses the development of GUI software in the area of the present application. Overviews of Waste Water Monitoring Systems are given in section 4.1. Subsequently, the structure and procedures of the Waste Water Monitoring (WWM) software is described in section 4.2 with its user manual in section 4.3. Next, communications of the WWM program with two companion packages, DynaComm and Neural Desk, are illustrated in sections 4.5 and 4.6, and finally a summary is given in section 4.7.

4.1 Waste Water Monitoring System

The waste monitoring system described here evolved out of software developed initially to control, transmit and analyse data from an on-line system for monitoring at industrial sites. This system was employed in low resolution UV-Vis spectrometry on-line monitoring experiments measuring treated industrial waste water from Unilever research laboratories and untreated waste which came from a modern brewery. The experimental setup used in these experiments is illustrated in the diagram shown in Fig. 4.1.

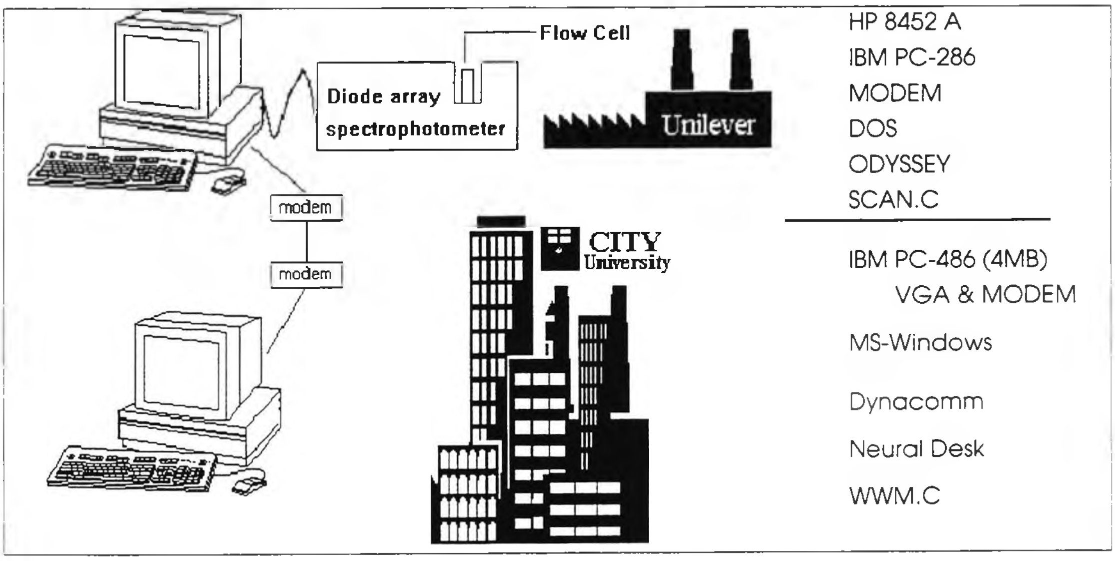


Fig. 4.1 The experimental setup scheme

This remote monitoring system consisted of a Jobin-Yvon spectraview 1D[48] spectrometer, and a Deuterium lamp source, under the control of an IBM-PC. The MS-DOS based software system for this computer, with the program named SCAN.C, was described in detail in Chapter 3. Briefly, this program controls the spectrometer and lamp and together with a communication program ODYSSEY[49] communicates remotely with a host computer in the laboratory at City University where data analysis is done.

The host machine for this task was an IBM PC-486 with 4 MB RAM, modem, and a VGA display. The software in the host machine was developed to provide remote control, maintenance monitoring, data acquisition, and data analysis of spectral data from the remote system by statistical methods and artificial intelligence. Data were acquired from the remote system periodically by a software process consisting of four nested cycles termed the standard cycle, measuring cycle, storage cycle and reset cycle, as discussed below.

Reset cycle: Shortly after the start-up of the system the WWM program begins a reset cycle, which is repeated at a preset time once a day. Here, following the flow through the measuring cell at the remote site, it is flushed with pure water and thus cleaned. A reference spectrum is then obtained and compared with the reference spectrum recorded during the last reset. This enables early recognition of a baseline drift caused by the contamination of the flow cell. If the spectral baseline exceed the pre-defined values, a ‘clean-cell’ signal occurs to initiate the flushing with pure water or biocide and then obtain the new reference spectrum.

These values of the measurement are stored as a reference to be compared with subsequent measurements as seen in Eq. 4.1.

$$A_{i\lambda} = \log\left(\frac{I_{blank\lambda} - background}{I_{i\lambda} - background}\right) \qquad ....(4.1)$$

where $A_{i\lambda}$ = the absorbance of the *i*th sample at wavelength λ,

$I_{i\lambda}$ = the intensity of the *i*th sample at wavelength λ,

$I_{blank,\lambda}$ = the intensity of the distilled water at wavelength λ, and

background = the intensity signal obtained during the case where no light is emitted.

Standard cycle: Here the average and the standard deviation (S.D.) in the previous 24 hours of sample spectra acquisition are stored and used in conjunction with a control chart for detecting instrumental malfunctions. The control chart[54] is defined by Eq 4.2 and shown as Fig. 4.2.

$$Control\ Charts = Average \pm 2 \times SD \qquad ....(4.2)$$

If values in a measured spectrum exceed the Control Chart, an instrument malfunction is signalled.

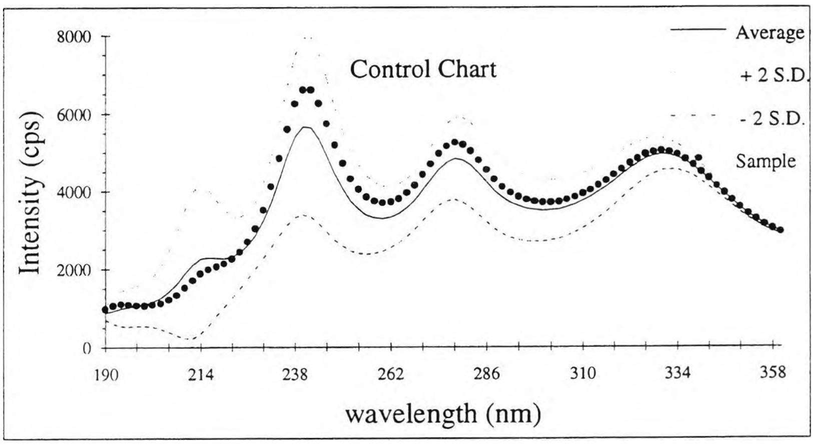


Fig. 4.2 The Control Chart.

Measurement and storage cycle: Waste water measurements are carried out in cycles at user-defined periods, with a minimum of 5 seconds. An intensity spectrum at each measurement cycle is obtained at the remote site. The spectrum is then converted to an absorption spectrum and sent via a RS-232 or link via a modem to the host machine.

The waste water monitoring program (WWM.C) in the host machine is implemented in a MS Windows 3.1 environment which provides a graphical interactive for the user. The program controls two commercial packages by using DDE. The two packages are DynaComm[55] which provides the on-line communication with the remote computer and Neural Desk[56] which provides real-time neural network analysis of the spectra. The details of WWM procedures and its companion packages will be described in the next sections.

4.2 GUI-Software for waste water monitoring system

The test engineer or data acquisition system developer is caught in the middle of the changing conditions in the PC and instrument technology industries. Instrumentation systems require a software solution that can grow and expand as system parameters change in the future. However, developers also must ‘get the system up and running’ quickly to ensure feasibility, quality of performance, and hardware compatibility today. The users demand intuitive, easy-to-use systems at low-cost, and readily available platforms that still meet performance specifications. To navigate through the changing and improving conditions of the PC and instrumentation industries, developers must select a software framework in which applications can be developed that can grow and adapt for the future. It is believed that such a software framework can be achieved with a combination of the C programming language and the Windows operating system with its advantages for implementing GUIs. The C language was developed to create very fast and efficient applications. C is also flexible enough to meet the requirements of almost any system. Therefore, Borland C^{++} version 3.1[57] and Microsoft Windows[58] are selected in order to meet the middle point between flexibility and ease-of-use.

Using C and Windows, the neural network data analysis methods developed in Chapter 6 are combined with data acquisition and control software resulting in an integrated software system called WWM. This software system also utilizes two commercial packages, DynaComm -- to handle the communications module, and Neural Desk -- to handle the neural network analysis. In operation, this software performs DDE with DynaComm, which obtains data from the remote system, to obtain absorption spectra, and then pre-processes these spectra into the format required by the neural network. Next, DDE is used to supply and obtain data from the Neural Desk package which determines the water constituents. Finally, alarm monitoring is performed by comparing the results from the Neural Desk with user-defined levels. A step-by-step procedure of this software will be illustrated in the following section.

The system starts up with three steps: Firstly, the WINDOWS program is loaded. Now with the system operating under a windows environment, the DynaComm communication Package is launched under the control of a script language program called ‘WWM.DCT’. Finally, a controlled program, WWM.EXE, is launched. On starting up this program, the menu SETUP allows the user to define all the operating parameters, after which the system performs on-line and connects with the remote system.

This description of the system operation is shown in the flow chart illustrated in Fig. 4.3. When a Dynacomm receives the spectral data from the remote computer, the script program, WWM.DCT, commands DynaComm to save it in a RAM memory-based drive file, RAMDRIVE, as “specdata.txt” in a form ready for WWM.EXE to use. With multi-tasking in the windows environment, the WWM.C program is then launched, it starts its operations by geting the required operational parameters from the ”SETUP” file and then sets the TIMER variable according to the user-defined from the SET-UP menu. The program then enters the program loop depicted in Fig. 4.3, analysing and displaying the spectral data before returning to obtain the next update of ‘specdata.txt’ obtained by DynaComm.

The nested loop procedure of WWM.C consists of the following 10 stages which are described as following:

(1) **obtaining the intensity spectrum:** This module reads data from the “specdata.txt” file and converts it into absorption data. The turbidity is also estimated from the absorption at a wavelength 500 nm.

(2) **plotting absorbance and turbidity:** The absorbance plot is displayed against wavelength whereas the turbidity plot is animated with 50 time units from current time to 49 units beyond.

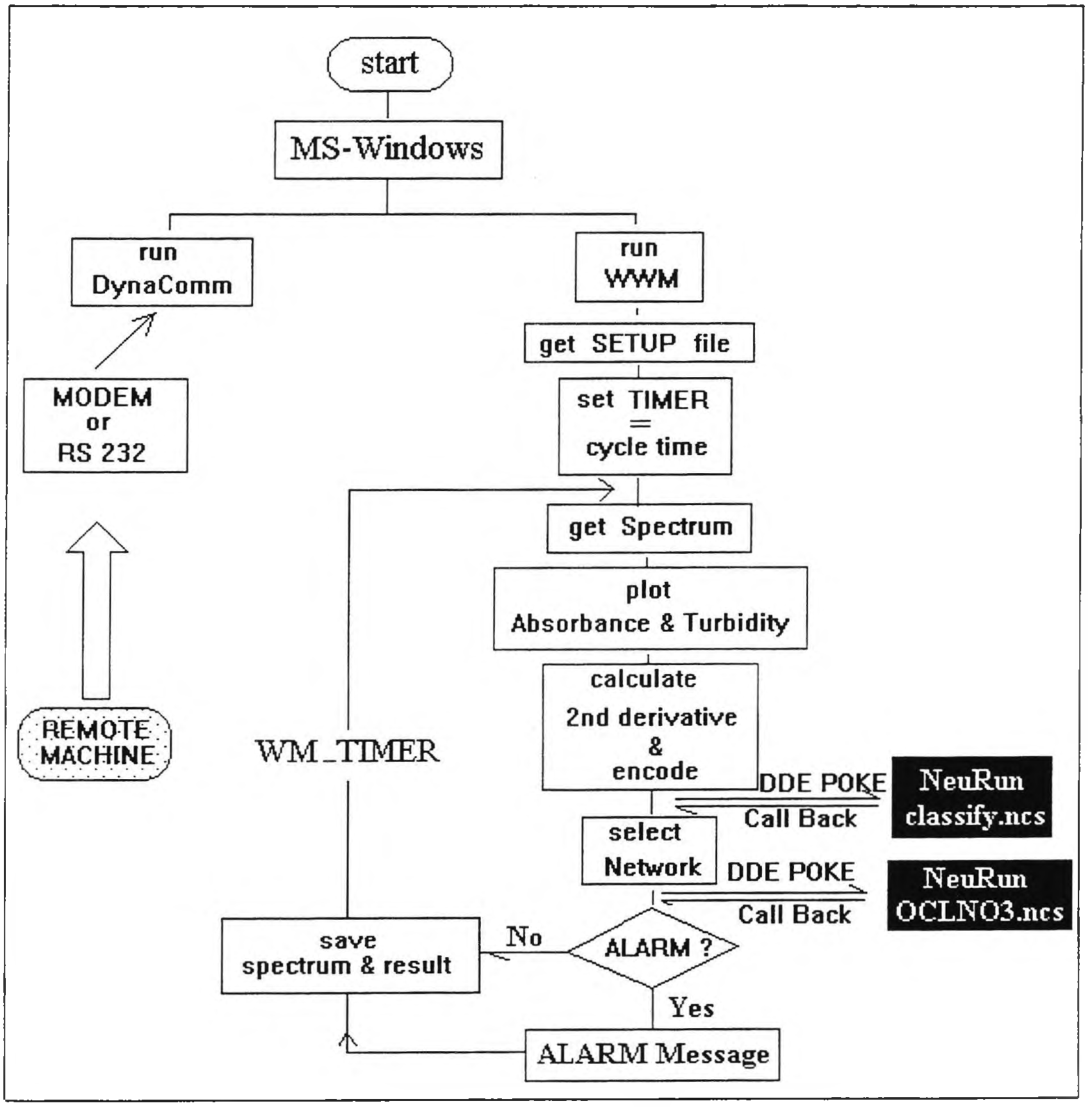


Fig. 4.3 The WWM system flow chart.

(3) calculating the second derivative and encode: The second derivative spectrum is calculated and divided in 10 intervals. Each interval is encoded by two digits to represent shape information in that interval, for example, 00 for unchanged, 01 for decrease, 10 for increase and 11 for convex. This step is carried on according to the study discussed in Chapter 6.

(4) Dynamic Data Exchange of the encoded data to "NeuRun - classify.ncs": "NeuRun - classify.ncs" is the Run-Time application of the Neural Desk package using the classification network defined in "classify.ncs" (see Chapter 6). The DDE conversation is established with the NeuRun package running the neural networks for classification procedure. After an acknowledge message from NeuRun is received, the encoded data from step 3 will be sent to NeuRun.

(5) **getting a Call Back from "NeuRun":** The WWM.C waits for NeuRun to interrogate the data and send back its classification results by implementing a function DDE Call Back. The WWM.EXE then terminates the DDE conversation after it gets the Call Back message or a DDE_TIMEOUT message which is three seconds after the data has been sent to NeuRun.

(6) **selecting the second network and preparing the corresponding inputs:** Three outputs from "NeuRun: classify.ncs" determine the presence, the uncertainty, or the absence of each of the three species -- OCl^-, NO_3^-, and NH_4^+. These three categories are separated by the values in the range 0.70 - 1.00 for presence (√), 0.30 - 0.69 for uncertainty (?), and 0.00 - 0.29 for absence (X). Then these outputs are used to determine which second-step network is used to estimate the concentration of the species involved. This step is carried out according to Eq. 4.3 as:

$$\text{Net.No.} = \sum_{i=1}^{3} O_i \times 2^{i-1} \qquad ....(4.3)$$

where O_i equals 0, if the category result is absence otherwise O_i equals 1. The Net.No. is the number in range 0-7 as categorized shown the description in Table 4.1.

Table 4.1 A list of networks for concentration estimation.

Net No.	species	NeuRun file name
0	no contaminated	no concentration estimate
1	OCl^-	OCL.ncs
2	NO_3^-	NO3.ncs
3	OCl^- & NO_3^-	OCLNO3.ncs
4	NH_4^+	NH3.ncs
5	OCl^- & NH_4^+	OCLNH3.ncs
6	NO_3^- & NH_4^+	NO3NH3.ncs
7	OCl^- & NO_3^- & NH_4^+	OCLNO3NH.ncs

The input data for the selected network is prepared according to the feasibility study discussed in Chapter 6. In the case where the Net. No. equals 0, the concentration of each species is given as 0.00 mg/l and then the program steps over to step 9.

(7) **DDE of inputs data to "NeuRun: - ********.NCS":** The DDE conversation has been established to the NeuRun package with the appropriate network file as shown in Table 4.1. After an acknowledge message from NeuRun is received, the input data will be sent to NeuRun.

(8) **getting a Call Back from "NeuRun - ********.ncs":** The WWM.C waits for NeuRun to interrogate the concentration results and send them back using the DDE Call Back function. The WWM.C then terminates the DDE conversation after it gets the Call Back message or a DDE_TIMEOUT message, which is three seconds after the data has been sent out is received.

(9) alarm monitoring: The concentrations of each species and the turbidity are indicated on gauges and also compared with a "alarm" limit. If one or more exceeds the alarm limits then the time, alarm messages and the concentration of exceeded species will be displayed in the alarm window. The instrument malfunction alarm will be displayed when an absorption spectrum exceeds the control chart. This alarm may indicate that the absorption which is bigger than the control chart could cause by lamp failure whereas the absorption less than the control chart may cause by pump failure.

(10) saving spectral data, results, alarm messages, date and time: The spectral data, concentrations and turbidity, and alarm messages (if they occur) are recorded with the time appended in "SPECDATA.***," "OUTPUT.***," and "ALARM.***" in the RAMDRIVE. The extensions of these filenames are given a running number which is incremented until it reaches a predefined number in the "SETUP" file. This file is stored and appended continuously for an hour in the RAMDRIVE after which it is closed and then transferred to hard disk. At the same time the new file with the incremented extension will be opened in RAMDRIVE to await the incoming data.

More technical details of the software are given in Appendix C as a reference manual for WWM.C. This reference manual together with the Appendix A, which describes the layout of the files in the software system, will be of benefit in adapting the software for future work. The layout of the files provides information for further analysis or implementation of the kept data. A WWM User Manual has been created to show its capabilities and it is demonstrated in the next section.

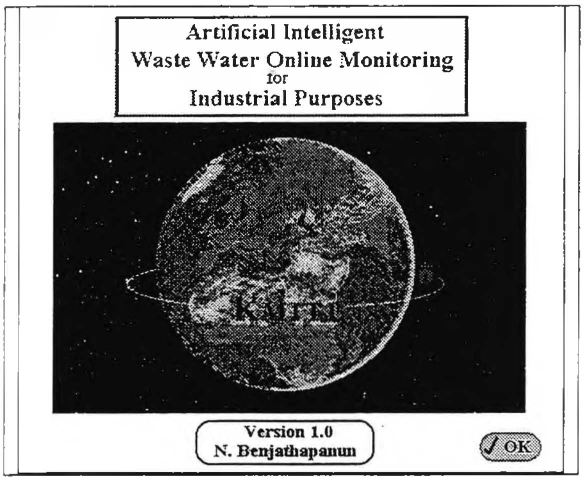
4.3 WWM User Manual

Many people, including maintenance technicians, find it easier to work with pictures instead of words, thus the WWM package has been developed to be easy for the users who are familiar with MS-Windows where the entry, dialog, menu, windows, buttons, and help functions are similar across most applications. In the WWM program, the screen display is designed to be user-friendly with a real-time animated display clearly presenting data. Also in the program, all data are stored in CF_TEXT format which can be directly transferred to any other windows application such as Excel, Word, Matlab, etc. Thus it is simple to extend the analysis and implementation. For example, the data files can be opened by the spread sheet program, 'Excel' then plotted on a graph, or transferred to 'Matlab' for manipulation by mathematical tools such as a Fourier Transform, or transferred to 'MS-Word' for presentation in a report.

As stated, the operation of the WWM program is like any other menu-driven windows application. It has the following menus; Main, Setup, Statistics, On-line, Help, and Exit menus, the purposes of which are illustrated in the following sections.

4.3.1 Title Screen

When the WWM program starts up, the title screen in this case is shown as an animation of King Mongkut's Institute of Technology and City University logos as shown in Fig. 4.4. This screen is edited from paintbrush as bitmap file. This bitmap file is converted to a Dynamic Link Library, Logo.dll, file thus freeing up a huge graphic memory space when an OK button or Enter Key is pressed (see reference manual) and the picture is removed. The program will then start with the main menu screen as shown in Fig. 4.5, the functioning of which is discussed in the next section.



Artificial Intelligent
Waste Water Online Monitoring
for
Industrial Purposes

Version 1.0
N. Benjathapanum

Fig. 4.4 The Title Screen.

4.3.2 Main Menu

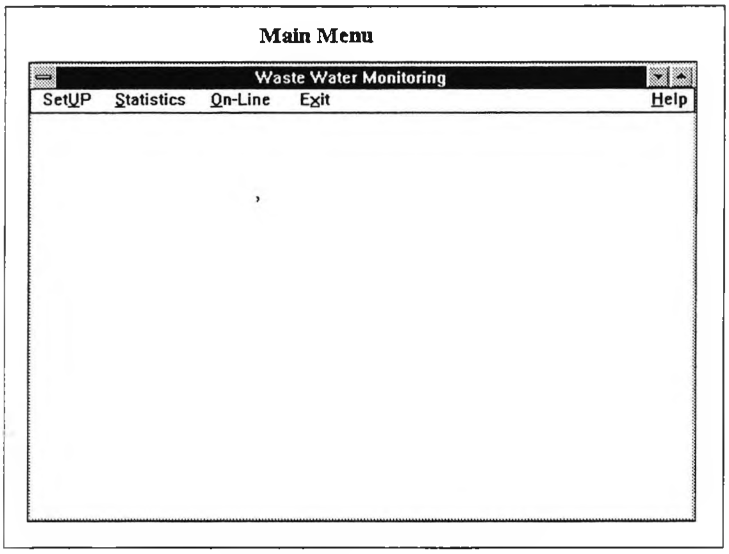


Fig. 4.5 Main menu screen

The WWM program provides four popup menus:

(1) **SetUP:** for setting the operation parameter and alarm limits.

(2) **Statistics:** for displaying chemical concentration, turbidity, and the occurrence of alarms in term of average, standard deviation, maximum and number of events over limit in a user-defined interval.

(3) **On-line:** for on-line operation with the remote site via dynamic data exchange with the communication program 'DynaComm', and for real-time analysis by interprocess communication with the 'NeuRun' program.

(4) **Help:** for information about "How the program functions."

More details of each menu will be described in the following sections.

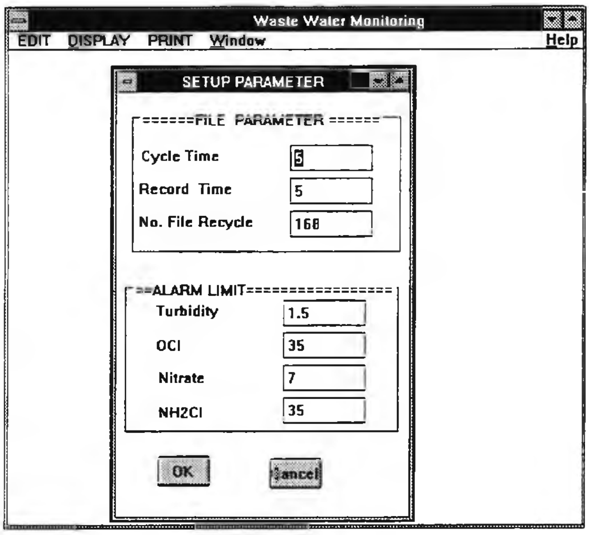
4.3.3. SetUP Menu

This menu is used for setting three operation parameters and six alarm limits which are saved in the "SETUP" file. These parameters are:

(a) cycle time, which is a duration time of at least 5 seconds to gather a new spectral data set and analyze it. This parameter is set for 5 seconds as default,

(b) record time, which is a duration time to record the data and results. The default value is also set to 5 seconds.

(c) number of file to numbering. The spectral data and results of the data analysis are appended to files in a RAM-DRIVE for an hour. After this, these files are transferred to hard disk as 'SPECDATA.***', 'OUTPUT.***', and 'ALARM.***'. These filename extensions, '***' are given by a number which is incremented until it reaches the number that is user-defined and then starts at 000 again. The default value is 168 which will be overwritten after one week. The alarm limits are the turbidity level and concentrations of the chemical species. The setup screen is shown in Fig. 4.6.



======FILE PARAMETER ======

Cycle Time: 5
Record Time: 5
No. File Recycle: 168

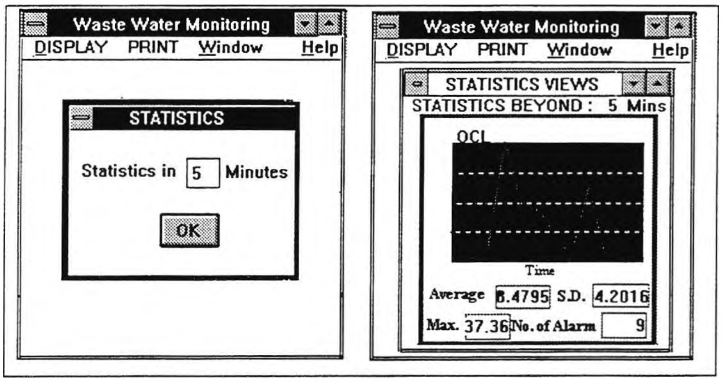
==ALARM LIMIT================

Turbidity: 1.5
OCl: 35
Nitrate: 7
NH2Cl: 35

Fig. 4.6 Set Up screen

4.3.4. Statistics Menu

The Statistics menu provides the necessary statistical information such as average, minimum, maximum, and standard deviation of the data that have been record during the period of time that required in the range 1-60 minutes. The DISPLAY menu allows the user to choose to display the statistics of turbidity, hypochlorite ion, nitrate ion, or monochloramine as shown in Fig. 4.7. The PRINT menu will print this screen out and the WINDOW menu allows the user to select other windows which can be by tile or cascade together.



Statistics in 5 Minutes

STATISTICS BEYOND : 5 Mins

Average 8.4795 S.D. 4.2016
Max. 37.36 No. of Alarm 9

Fig. 4.7 Statistics Screen

4.3.5. On-line Menu

The On-line screen consists of three windows; including a display of the real time implementation, Neural-Nets analysis, and Alarm windows, as illustrated in Fig. 4.8. In the first window, the current absorption spectrum and reference spectrum, the average of previous spectra in a 24 hour period, are displayed. The turbidity is plotted against time units with the display showing the current value and the previous 49 values. In a second window, the second-derivative and encoding processes are displayed. The resulting binary codes are dynamically linked to NeuRun: CLASSIFY.NCS for classification, whose outputs are displayed by the circled ticks and crosses which depict which species are found in the water. The concentrations for these species are displayed after the Call Back signal is informed by the second-step network. In the last window, these concentrations and the turbidity are illustrated with the user-friendly gauges which are scaled in low, warning and alarm ranges. When an alarm occurs, the current time and an alarm message is displayed in the last alarm window and a beep sound is given. Subsequently, the spectrum is stored in "SPECDATA.***" the results in "OUTPUT.***," and the alarm message in "ALARM.***." These files are stored in CF_TEXT format which is able to directly transfer to any windows applications for further analysis. The layout of these files is reported in Appendix A.

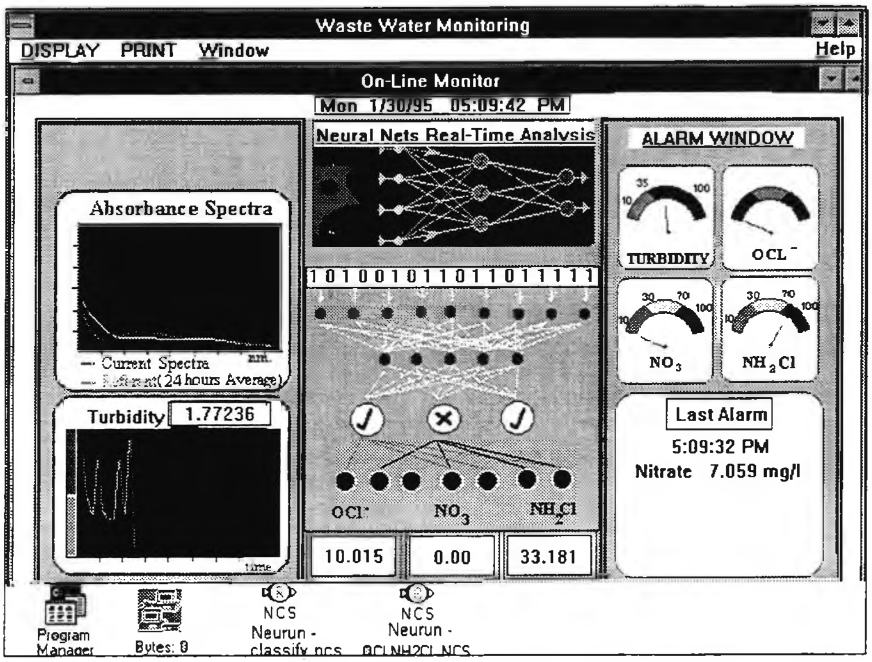
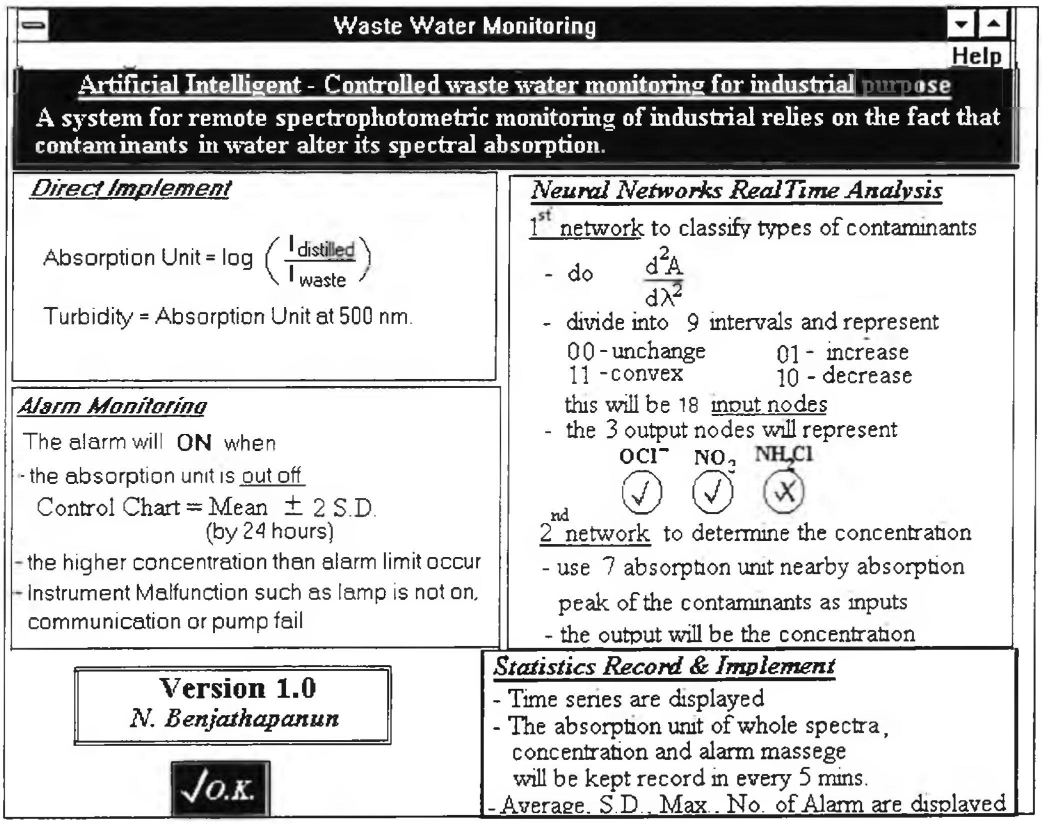


Fig. 4.8 On-line screen

4.3.6. Help About Screen

The Help menu will display the “Help about” screen that informs the users about “How this package works”. It explains how absorbance and turbidity are implemented, when an alarm will be triggered, how the data are preprocessed for both networks, and what statistics are used. It is shown in Fig. 4.9.



Artificial Intelligent - Controlled waste water monitoring for industrial purpose

A system for remote spectrophotometric monitoring of industrial relies on the fact that contaminants in water alter its spectral absorption.

Direct Implement

Absorption Unit = $\log\left(\frac{I_{distilled}}{I_{waste}}\right)$

Turbidity = Absorption Unit at 500 nm.

Alarm Monitoring

The alarm will **ON** when

- the absorption unit is out off

  Control Chart = Mean ± 2 S.D.
  (by 24 hours)
- the higher concentration than alarm limit occur
- Instrument Malfunction such as lamp is not on, communication or pump fail

Neural Networks RealTime Analysis

1st network to classify types of contaminants

- do $\frac{d^2A}{d\lambda^2}$
- divide into 9 intervals and represent

  00 - unchange 01 - increase
  11 - convex 10 - decrease

  this will be 18 input nodes
- the 3 output nodes will represent

  OCl^- NO_3 NH_2Cl
  ✓ ✓ ✗

2nd network to determine the concentration

- use 7 absorption unit nearby absorption peak of the contaminants as inputs
- the output will be the concentration

Version 1.0
N. Benjathapanun

Statistics Record & Implement

- Time series are displayed
- The absorption unit of whole spectra, concentration and alarm massege will be kept record in every 5 mins.
- Average, S.D., Max., No. of Alarm are displayed

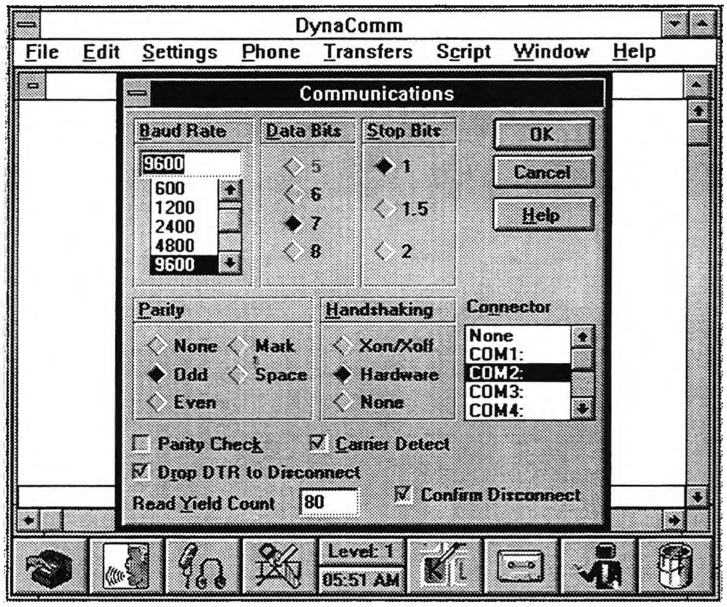
Fig. 4.9 Help about screen

It has been already shown that the WWM software does not implement the waste water monitoring alone but does so together with two companion packages called DynaComm and Neural Desk. These two applications are described in the next section.

4.4 Communication link with Dynacomm

DynaComm version 3.1 is a communication package which runs under MS-Windows. It contains a set of terminal emulations, supports binary transfer protocols to allow considerable flexibility in transmitting and receiving data, and includes its own language. This Script language allows DynaComm to be customized to suit the user needs and to access other applications via Dynamic Data Exchange.

When the WWM system is started, firstly the communication parameters are initialized both at the remote machine and the local machine. These parameters are Baud Rate, Data bits, stop bits, Parity, Handshaking, Connector, and Parity Check as indicated in Fig. 4.10. Next, the DynaComm script language program, "WWM.DCT", is launched. This WWM.DCT script runs periodically to checks the connection, retrieve the data via communication line and transfer these data to WWM.EXE. If the connection has been established, the "text.txt" file will be opened to record the data. Once the "EOF" string has been received, this file will be closed and then copied to the "SPECDATA.TXT" file. This SPECDATA.TXT file is ready and waiting for WWM.EXE to retrieve it for analysis. The script program re-opens the "text-txt" file and waits for new data to come through the connection. This process is being performed continuously until the communication is disconnected.



Baud Rate: 9600
600
1200
2400
4800
9600

Data Bits: ◇ 5 ◇ 6 ◆ 7 ◇ 8

Stop Bits: ◆ 1 ◇ 1.5 ◇ 2

Parity: ◇ None ◇ Mark ◆ Odd ◇ Space ◇ Even

Handshaking: ◇ Xon/Xoff ◆ Hardware ◇ None

Connector: None, COM1:, COM2: (selected), COM3:, COM4:

□ Parity Check ☑ Carrier Detect

☑ Drop DTR to Disconnect

Read Yield Count 80 ☑ Confirm Disconnect

Fig. 4.10 DynaComm communication setup screen

4.5 Neural Networks by NeuDesk

The neural networks package used here is NeuralDesk version 2.1 from Neural Computer Sciences. It operates within the MS-WINDOWS environment. This package consists of two components; NeuDesk and NeuRun. NeuDesk is the interface program that enables the user to design, train and run a neural net automatically from a set of data input in spreadsheet format. NeuRun is the run time processor, controlled by external programs which supervises the processing and interfacing of the neural net.

There are two stages of using Neural Networks, which are the training stage and query stage. The training stage will be described in detail in the study presented in Chapter 6. In this stage, the EXCEL spreadsheet package was used to transform the spectral data for each of particular preprocessing technique. Both EXCEL and NeuDesk run under MS-WINDOWS that supports inter-application communications in Clipboard, so that the data in EXCEL3 Spreadsheet can be cut or copied and then pasted into the NeuDesk spreadsheet.

4.5.1 NeuDesk

There are six steps in using NeuDesk for the feasibility study which will be discussed in Chapter 6. Fig. 4. 11 shows the setup training data screen. These six steps involved are:

1. Set Up Training Data: the data used in this step be pasted from EXCEL,
2. Set Up Validation Data,
3. Set Up Network Topology,
4. Train Network,
5. Set Up Query Data: this query inputs spreadsheet also be pasted from EXCEL,
6. Query Network: the results from query outputs spreadsheet will be copied and then pasted in EXCEL spreadsheet for analyzing.

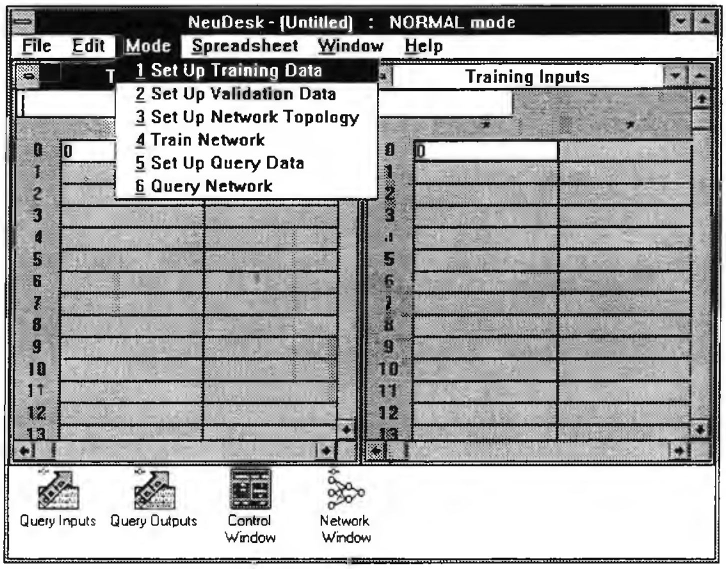


Fig. 4.11 NeuDesk Screen

The spreadsheet data; training inputs, training outputs, query inputs, and query outputs are saved in "*.dsk" while the network topology and weights are saved in "*.ncs" which can be used during RunTime or NeuRun. The neural network feasibility study results are held in eight network files; CLASSIFY.NCS, OCL.NCS, NO3.NCS, OCLNO3.NCS, NH2CL.NCS, OCLNH2CL.NCS, NO3NH2CL.NCS, and OCLNONHC.NCS. These eight files then are used by NeuRun during runtime to analyse data via DDE with WWM.EXE.

4.5.2 NeuRun

NeuRun is a runtime processor which is designed to be controlled by external programs using Microsoft's communication protocol, Dynamic Data Exchange. NeuRun will not run from the keyboard but the processor's main functions must be controlled by other applications such as WWM.EXE, EXCEL macro language, DynaComm Script language, etc.

NeuRun has three processing modes; Learn, Relate, and Idle, as shown in Fig. 4.12. These three processing modes state the current status of the neural network as the following:

<u>Learn</u>: in which the neural net is being trained using the current set of data;

<u>Relate</u>: in which the neural net uses the current set of data to produce a response;

<u>Idle</u>: in which is the default state where the neural net is standing by.

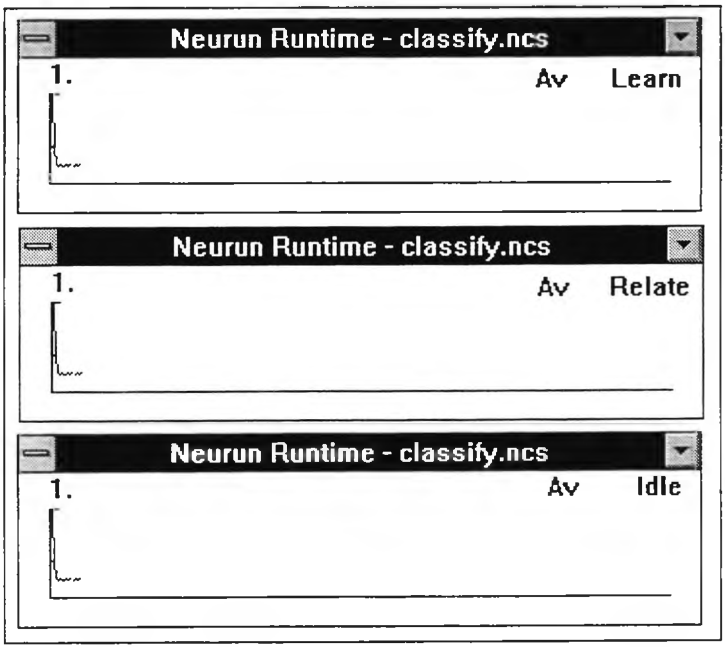

Neurun Runtime - classify.ncs
1.
Av
Learn
Neurun Runtime - classify.ncs
1.
Av
Relate
Neurun Runtime - classify.ncs
1.
Av
Idle

Fig. 4.12 NeuRun screen

Before the run time processor, NeuRun, can be invoked, it has to communicate to receive commands. The DDE conversation is established and controlled by WWM.EXE with NeuRun acting as a server to receive the data and transmit data in response to a request from WWM. In this work, the WWM first invokes a NeuRun-classify.ncs and sends the commands to interrogate the result. After WWM receives the result, it terminates NeuRun-classify.ncs. The results from classify.ncs are then used to determine which network will be used to estimate the concentration. The WWM.EXE establishes conversation with the appropriate network and then terminates the NeuRun when it obtains the results. This results in only a proprietary network which is invoked at one time, and then terminated when it is finished and the results are sent. The reason for this is to free the memory space for another module. This cycle is continuous until the on-line process is finished. The Reference Manual in Appendix D will describe more of the technical aspects of WWM.

4.6 Summary

This GUI-software is very easy to use for those are familiar with MS-Windows. The entry, dialog, menu, windows, buttons, or help are similar to most other applications under the windows environment. The screen display is designed to be user-friendly and lively with real time animation; moreover the use of the DDE protocol results in an automatic real time data analysis. Under the windows environment, DDE also gives rise to the possibility of transfer of data between several windows applications for further analysis or implementations. Although, using a windows application is very easy, in contrast developing windows programs is difficult. In fact, testing for each module is done incrementally with all modules integrated together. Testing and debugging in small steps like this is very necessary when using the DDE module to exchange data between commercial packages where no technical details have been provided. More often than in desirable a problem is shown as the result of a missing “call back” or sometime the execution just halts with the message “NeuRun error at xxxx : xxxx Protection Fault!”.

The “GDI object” really belongs to the Windows Kernel which in the applications under windows, shares resources. Failing to delete GDI objects that WWM creates can be a more difficult bug to discover, because the program will appear to work well. Sometimes the program will not fault after a few iterations, but will after many hundreds of iterations, with the system “hanging”. This occurs when each iteration calls a procedure and does not free local memory after use, resulting in a continuous reduction in the free memory. It is tedious to test hundreds of iterations runs that seems all right, for the system the suddenly to hang.

The wide range of possible user requirements, together with the growing richness of technological opportunities, means there is no panacea in defining user-computer interaction, no single right answer.[59] It is appropriate that we can develop the right system to fit a specific purpose. Therefore, the application must be flexible to adapt for the future or for a specific industry site or integrated front end sensor. In this application, the screens are created by a Graphic tool as a bitmap file that can be easily redesigned and also with the editable setup file which makes it is easy to modify the parameters used to control process. However, developments under C language is flexible enough to meet the requirements of almost any system. Furthermore, WWM is developed under the Windows environment which there are now over 10 million copies of Windows[51] in used, and hundreds of Windows applications[51] are currently available. Therefore this program can exchange the data with those hundreds of applications in either DDE or Cut and Paste. This results in a powerful software system which can be easily tailored to specific applications by other several expert teams. Currently, Windows is the dominant operating system on the PC. The combination of C programming language and the Windows operating system can be efficiently used to develop the WWM for the growth of instrumental and human requirements in the future.

Higher performance and faster machines, in combination with Wide Area Network (WAN) system make it possible not only to transmit data but also to process and interact with this data to immediately control the front end machine or even control the industrial process. To do so, a suitable data analysis method has to be investigated.

The implementation of Neural Networks based data analysis used here is based on an extensive programme of work in developing data analysis methods. This work was required to be suitable for on-line and real-time data analysis methods and the design of a practical algorithm for integrating with the GUI system.

The development in the data analysis is based on laboratory simulation samples on which a mathematics and artificial intelligence analysis can be undertaken with confidence in the source data. Thus two feasibility studies are carried out in the following chapters which evaluate the use of Principal Component Analysis (PCA) and Artificial Neural Network (ANN). The experiments to collect these data and the evaluation of PCA technique will be described in the next chapter.

CHAPTER 5

Feasibility Study of Principal Components Analysis for Water and Environment Monitoring

> *"Data rich but information poor is an excellent way to characterize most chemical process today."*
>
> *M.J. Piovoso.* Process Data Chemometrics[10]

5.0 Introduction

Most species dissolved in water, absorbing in the ultraviolet and visible parts of the spectrum, have broad absorption peaks with peak widths of typically 20 nm or more. Such broad absorption peaks lead to difficulties in identifying different species that may produce an overlapping absorption bands. To overcome this problem, chemical methods can be employed in what is termed '*differential absorption spectroscopy*' whereby the species involved is removed through filtration or chemical reaction and then a ratio measurement without the species of interest is carried out. To do this, the sample chemistry is modified by the addition of reagents, either added directly or possibly leached from a soluble glass or resin. This technique applies only for the measurement of one specific species while waste water samples may contain many chemical species of interest. It may be possible to determine more than one species at once by the use of mathematical techniques to determine species that contribute to the overall spectra.

Another motivation for using sophisticated mathematical techniques is that advances in automated sensing systems make possible the collection of large amount of data, but this has occurred without corresponding advances in data analysis. Every modern industrial manager believes that this data bank is a gold mine of information if only the "*important*" and relevant information could be extracted painlessly and quickly. Such an interpretation of data would improve quality, safety, reduce waste, and improve business profits.

The combination of suitable mathematical algorithms and computer programming may give a good answer for the best use of this information. In this context, the study is concerned with the answers to following questions:

- Can low-resolution UV-Vis spectrometry be used to monitor several different impurities at the same time?,
- Can the chemical state of industrial outflows be determined by classifying features in the UV-Vis spectra without adding reagents to remove interfering absorption peaks?, and
- What is the possibility of using mathematical methods and modern Artificial intelligence approaches to determine pollution events?.

The 'breakthrough monitoring' that was discussed in Chapter 3, while determining the state of "Pollution" or "Non Pollution" in a continues flow sample is limited in application. It would be highly desireable to know what species are present, and at what concentrations, because this information could be used to reduce reagent use in water treatment, reduce material consumption in chemical processes, and identify and control pollutants. To accomplish this task, ideally an understanding of the industrial process is needed, such as what chemicals are used in a particular process, what will cause a pollution event, and what is a possible accident event. It is also necessary to have a knowledge of the absorption spectra of expected species and time series data from the process under study. The achievement of these requirements result in several problems. It is difficult and expensive to get controlled industrial or waste process data, because it means tying up equipment and it takes a long time to collect useful data. An attempt to do so can lead to poorly defined experiment data. As a result, it was decided in this study to carry out a series of simulations in the laboratory to investigate the system under more carefully controlled conditions.

The aim of this study is to investigate the feasibility of using mathematical techniques in UV-Vis absorption spectroscopic monitoring schemes, and is aimed especially at industrial process and waste water monitoring. In such environments the type and range of constituents that may be present in water is likely to be better known than in open water and sewage water monitoring. More specifically in this study, the abilities of mathematical techniques to identify the presence and to quantify the level of the three chemical components considered, nitrate, ammonia and chlorine in sample solutions, are studied. Ammonia and chlorine are studied because they are representative of chemicals expected, in information obtained through consultation with chief chemists at industrial sites. Nitrate exists in most natural waters and its absorption band interferes with that of ammonia and chlorine. The level of these chemical species is of interest in most free waters and industrial waste waters. They may result from chemical processes, washing processes, or the quality of water supply.

Based on an exhaustive and positive literature survey, the author was encouraged to investigate the use of Principal Component Analysis (PCA) to analyse the data. However, it is believed that the success of such analysis applied to examples given in the literature is dependent on the precise chemical preparation and ideal conditions current in chemical laboratories, and therefore the technique may have serious limitations in the less controlled environment of waste water monitoring. Waste water is a very complex mixture, composed of the chemical substances used in factory, chemicals in water supply, and bio-organics from sewage systems. In real water systems there are certainly significant interactions between these components, for example, ammonia from a sewage system which interacts with chlorine from a cleaning agent. However, there exists a wide variety of different process industries all discharging effluents, often of unique composition. These discharge can be specifically modelled by generating and observing a large number of samples in which the amount of each component is varied. Even though PCA is a linear multivariate technique, the use of PCA is appropriate to investigate the analysis of linearly and non-linearly related samples. The justification for this application is that although the interaction between ammonia and chlorine is non-linear, it can be treated as linear by assuming that the reaction between these, to form monochloramine, goes to completion. Therefore the amount of new product can be calculated as well as the consumption of ammonia and chlorine.

In this Chapter, a literature survey and a short mathematical explanation of PCA are given in sections 5.1 and 5.2 respectively. The experimental procedures are explained in section 5.3. The application of PCA in classification is given in section 5.4 and quantitative analysis and results are given in section 5.5 and 5.6. The feasibility study of the use of PCA for water environment is concluded in section 5.7, with comments on its suitability and utility.

5.1 Principal Component Analysis Review

PCA was originally developed by Pearson[60] and was introduced into chemistry by Malinowski.[61] PCA forms the basis for solving many chemical problems. For example, the analyst may be interested in exploring the relationships between samples and may ask the questions of their spectra: Do they fall into classes or are there any outliers--i.e., a typical sample or measurement -- in the data? If so, which variables are important in distinguishing between classes or differentiating the outliers? Alternatively, the aim may be to simplify the data, to model and subsequently to predict chemical properties for unknown samples.

Chemical data contains many sources of variation. It does not only change with the chemical composition of samples, but it also depends on variations such as drift or artifacts associated with the measurement system and, of course, random noise. PCA separates these sources of variation, expressing the data as linear combinations of the independent contributions. This is very attractive because it reduces the complexity of large data sets and simplifies the overall analytical problem.

PCA attempts to identify the sources of variation by pooling correlated information contained within the data matrix into a new set of variables called principal components (PCs). A simple example of correlated information is in the use of spectra of mixtures. When the concentration of one component increases, all variables associated with that component increase together. Such correlations between variables provide a way of isolating the individual sources of variation.

With the above outlined advantages, PCA has become a conventional method used in many successful analytical chemistry[12-20] and waste water pollution studies[11]. Most analytical methods use PCA in order to reduce the n-dimensional-space of data to a few significant PCs and follow with other methods to interpret the physical or chemical meaning of the PCs. For instances, Saaksjarvi, and *et al.*[11] used PCA to investigate the origin of effluents by applying PCA to water quality monitoring data from Lake Saimaa, Finland. The water quality in this lake was affected by two industrial sources discharging their effluents into the lake and the main clean water current. PCA modeling showed that there were two outliners and a few samples which differed from the others significantly because of the presence of very high values of chemical oxygen demand, colour and conductivity in the data training set. It was shown that this PCA research tool can be used to interpret complicated data matrices to extract useful information, related to the given problem.

Frequently, following a reduction of n-dimensional space, the first 2 PCs are used to observe clustering in the data. This clustering is carried out by plotting the scores, or loading, of the first PC relative to the second PC. Clustering of the resulting points in this plot can be observed by eye or higher dimensions of data space can be determined automatically using the *K-nearest-neighbour* method. An example is classification of the near-infrared reflectance spectra of authentic currency and other paper stock,[12] Dale and Klatt used PCA to reduce the 431-space data set to 2 dimensions with the retention of 91.3% of the variance. This 2-dimension data set separated the samples into distinct classes without there being any prior knowledge of their chemical or physical properties. The results were 100% correct in classification of 10 samples of paper stock and 40 samples of authentic currency bills. Another example is the grading of apples into 4 quality classes of sugar content.[13] Each near-infrared spectrum of the apples was composed of 126 data points which was reduced to the first five Principal components which accounted for 97.6% of the cumulated variance. Following this, the projection space of the spectra was mapped out according to chemical values with the use of 45 specimens of calibration. A prediction with 43 verification specimens gave a standard error of 0.8% for grading apples in 4 quality classes.

For quantitative analysis, the significant factors from PCA are obtained and then *Target Transform Factor Analysis* (TTFA) is used to transform these PCs into a data matrix with values closest to the actual data matrix by means of the least square error method. From this, the calibration matrix from a training data is generated. TTFA is identical to the statistical technique called *Principal Components Regression* (PCR). This technique is widely used in pharmaceutical and food industries for quantitative determination of several components in a mixture. Several reports describing the application of PCA for quantitaive analysis in UV-Vis spectra include:

- the construction of a multivariate calibration for mixture solutions of $Ni(NO_3)_2$, $Co(NO_3)_2$ and $Cr(NO_3)_2$[17] which can find the concentration of each absorbance without the prior knowledge of each component spectrum.
- a 94% accuracy in determination of two components between a pharmaceutical product and benzyl alcohol.[16]
- the successful use of target factor analysis to determine simultaneously the numbers, identities and concentration of six amino acids with overlapping spectra in mixtures[18]
- the potential application of PCA which were introduced in a study in which three categories are displayed and classified, interpreted and quantified[62] with examples of application to drinking water, milk, and chocolate.
- the analysis of UV spectra of the four major nucleotides of DNA[15] which was evaluated with less than 3% error.

The very good results from above examples encouraged the author to investigate this technique in application to waste water monitoring. An understanding of the basic principles of PCA is illustrated in next section.

5.2 Mathematical Formulation of PCA

When an analytical method gives a single measurement such as pH or concentration of each sample, the data are called *univariate.* When multiple measurements, such as those making up a spectrum are generated, the data are *multivariate.* For each wavelength there is a datum value, which is referred to as a variable. When multivariate data for a number of samples are available, the data may be arranged into a matrix with one column for each sample; with each row containing the values of a given variable for all samples. This data matrix is the basis for all multivariate data analysis methods and is the starting point for a mathematical description of principal component analysis (PCA).

5.2.1 Criteria for component analysis

In general, a multivariant experimental data set can be labelled as a matrix where each data point is represented by the symbol d_{ik} , and *i,k* refers to a particular row and column respectively. Factor analysis is applicable to the analysis of such a data matrix whenever a measurement can be expressed as a linear sum of product terms as expressed in Eq.(5.1)

$$d_{ik} = \sum_{j=1}^{n} r_{ij} c_{jk} \qquad ....(5.1)$$

$$[D] = [R][C] \qquad ....(5.2)$$

where d_{ik} is data point in the matrix, r_{ij} is termed the score or Row Designee and c_{jk} is termed the coefficient or loading or Column Designee. [D], [R], [C] may be written as matrices whose elements are d_{ik} , r_{ij} and c_{jk}, respectively.

PCA is a mathematical procedure that derives a set of orthogonal vectors called Principal Components (PCs) so that each successive PC explains the maximum amount of variance possible in the data not accounted for by the previous PCs. In simple terms, this is expressed in matrix form as:

$$\begin{bmatrix} Data \\ matrix \\ \end{bmatrix}_{n_s x n_v} = \begin{bmatrix} Sample \\ Scores \\ matrix \end{bmatrix}_{n_s x n_f} \begin{bmatrix} Variable \\ Loadings \\ matrix \end{bmatrix}_{n_f x n_v} + \begin{bmatrix} Residual \\ (noise) \\ matrix \end{bmatrix}_{n_s x n_v} \qquad ....(5.3)$$

where n_s is number of samples, n_v is number of variables, and n_f is number of significant PCs. In practice, most of the variance is described in the first few PCs whereas the later PCs are related to the variance at the noise level. As a result, data reduction can be obtained by retaining only significant PCs which is described by two smaller matrices sizes n_s x n_f and n_f x n_v. This decomposition is achieved by an eigen analysis which is described in the next section. A numerical algorithm for implementing PCA is given in Appendix D and a software flow chart is illustrated in Appendix E. More details of PCA can be found in several texts.[61,63,64]

5.2.2 Mathematical Synopsis

The key steps are presented in Fig. 5.1. The problem to be solved by factor analysis can be formulated as follows: From known values of [D], it is necessary to find various sets of [R] and [C] which reproduce the data in accordance with Eq.(5.2).

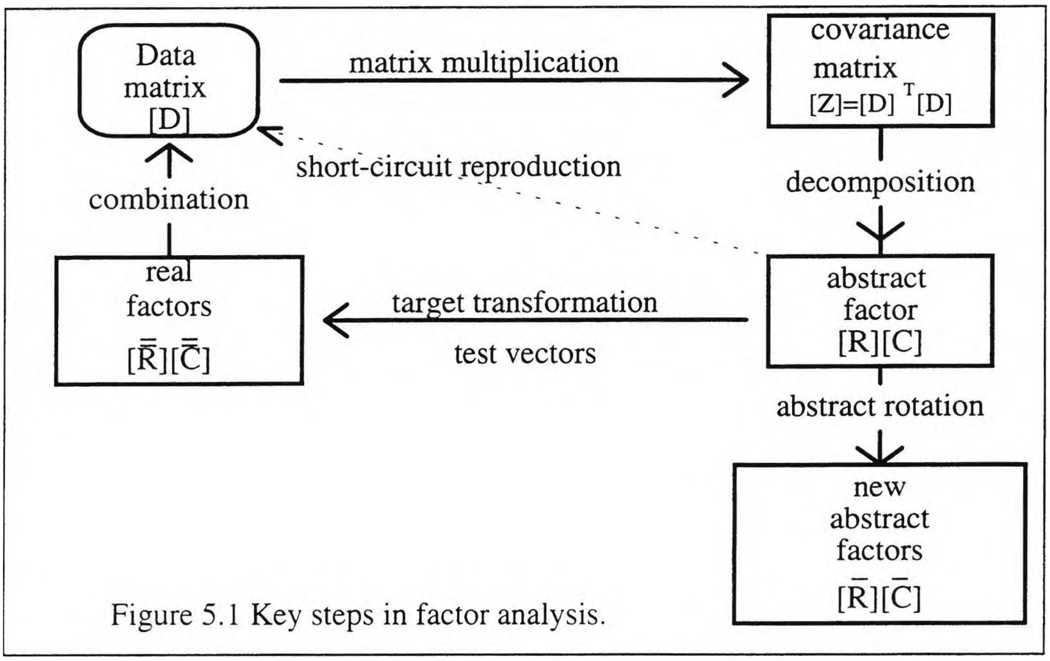


Figure 5.1 Key steps in factor analysis.

To do this, the following procedural steps are carried out:

First, the raw data are converted into a covariance [Z] or correlation matrix $[Z]_N$ which is constructed by postmultiplying the data matrix by its transpose as:

$$[Z] = [D]^T [D] \qquad ....(5.3)$$

$$[Z]_N = [D]_N^T [D]_N$$

where $[D]_N$ means the normalized data matrix.

Second, the covariance or correlation matrix is decomposed by standard mathematical (eigen analysis) techniques, via a short-circuit route, into a set of "abstract" factors [R],[C] which, when multiplied, reproduce the original data.

This step is carried out by way of the following procedures:

The covariance matrix is diagonalized by finding a matrix [Q] such that

$$[Q]^{-1}[Z][Q] = [\lambda_j \quad \delta_{jk}] = [\lambda] \qquad \text{....(5.4)}$$

Here δ_{jk} is the well-known Kronecker delta,

$$\delta_{jk} = \begin{cases} 0 & \text{if} \quad j \neq k \\ 1 & \text{if} \quad j = k \end{cases} \qquad \text{....(5.5)}$$

and λ_j is an eigenvalue of the set of equations

$$[Z]Q_j = \lambda_j Q_j \qquad \text{....(5.6)}$$

where Q_j is the j^{th} column of [Q]. These columns, called eigenvectors, constitute a mutually orthogonal set which is usually normalized to form an orthonormal set. Hence

$$[Q]^{-1} = [Q]^{T} \qquad \text{....(5.7)}$$

$$\begin{aligned}[Q]^{-1}[Z][Q] &= [Q]^{-1}[D]^{T}[D][Q] \\ &= [Q]^{T}[D]^{T}[D][Q] \\ &= [U]^{T}[U]\end{aligned}$$

where $[U] = [D][Q] \Rightarrow [D]= [U][Q]^{T}$

from (5.2) $[Q]^{T} = [C]$ ....(5.8)

and $[D][Q] = [R]$ ....(5.9)

Eq.(5.8) shows that the transpose of the matrix which diagonalizes the covarience matrix represents the column matrix. This matrix is an eigenvector therefore

$$[C]^{-1} = [C]^{T} \qquad ....(5.10)$$

The row matrix [R] is then calculated from Eq. (5.8) and Eq. (5.9), and the data matrix [D] can then be reproduced as Eq. (5.2). Thus this scheme is readily accomplished in an abstract manner and these factors are called abstract because, although they have mathematical meaning, they have no real physical or chemical meaning in their present forms.

The purpose of PCA, however, is to reproduce the data within experimental error, with a minimum of eigenvectors. Eigenvectors associated with the largest eigenvalues are most important while the least important eigenvectors (the smallest eigenvalues) may be dropped from the factor analysis (*factor compression*). There is a variety of mathematical ways to decompose the covariance matrix. Principal component analysis is by far the most widely used method.

Third, Target transformation will enable conversion of these factors into physically significant parameters by means of the least-squares error method.

PCA can also be represented from a vector viewpoint, as described in the next section.

5.2.3 Vector Interpretation

An insight into the overall operational details of factor analysis can be obtained from a vector viewpoint. With this perspective in mind, the columns of the data matrix are considered as vectors. The elements of the covariance matrix are generated by taking dot products of every pair of columns in the data matrix. On the other hand, when forming the correlation matrix, each column of data is normalized before taking the scalar products. Each element of the correlation matrix represents the cosine of the angle between the two respective data column vectors. The diagonal elements are unity since they are formed by taking dot products of the vectors onto themselves.

If n eigenvectors are needed to reproduce the data matrix, all the column vectors will lie in *n* space, requiring *n* orthogonal reference axes. It will be easier to understand by giving an example, that is to consider a normalized data matrix, consisting of four data columns (D_1, D_2, D_3, and D_4), generated from two factors. For this, the correlation matrix, $[Z]_N$' is shown below:

$$[Z]_N = \left[\begin{array}{c|cccc} & D_1 & D_2 & D_3 & D_4 \\ \hline D_1 & 1.00000 & 0.06976 & -0.58779 & 0.80902 \\ D_2 & 0.06976 & 1.00000 & 0.76604 & 0.64279 \\ D_3 & -0.58779 & 0.76604 & 1.00000 & 0.00000 \\ D_4 & 0.80902 & 0.64279 & 0.00000 & 1.00000 \end{array}\right] \qquad ....(5.11)$$

The elements of this matrix are the cosines of the angles between the data column vectors. Fig. 5.2 shows all four data vectors lie in a common plane of two dimensions, which means that only two factors are involved.

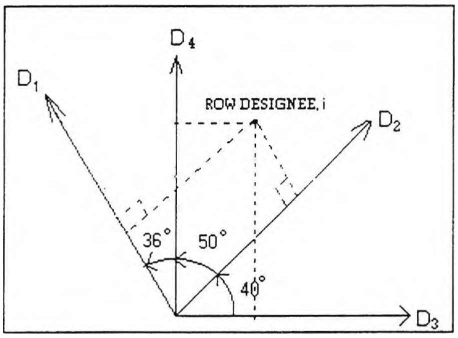


Fig. 5.2 Vector relationships of data column vectors.

Each vector axis in Fig. 5.2 corresponds to a column designee of the normalized data matrix. Each row designee of the normalized matrix is represented by a point in the two-dimensional plane. The value of a data point associated with a given row and column is obtained by first drawing a line through the row-designee point, perpendicular to the appropriate column vector, and then reading the distance along the vector from the origin to the intersection. These projections are the normalized data values, since the vector axes represent normalized data columns. The value of a point projected onto any axis is called the *factor component* or "*score*". If one of the vectors in the figure did not lie in the plane, then the space required to described the data would be three-dimensional. Three factors would then be required to account for the data and three axes would be required to locate the data points in the factor space. In many problems the factor space has more than three dimensions. It is impossible to sketch such multidimensional situations on two dimensional graph paper. However, it is possible to extract all of the necessary information by using factor analysis which can determine the exact dimensions of the factor space. The eigenvectors that emerge from factor analysis span the factor space but do not coincide with the data vectors; as they merely define the factor space in which all the experimental data points coexist.

In the above example, when the correlation matrix is subjected to the decomposition step, two mutually orthonormal eigenvectors, C_1 and C_2, and their associated eigenvalues, λ_1 and λ_2 emerge:

$$C_1 = \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \end{bmatrix} = \begin{bmatrix} 0.5084 \\ 0.5084 \\ 0.0847 \\ 0.6899 \end{bmatrix} \quad \text{and} \quad C_2 = \begin{bmatrix} c_{21} \\ c_{22} \\ c_{23} \\ c_{24} \end{bmatrix} = \begin{bmatrix} 0.4909 \\ -0.4909 \\ -0.7044 \\ 0.0877 \end{bmatrix} \qquad \text{....(5.12)}$$

$$\lambda_1 = 2.070 \quad \text{and} \quad \lambda_2 = 1.930$$

Each of the four data-column vectors can be expressed in terms of the basic eigenvectors in the following way:

$$D_k = \sum_{j=1}^{n} \sqrt{\lambda_j} c_{jk} C_j \qquad \text{....(5.13)}$$

Accordingly the four data-column vectors may be expressed in terms of the two eigenvectors:

$$D_1 = 0.7314C_1 + 0.6819C_2$$

$$D_2 = 0.7314C_1 - 0.6819C_2 \qquad \text{....(5.14)}$$

$$D_3 = 0.1218C_1 - 0.9926C_2$$

$$D_4 = 0.9926C_1 + 0.1218C_2$$

The c_{jk} coefficients that define the eigenvectors also measure the importance of each eigenvector on each data column. The $\sqrt{\lambda_k} c_{jk}$ coefficients are known as "*loading*" in classical factor analysis, whereas they are called "weightings" in regression analysis.

The geometrical relationship between the two eigenvectors and the four data column vectors can be obtained by taking dot products between the eigenvectors and the data vectors as expressed in Eq.(5.14). The vector dot product equals the cosine of the angle between the two vectors. The relationships so obtained are illustrated in Fig. 5.3. The projection of each data vector onto each eigenvector represents the loading of the eigenvector for the respective data-column vector. The projection of each row designee onto each eigenvector axis gives its *score* on the axis.

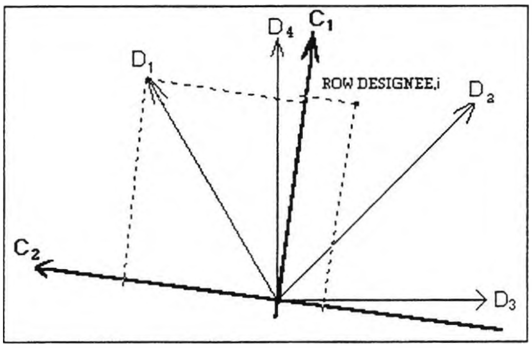


Fig. 5.3 Geometrical location of eigenvector C_1 and C_2

The following sections presents a feasibility study of the use of PCA to classify and quantify chemical components in water.

5.3 Experimental study using PCA to determine the constituents in water

In the feasibility study of PCA, a simulation UV spectroscopic assay of batch samples of nitrate ion (NO_3^-), ammonium ion (NH_4^+) and hypochlorite ion (OCl^-) mixtures is used to illustrate a typical pollution suitation in water. The study undertaken in the laboratory is an attempt to classify these three species and their concentrations. The chemical preparation and instrumentation of these samples are described in the following section.

5.3.1 Preparations of Chemical Samples

The chemical preparations of the nitrate ion (NO_3^-), ammonium ion (NH_4^+) and hypochlorite ion (OCl^-) samples are explained in detail. The original stock solutions for each species were prepared by using analar sodium nitrate, ammonia liquid and sodium hypochlorite.

Analar sodium nitrate ($NaNO_3$) was mixed with distilled water to provide a stock solution of nitrate ion (NO_3^-) as described by Eq.(5.15)

$$NaNO_3 + H_2O \Rightarrow Na^+ + NO_3^- + H_2O \qquad ....(5.15)$$

Ammonium ion (NH_4^+) stock solution was produced by adding ammonia liquid to distilled water as indicated by Eq.(5.16)

$$NH_3 + H_2O \Rightarrow NH_4^+ + OH^- \qquad ....(5.16)$$

Hypochlorite ion (OCl^-) stock solution was prepared by mixing sodium hypochlorite ($NaOCl$) with distilled water to produce hypochlorite ion as illustrated by Eq.(5.17)

$$NaOCl + H_2O \Rightarrow Na^+ + OH^- + OCl^- + H^+ \qquad ....(5.17)$$

Each stock solution of known concentration was mixed with different volumes of distilled water to produce the samples at the required concentration used in the experiment.

In this experiment, samples were prepared and divided into two sets. One set was used for training the procedures for qualitative and quantitative analysis of the chemical species and the other was used for testing the performance of the training. The comparison between training and testing set is illustrated in Appendix F.

Each set consisted of 64 samples prepared by mixing nitrate ion (NO_3^-), ammonium ion (NH_4^+) and hypochlorite ion (OCl^-) in various concentrations as shown in Table 5.1.

Table 5.1 The various concentration of chemical species in both sets.

	NO_3 (mg/l)		NH_4^+ (mg/l)		OCl^- (mg/l)	
Levels	Training	Testing	Training	Testing	Training	Testing
A. high	7.75	9.75	105.83	137.70	49.23	60.48
B. medium	3.88	4.88	32.55	45.90	24.62	30.24
C. low	1.94	1.95	11.18	15.30	6.15	5.04

5.3.2 Spectroscopic Apparatus

UV-Vis spectra of light transmitted through the samples were obtained by using a Hewlett-Packard 8452-A diode array spectrometer equipped with a 1 cm path length quartz cell. An IBM PC was used to control the spectrometer through an IEEE 488 communications interface[65].

UV-Vis spectra were obtained for all the solutions discussed. These consisted of 316 measured light intensity points at wavelengths over the range 190-820 nm with 2 nm intervals. The dark-current signal was also recorded to account for detecting noise and a spectrum from distilled water recorded to provide a reference signal. These measurements were used to calculate the absorbance spectra as indicated by Eq. (5.18)

$$A = \log_{10}\left(\frac{I_{distilled} - darkcurrent}{I_{sample} - darkcurrent}\right) \qquad ....(5.18)$$

where $I_{distilled}$ are intensity values obtained for distilled water, I_{sample} are intensity values obtained for sample, and *darkcurrent* are intensity values obtained when no light was emitted from the lamp. The comparison between the simple intensity and the logarithmic of intensity ratio, called absorbance spectra is shown in Fig. 5.4 which the absorbance spectra illustrates much better resolution than intensity spectra.

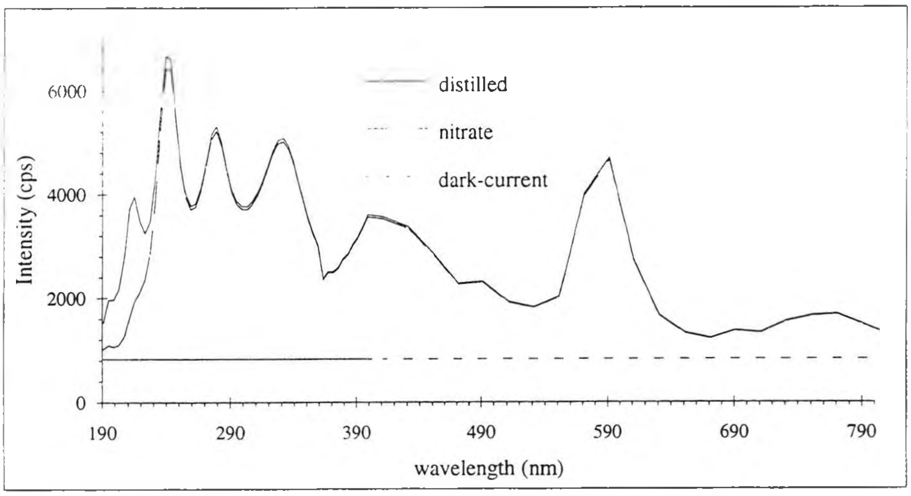


Fig. 5.4a intensity plot of nitrate ion solution.

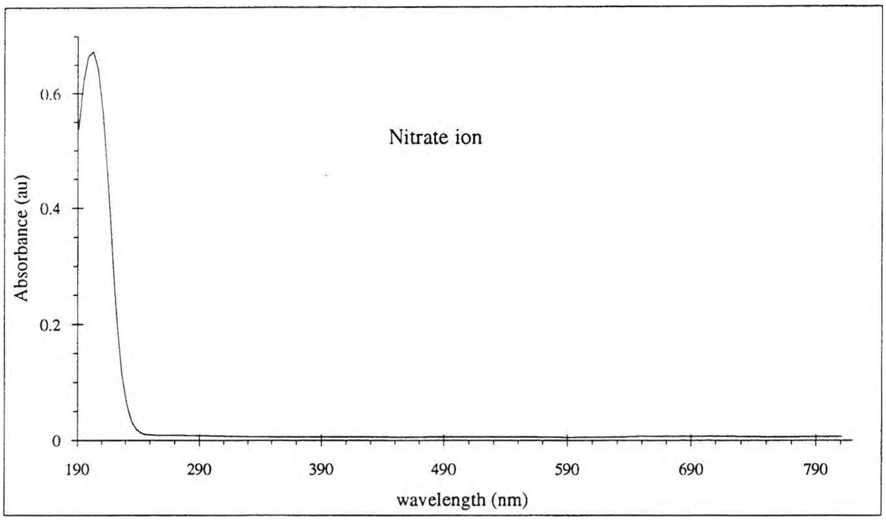


Fig. 5.4b absorbance plot of nitrate ion solution.

An experimental data matrix was constructed from these absorbance spectra. This data matrix is interpreted by PCA techniques, as discussed in the next section.

5.4 Analysis of experimental data using PCA for Classification.

PCA has two major applications in data analysis; these being classification and quantification. In this section the experimental data described above are classified using proprietary PCA software written by the author in Borland C^{++}. A flow chart of the program is shown in Appendix E.

Most common methods for classification include a visual technique which uses a plot of the sample scores for any two components. This so called "*score plot*" is a view obtained when looking at the data in the plane defined by the two principal components. Such score plots often show up trends such as groupings and outlying samples within the data set. In the experimental data obtained, the first four principal components account for 78.5%, 13.7%, 4.5%, and 2.9% of the variance in the data. The score plots for these PCs are shown in Fig. 5.5 and 5.6 .

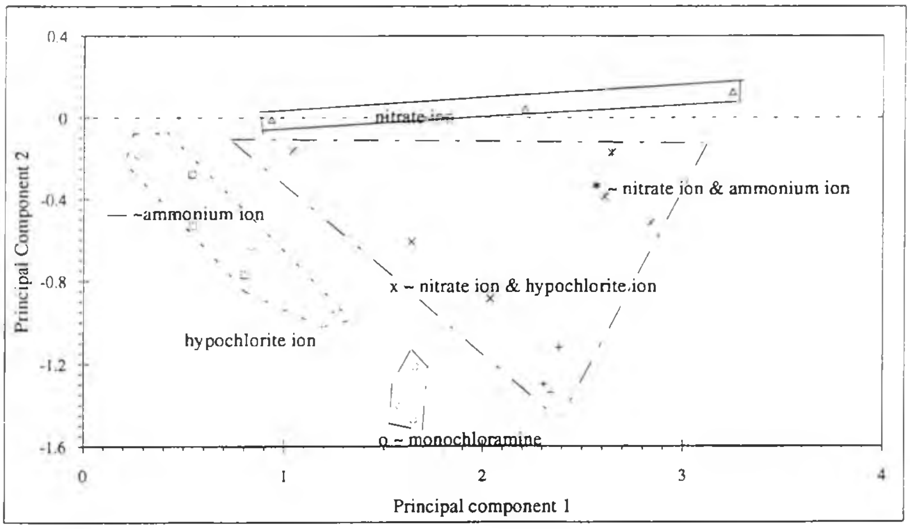


Fig. 5.5 PC 1 vs. PC 2 score plot

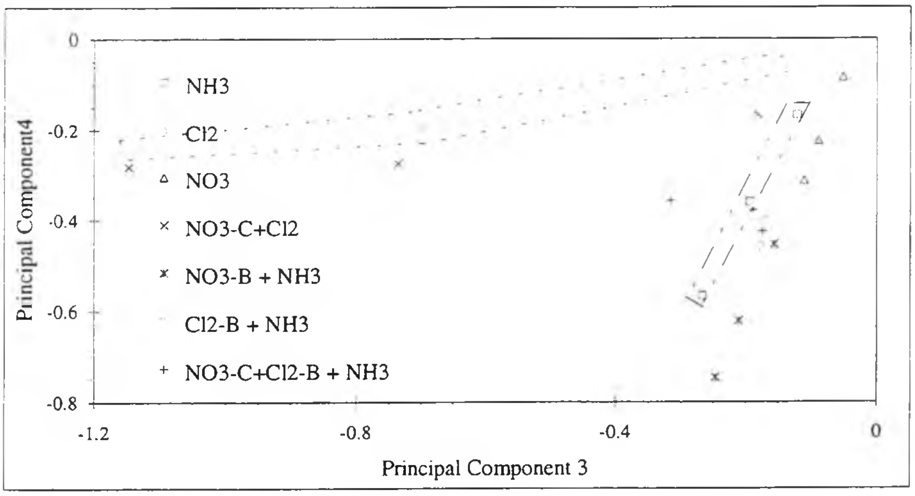


Fig. 5.6 PC3 vs. PC4 scores plot

The score plot for the first two PCs is shown in Fig. 5.5. This plot shows that the data for NO_3^- are separated well from the other data, while the data NH_4^+ and OCl^- are clustered together. That is caused by the overlapping of the absorption peaks of NH_4^+ and OCl^- in range 190-230 nm, as shown in Fig. 5.7.

The score data of the mixtures; $NO_3^-+OCl^-$, $NO_3^-+NH_4^+$, and $NO_3^-+OCl^-+NH_4^+$; are located in a wedge between the clusters for NO_3^- and the cluster for OCl^- and NH_4^+. The score data of OCl^- and NH_4^+ form a new cluster that does not lie between $OCl^-$and NH_4^+. This outlying cluster can be explained by the fact that the reaction between $OCl^-$and NH_4^+ produces monochloramine *(NH_2Cl)* which has its own absorption band at 245 nm, as shown in Fig. 5.7.

The score plot of the third PC (PC3) against the fourth PC (PC4) is shown in Fig. 5.6. In this plot a cluster of hypochlorite ion data is well separated from a cluster of ammonium ion data while these data are grouped together in the score plot for PC1 and PC2. Further combinations of PCs may be used to differentiate the other groups. The use of PCA to quantify amounts of component is described in the next section.

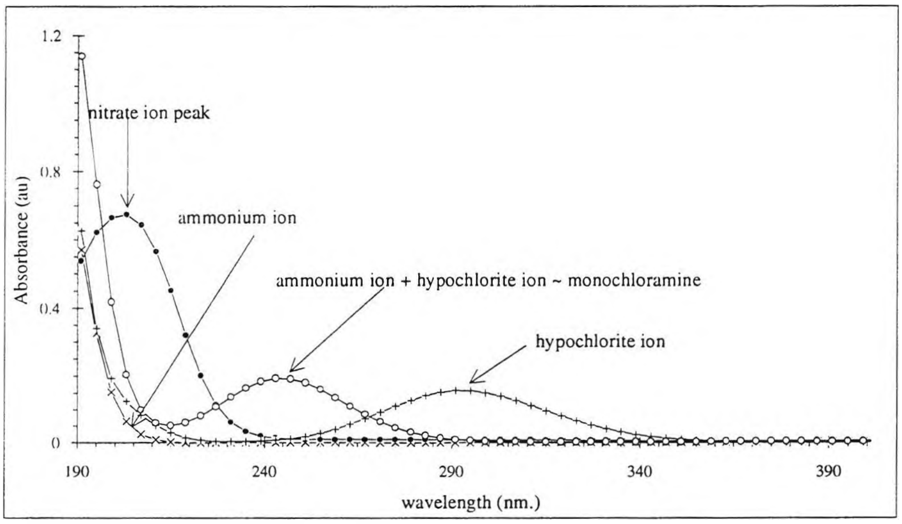


Fig. 5.7 Absorption spectra of the samples; NO_3^-, OCl^-, NH_4^+, and $OCl^- + NH_4^+$ *(NH_2Cl)*

5.5 PCA Quantitative Analysis

To determine the components and their concentration from a spectrum of an unknown sample requires calibration data from samples with known constituency and concentrations. Target Transformation Factor Analysis (TTFA) can then be applied. This technique uses the PCs of the training data set and concentration data to generate a calibration matrix. This calibration matrix can then be used directly to transform the spectra from an unknown sample to "best guess" estimates of concentration for the components in the sample.

For the different PCA trials, the training samples were separated in three categories as shown in Table 5.2. These categories consisted of sample mixtures of NO_3^- & NH_4^+, NO_3^- & OCl^-, and NO_3^- & OCl^- & NH_4^+. The first two trials are cases of linearly related data where no chemical reaction between nitrate and ammonium ion or nitrate and hypochlorite ion does take place. The third trial is non-linear: however it was considered as being linear by assuming that the reaction between the ammonium ion and hypochlorite ion to form monochloramine goes to completion. The results of these three trials were compared in order to evaluate the ability of PCA to predict the content of samples.

The procedure and result of these three trials are discussed in section 5.6.1, 5.6.2, and 5.6.3.

Table 5.2 Three categories of training samples.

Nitrate ion and Ammonium ion	Nitrate ion and Hypochlorite ion	Nitrate ion, Hypochlorite ion and Ammonium ion
1. Distilled Water	1. Distilled Water	1. Distilled Water
2. NO_3-A	2. NO_3-A	2. NO_3-A
3. NO_3-B	3. NO_3-B	3. NO_3-B
4. NO_3-C	4. NO_3-C	4. NO_3-C
5. NH_3-A	5. Cl_2-A	5. NH_3-A
6. NH_3-B	6. Cl_2-B	6. NH_3-B
7. NH_3-C	7. Cl_2-C	7. NH_3-C
8. NO_3-A + NH_3-A	8. NO_3-A + Cl_2-A	8. NO_3-A + NH_3-A
9. NO_3-A + NH_3-B	9. NO_3-A + Cl_2-B	9. NO_3-A + NH_3-B
10. NO_3-A + NH_3-C	10. NO_3-A + Cl_2-C	10. NO_3-A + NH_3-C
11. NO_3-B + NH_3-A	11. NO_3-B + Cl_2-A	11. NO_3-B + NH_3-A
12. NO_3-B + NH_3-B	12. NO_3-B + Cl_2-B	12. NO_3-B + NH_3-B
13. NO_3-B + NH_3-C	13. NO_3-B + Cl_2-C	13. NO_3-B + NH_3-C
14. NO_3-C + NH_3-A	14. NO_3-C + Cl_2-A	14. NO_3-C + NH_3-A
15. NO_3-C + NH_3-B	15. NO_3-C + Cl_2-B	15. NO_3-C + NH_3-B
16. NO_3-C + NH_3-C	16. NO_3-C + Cl_2-C	16. NO_3-C + NH_3-C
		17. Cl_2-A
		18. Cl_2-B
		19. Cl_2-C
		20. NO_3-A + Cl_2-A
		21. NO_3-A + Cl_2-B
		22. NO_3-A + Cl_2-C
		23. NO_3-B + Cl_2-A
		24. NO_3-B + Cl_2-B
		25. NO_3-B + Cl_2-C
		26. NO_3-C + Cl_2-A
		27. NO_3-C + Cl_2-B
		28. NO_3-C + Cl_2-C
		29. NO_3-A + Cl_2-A + NH_3-A
		30. NO_3-A + Cl_2-A + NH_3-B
		31. NO_3-A + Cl_2-A + NH_3-C
		32. NO_3-A + Cl_2-B + NH_3-A
		33. NO_3-A + Cl_2-B + NH_3-B
		34. NO_3-A + Cl_2-B + NH_3-C
		35. NO_3-A + Cl_2-C + NH_3-A
		36. NO_3-A + Cl_2-C + NH_3-B
		37. NO_3-A + Cl_2-C + NH_3-C
		38. NO_3-B + Cl_2-A + NH_3-A
		39. NO_3-B + Cl_2-A + NH_3-B
		40. NO_3-B + Cl_2-A + NH_3-C
		41. NO_3-B + Cl_2-B + NH_3-A
		42. NO_3-B + Cl_2-B + NH_3-B
		4$_3$. NO_3-B + Cl_2-B + NH_3-C
		44. NO_3-B + Cl_2-C + NH_3-A
		45. NO_3-B + Cl_2-C + NH_3-B
		46. NO_3-B + Cl_2-C + NH_3-C
		47. NO_3-C + Cl_2-A + NH_3-A
		48. NO_3-C + Cl_2-A + NH_3-B
		49. NO_3-C + Cl_2-A + NH_3-C
		50. NO_3-C + Cl_2-B + NH_3-A
		51. NO_3-C + Cl_2-B + NH_3-B
		52. NO_3-C + Cl_2-B + NH_3-C
		53. NO_3-C + Cl_2-C + NH_3-A
		54. NO_3-C + Cl_2-C + NH_3-B
		55. NO_3-C + Cl_2-C + NH_3-C
		56. Cl_2-A + NH_3-A
		57. Cl_2-A + NH_3-B
		58. Cl_2-A + NH_3-C
		59. Cl_2-B + NH_3-A
		60. Cl_2-B + NH_3-B
		61. Cl_2-B + NH_3-C
		62. Cl_2-C + NH_3-A
		63. Cl_2-C + NH_3-B
		64. Cl_2-C + NH_3-C

5.6 Results and Discussion

5.6.1 Nitrate ion and Ammonium ion

The first trial set consists of spectra of two components, NO_3^- and NH_4^+ which are linearly related to concentrations because there is no interaction between the components. Fig. 5.8 shows that most of the nitrate ion absorption peak can be seen at λ_{max} = 210 nm, whereas the ammonium ion peak is out of range of the HP-8452 spectrometer specification, and only the tail can be seen. The figure shows that the absorption plot for the mixture is a linear summation of the individual spectra.

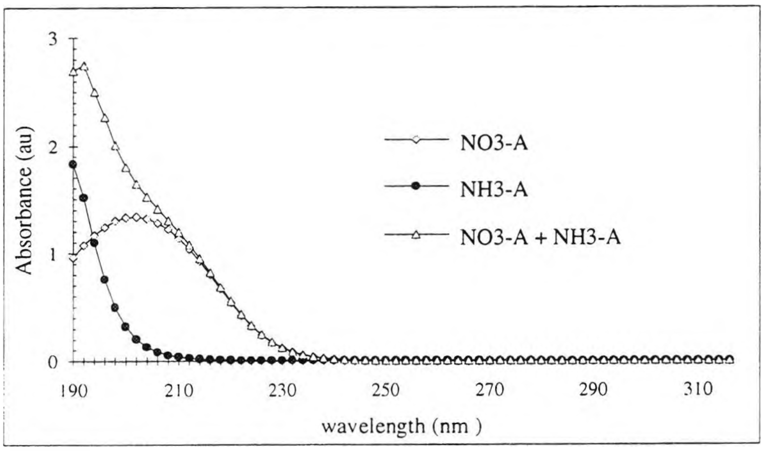


Fig. 5.8 The absorbance spectra of NO_3^- (7.75 mg/l) and NH_4^+ (105.83 mg/l)

The comparison between the predicted and the actual concentration for the nitrate and the ammonium ion is shown in Fig. 5.9 and predicting errors of training and testing sets are shown in Table 5.3. From the table it can be inferred that the accuracy of predicting for nitrate ion is very good at 94.16% and 93.57% while the accuracy for ammonium ion is worse at only 91.14% and 76.02%. The decreased accuracy in estimating ammonium ion is due to three reasons discussed below:

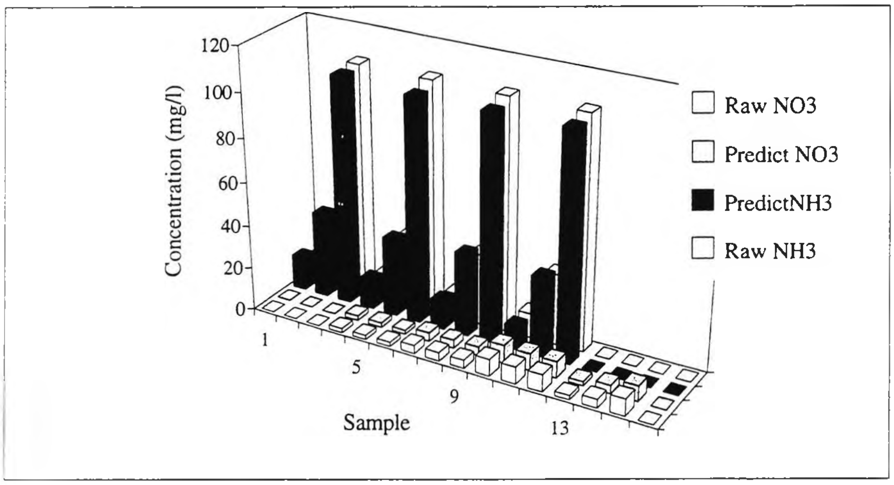


Fig. 5.9 The comparison between prediction and actual concentration.

Table 5.3 The predicting error from the first training set

Predicting Error	NO_3	NH_3
Training Set	5.84%	8.86%
Testing Set	6.43%	23.98%

Firstly, the measurements cannot cover the whole shape of the Guassian absorption peak of ammonium ion which centres at 180 nm and is thus beyond the measurement range of this study. Nitrate ion spectra have a well defined Guassian shape. These two peaks are illustrated in Fig. 5.10a and 5.10b.

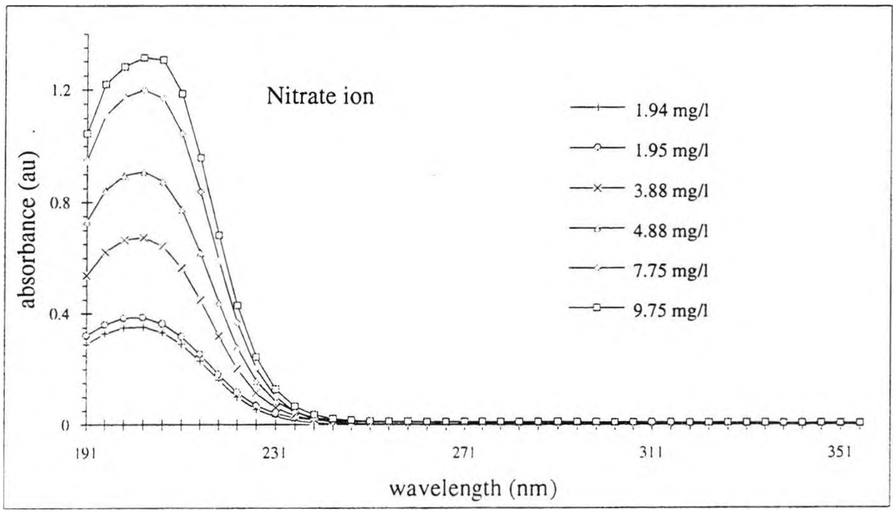


Fig. 5.10a Nitrate ion spectra

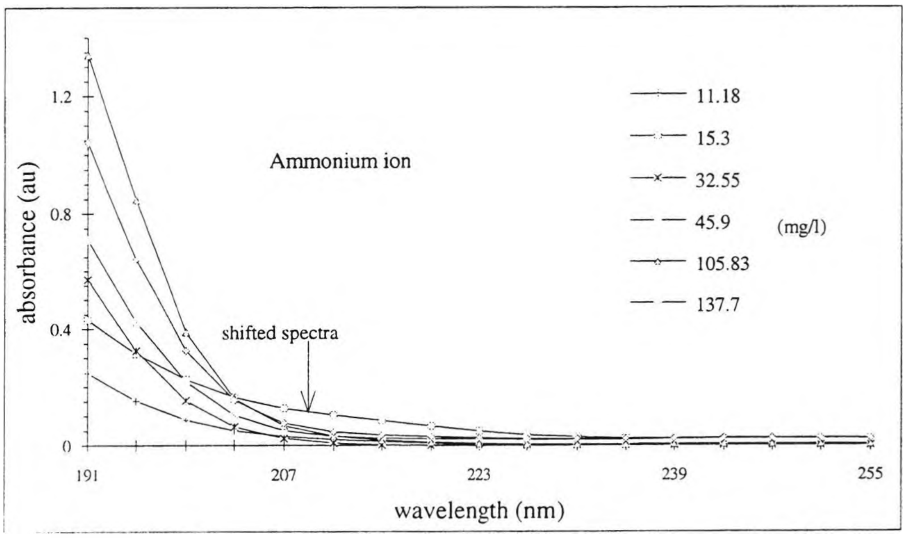


Fig. 5.10b Ammonium ion spectra

Secondly, The training and testing sets were prepared and measured at different times and with different blank sample and different dark current signals. These different conditions reflect the real situation which was encountered in using spectrometry in our remote on-line system. As a result, the 15.3 mg/l ammonium ion spectrum did not correlate with concentration as did most of the ammonium ion spectra. Figure 5.10b shows the shifted spectra for the 15.3

mg/l ammonium ion. The plots of absorption at the peak wavelength, λ_{max}, against concentration of training and testing data are shown in Fig. 5.11a, 5.11b. The figures show that the deviation of the nitrate ion is less than that of the ammonium ion.

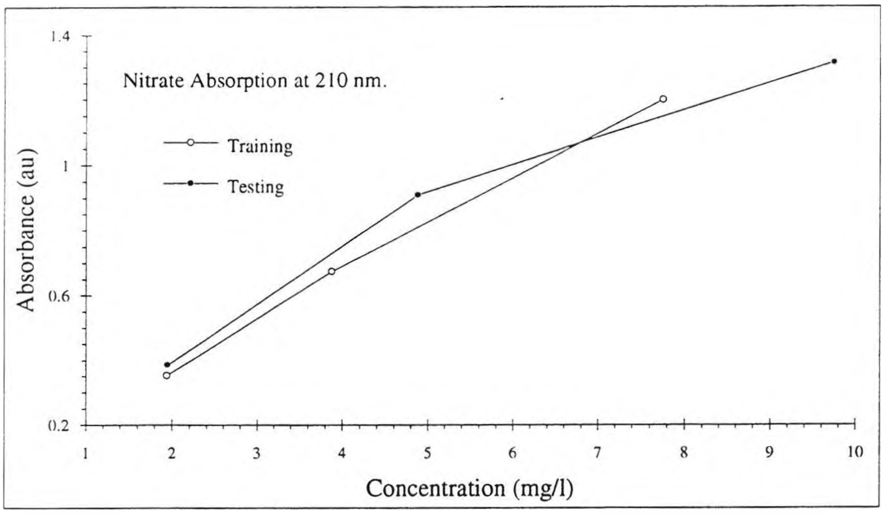


Fig. 5.11a. A plot of NO_3 absorption vs.concentration

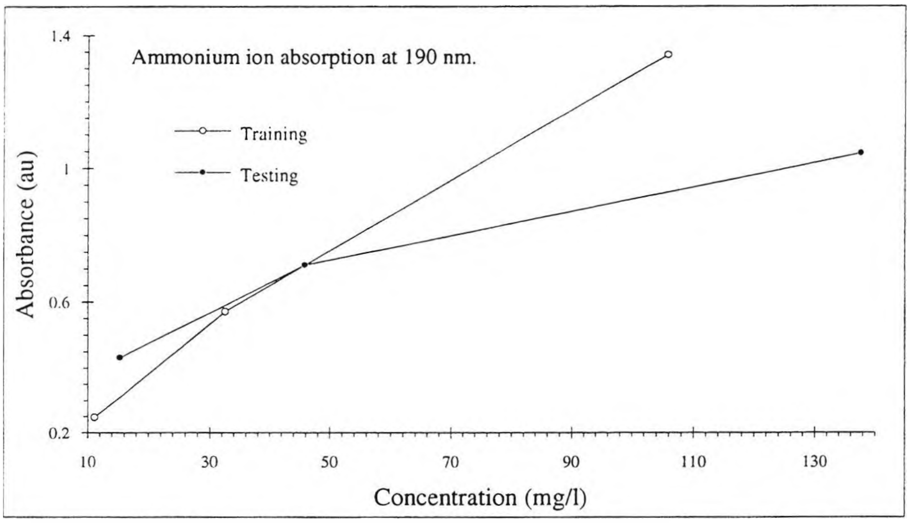


Fig. 5.11b. A plot of NH_3 absorption vs.concentration

Finally, this deviation in the individual components displayed in Fig. 5.10 is also reflected in the data from the mixtures. The deviation, explicit for the ammonium ion in Fig. 5.9b is also be seen for the mixtures of the nitrate and ammonium ions in Fig. 5.12.

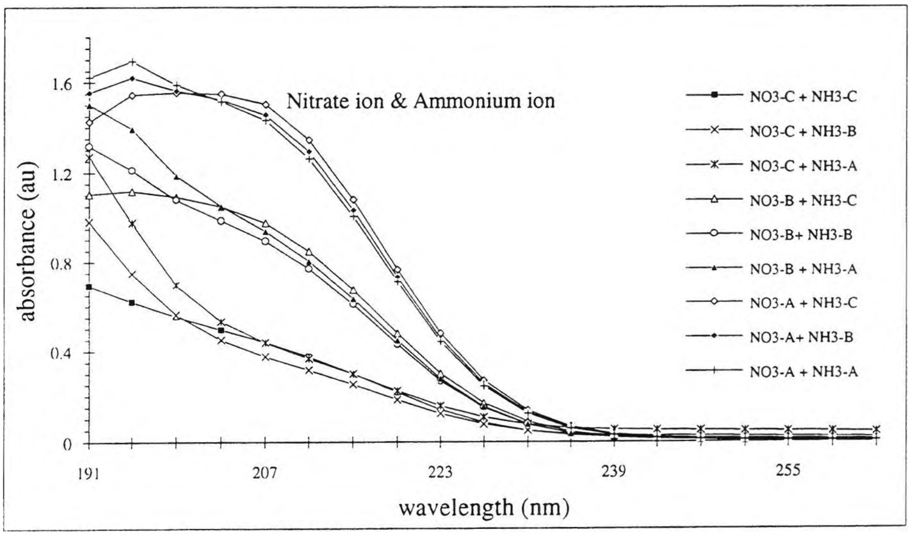


Fig. 5.12 Absorption spectra of mixtures of nitrate ion and ammonium ion.

5.6.2 Nitrate ion and Hypochlorite ion

The second set consists of 2 components; NO_3^- & OCl^- that are linearly related, which is similar to the first trial. In Fig. 5.13, the spectra of the hypochlorite ion show the main absorption peaks of the hypochlorite ion (290 nm) together with another chlorine-related absorption band in the 190-230 nm range. A comparison between the predicted and actual concentrations for this system is shown in Fig. 5.14 and errors for the training and testing set are shown in Table 5.4. The accuracy of predicting for the nitrate ion is very good at 96.15% and 93.38%, but that for the hypochlorite ion is much worse at 84.10% and 66.62%.

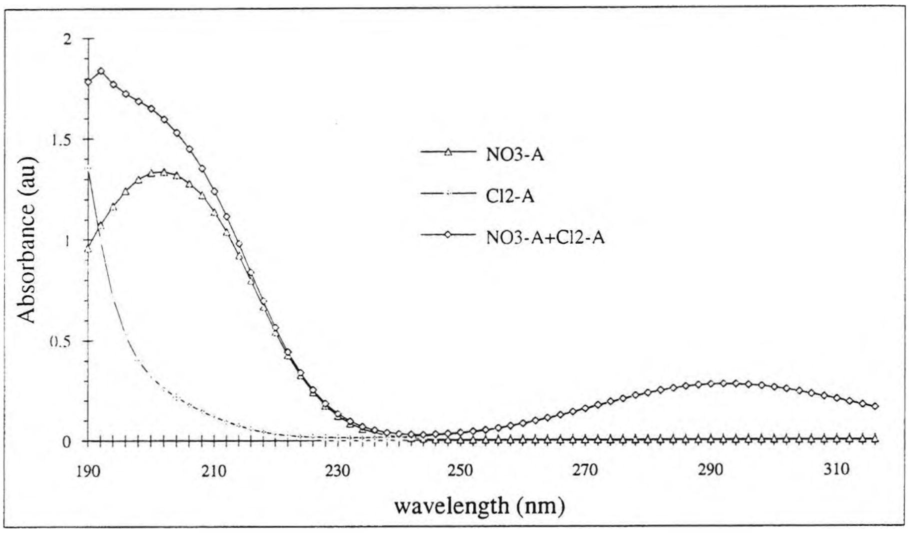


Fig. 5.13 The absorbance spectra of NO_3 7.75 mg/l and Cl_2 49.23 mg/l

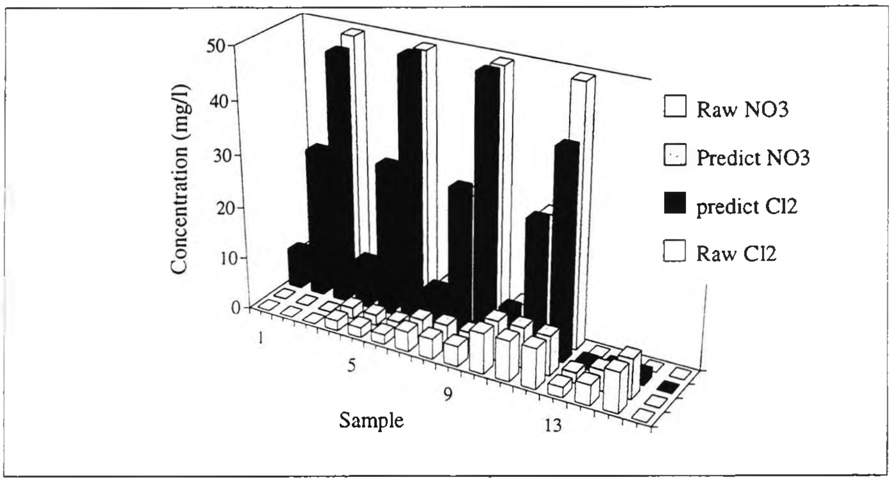


Fig. 5.14 The comparison between prediction and the actual concentration.

Table 5.4 The prediction error

Predicting Error	NO_3	Cl_2
Training Set	3.85%	15.90%
Testing Set	6.62%	34.38%

One of the reasons for the large error in the hypochlorite ion prediction is the existence of a second absorption peak at 190-230 nm, as shown in Fig. 5.15a. Fig. 5.15b also shows there is a large difference between the calibration curves of the training and testing sets for the hypochlorite ion at 195 nm. A further reason for this error is that this absorption between 191 - 230 nm interferes with the nitrate ion spectrum, as illustrated in Fig. 5.16.

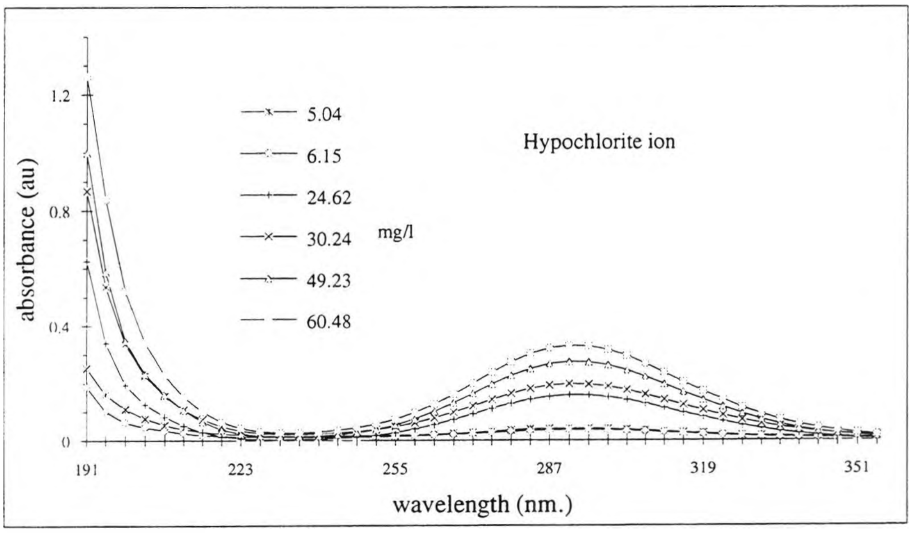


Fig. 5.15a Hypochlorite ion spectra.

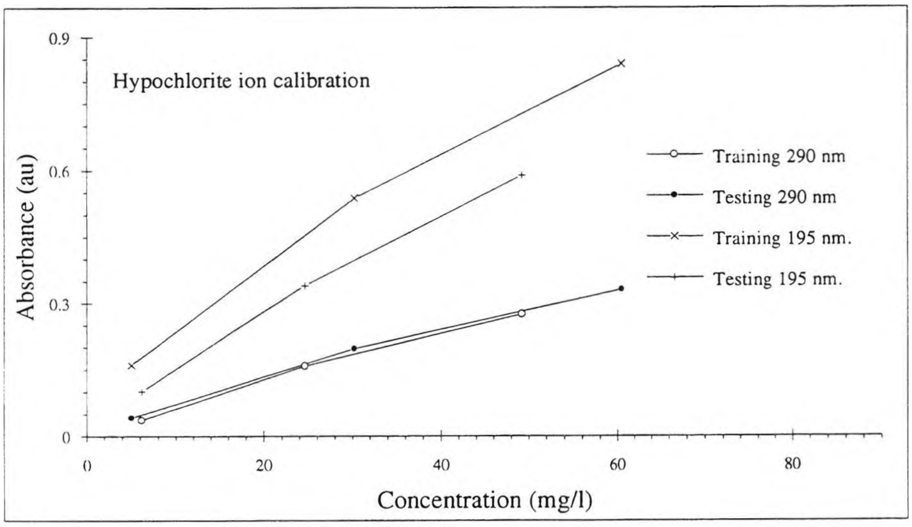


Fig. 5.15b A calibration plot at 295 nm

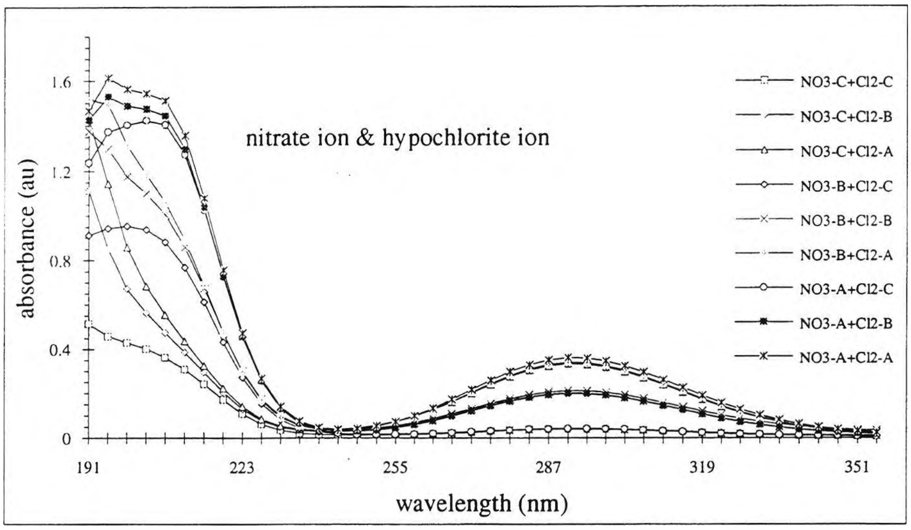


Fig. 5.16 Absorption spectra of mixtures between nitrate ion and hypochlorite ion

5.6.3 Nitrate ion, Hypochlorite ion and Ammonium ion

The third set consists of four components of the nitrate ion (NO_3^-), ammonium ion (NH_4^+), hypochlorite ion (OCl^-) and monochloramine (NH_2Cl). As discussed, this set is non-linear because of the chemical interaction between OCl^- and NH_4^+ to form NH_2Cl which absorbs in the UV at 245 nm, as shown in Fig. 5.17. The Fig 5.18 highlights the fact that when the hypochlorite ion and the ammonium ion mix together, the absorption peak of the hypochlorite and the ammonium ion decreases whereas the absorption of monochloramine increases. This is the result of the chemical reaction between these ions, as shown as Eq. (5.16).

$$OCl^- + NH_4^+ \Leftrightarrow NH_2Cl + H_2O \qquad ....(5.16)$$

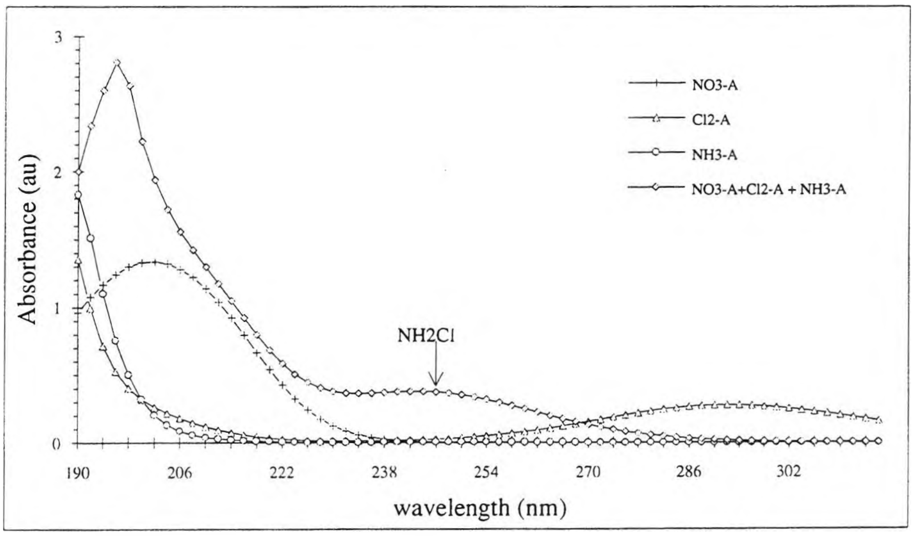


Fig. 5.17 The absorbance spectra of NO_3, NH_3, Cl_2 and NH_2Cl

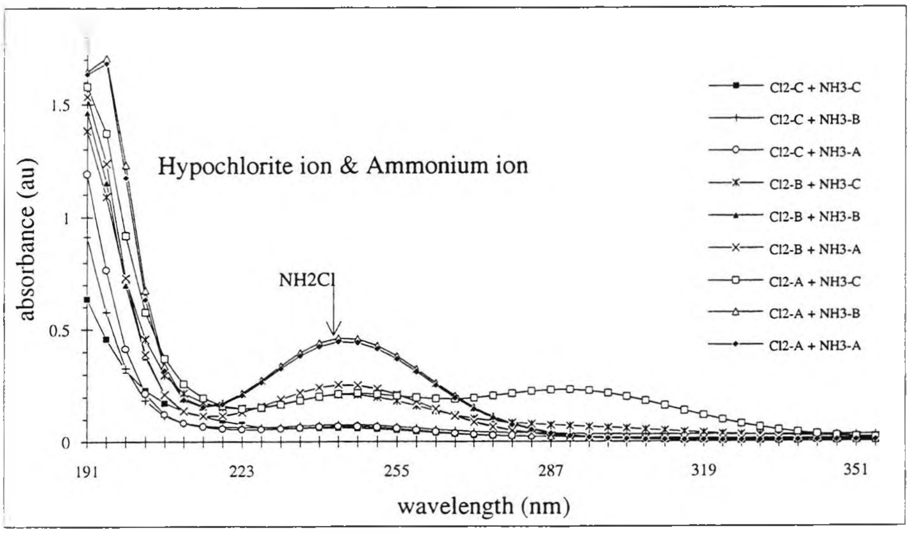


Fig. 5.18 The absorbance spectra between NH_3 and Cl_2 which form NH_2Cl

In an attempt to analyse the data further by linear methods, the NH_2Cl species is taken into account and added to the range of possible species. However for any particular mixture of hypochlorite ion and ammonium ion, the concentration of NH_2Cl is unknown. In this trial, it is assumed that the reaction goes to completion so it means that one molecular weight of the hypochlorite ion and ammonium ion is decreased while one molecular weight of monochloramine is increased, as presented in Eq.(5.16).

A comparison between the predicted and the actual concentrations is shown in Fig. 5.19 with the prediction error shown in Table 5.5. The results show that with the above approximation a very large value of error results in predicting the concentrations, with values of 95.44% for ammonium ion, 150.45% for hypochlorite ion and 95.44% for monochloramine. This implies that the assumptions made to implement the PCA are invalid and that other dynamic factors including pH, temperature and time must be accounted for.

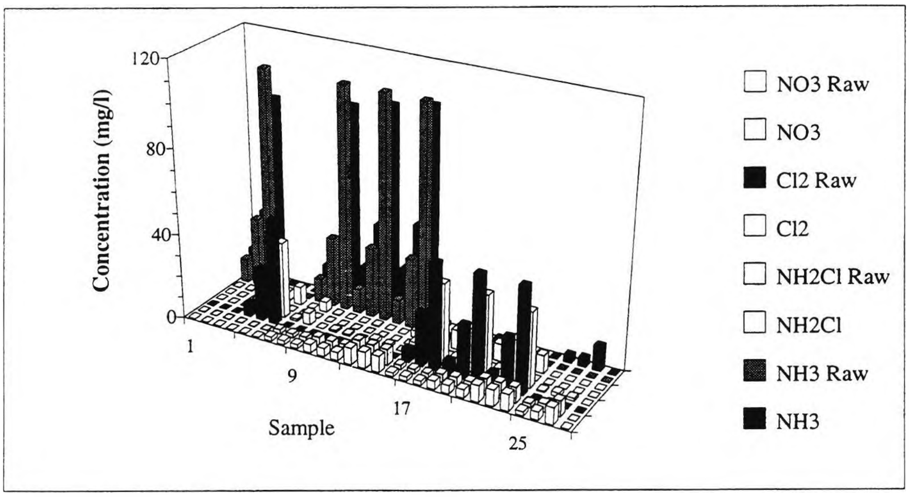


Fig. 5.19 The predicting v.s. actual concentration(sample#1-28).

Table 5.5 The prediction error

Prediction Error	NO$_3$	NH$_3$	NH$_2$Cl	Cl$_2$
Training Set	5.93%	60.28%	46.65%	127.69%
Testing Set	19.65%	95.44%	95.44%	150.45%

5.7 Discussion

In the data sets where a linear response was observed, PCA and TTFA gave a good performance as exemplified particularly in the prediction of the nitrate ion in the first two trials where the nitrate ion spectrum has a nearly Guassian shape and the absorption is linearly related to concentration. Deviations observed in the spectra of the testing set result in a decreased performance. There are two reasons for such deviations which were observed in the first two trials: The main reason is the fact that the training and testing sets are prepared and measured at different times and with different blank and dark current signals. This non-repeatibility reflects the real situation to be expected in the use of such method in industry, where the environment of the sensor system can be expected to vary with time. Another reason is the deviation from the Beer-Lambert law. These deviation effects include:

(a) the effect of stray light,

(b) the effect of concentration-dependent peak shifts, and

(c) the effect of concentration-dependent bandwidth changes, as discussed earlier.

These effects combine to make the determination of absorption by broad spectroscopy, where they can be accounted for within the data analysis, and more reliable than in the single wavelength measurement. In many such single wavelength sensors, these sort of effects are not accounted for and so the measurement is subject to a higher degree of error.

In the data set where a non-linear relationship between absorbance and concentrations was observed, PCA and TTFA gave a poor performance as shown in Table 5.5. For instance, the chemical interaction that occurs with mixtures of hypochlorite ion and ammonium ion dramatically changes the spectral response with the occurrence of the new species, monochloramine, so that the absorption peaks are no longer linearly related to the concentration of the constituents. In this study of PCA, the data are treated as linear by assuming that the reaction between the hypochlorite ion and ammonium ion goes to completion, whereby the amount of monochloramine can be accounted as well as the consumption of ammonium ion and hypochlorite ion. However, the problem of interaction between the components make the system more complicated than one which could be analysed by linear analysis. In reality, the situation may not go in completion and parameters such as pH, temperature and time, are required for the proper determination of the constituents during the chemical reaction. These parameters require laboratory-based batch sample measurements outside the scope of the automatic and real-time accident reporting which thus is a goal of this work.

Most problems of measurement are non-linear but scientists always try to idealize them as linear problems, because that makes the system easier to understand and manipulate. However many of the available non-linear analysis techniques are complex and limited in success in explaining phenomena in complex systems. A new approach to non-linear problems is Neural Networks which are implemented by an algorithm inspired by research into the function of the human brain. Moreover, neural networks can provide, in principle, methods for chemical analysis that require no detailed knowledge of the chemical mechanism involved because they can take available data and learn from it. With the last problem of having to analyze non-linear problems, a new approach including Neural Networks gives a possible way forward. Therefore the use of neural network technique is investigated to determine whether the data can be classified for the complicated situation described above, or not, as will be discussed in the next chapter.

CHAPTER 6

The Application of Neural Networks in Water and Environment Monitoring

"Artificial intelligence is the study of how to make computers do things at which, at the moment, people are better."

Patrick Henry Winston, Artificial Intelligence[66]

6.0 Introduction

What makes people smarter than machines? The human brain employs a basic computational architecture that is well suited to deal with "natural" information processing tasks. The ability of the human brain to learn, to retrieve contextual information from memory, to make plans, to carry out relevant actions, and to do a wide range of other natural cognitive tasks far surpasses the ability of any computer. This fascinating ability has led psychologists, mathematicians, and computer scientists to strive to find a model that can illustrate the functional characteristics of the human brain. Mathematical models inspired by the neurophysical structure of the brain have been created and simulated on computers. These mathematical models of neural activity lead into the field of Artificial Neural Networks.

Having been used experimentally for decades, neural networks are now reputedly "a solution in search of a problem"[*] and more recently have been used for a range of practical applications. This trend can only accelerate now that specialized hardware is available to speed neural network applications development. Many actual applications that use neural networks, often do so without public acknowledgment to preserve competitive advantage, for examples including applications such as pattern recognition, classification, function estimation, data compression, feature extraction, and statistical clustering.

There are several statistical clustering techniques such as the Partial Least Squares, Factor Analysis, PCA, Bayes, Neural networks, etc. PCA is the most widely used but it was shown to have failed in the non-linear problem discussed in Chapter 5. Therefore neural networks which are non-linear methods were used to analyse the UV-Vis spectral data in comparison with the PCA method. Neural networks can provide, in principle, methods for chemical analysis that require no detailed knowledge of the chemical mechanism involved because they can take data and learn from them. Therefore, they can be applied to a problem which is intractable to other methods where the only alternative would be laboratory-based batch sample measurements. Another attractive ability of neural networks is that they can generalize after training, and they can handle imperfect or incomplete data, providing a degree of fault tolerance.

[*] *an epithet applied to the laser in the early 1960s.*

There are several positive results in the application of neural networks for classification and quantitative analysis of spectroscopic data that have encouraged the author to use a neural-network method in this application to water monitoring. However, all of these successes suffer from one very serious problem in that solutions found with neural network analysis are not unique, with the added problem that there is no sure method that the best solution has been found in any specific case. These problems have been a major concern of researches in the application of neural networks. These problems imply that each application of neural networks must be taken on it own merits. Despite these difficulties, the successful application of neural network is worth applying effort to as the maxim goes "*A ship in port is safe, but that's not what ships are for.*"

This chapter is concerned with a feasibility study of neural networks based on an analysis of the data presented in Chapter 5 by using back propagation neural networks. The organization of this chapter is outlined as following:

- Section 6.1 presents a review of the neural network applications in spectroscopic data analysis including pre-processing techniques and data generating methods which are used as guidelines for this study.
- Section 6.2 discusses the basics of neural computing, including its similarities to neuro-physical models. This section also explains the development of neural networks from processing elements to layers to networks.
- Section 6.3 defines backpropagation learning and reveals how the network is trained. This section includes the details of backpropagation learning algorithm.

- Section 6.4 explains the attractive abilities of neural networks, which differ radically from those of standard software techniques.
- Section 6.5 discusses the weaknesses of neural networks, that can cause difficulties and render them unsuitable for some tasks.
- Section 6.6 describes the feasibility study of neural networks based on an analysis of the data presented in Chapter 5 by employing back propagation neural networks. This section shows the beginning of the creation of a data set which represents input and output vectors. This illustrates the development of pre-processing techniques from spectral data to PCA score data and derivative spectra to encoding data. This section also includes the development of data-generating methods.
- Section 6.7 gives a conclusion of feasibility study of the application in water monitoring, including suggestions for extended and further studies.

6.1 Application of Neural Networks to Analytical methods and Waste Water Pollution Review

The hundreds of actual applications that use neural networks can be categorized in terms of certain basic abilities in relation to how they employ the network itself. The two largest categories that result cover the use of a neural network for pattern recognition and function estimation. Other categories include data compression, feature extraction, and statistical clustering. Each basic capability enables many specific applications, for instance a neural network classifier combined with a spectrometer can be applied to analyze multi-component mixtures. Most of the current study on neural network learning is centred on the back-propagation algorithm. Its inherent ability to build arbitrary nonlinear boundaries between input and output layer representation allows its use for the estimation of chemical composition through spectroscopic data analysis. This is a new area of application. Previously, such estimation has been obtained using a mathematical model which utilizes the Principal Component Analysis or the Partial Least Square algorithm.

Recently, the interest in applying artificial neural networks in this area has steadily increased. There are several examples of successful uses in analytical chemistry[22-37] and of waste water pollution researches[38-40] using neural networks for the qualitative and quantitative analysis of multicomponents. Some of these examples have used neural networks directly to analyse original data. Other examples have included pre-processing to transform the data so that it becomes easier for the network to learn from them. Pre-processing techniques include manual selection of sensitive data, averaging, using Fourier Transforms, PCA etc. Mostly examples have preprocessing to reduce the dimensions of the data, otherwise the data would result in an impractically huge neural network. A large number of data patterns are still required for training and this affects the cost of gathering data. Therefore, in some examples, data are generated by adding noise into the original data to increase the number of available data patterns of training set. The addition of noise to a training set is analogous to training with a larger data set, as long as the added noise is representative of the "real world" noise. This is also sometimes done with testing data to evaluate the generalized performance of the networks. The following are examples of applications of neural networks in specific research which are used as guidelines to develop a suitable basis for this study.

Gemperline et al.[28] developed UV-Vis spectroscopic assays for routine determination of the active ingredients and preservatives in pharmaceutical products. In this work, backpropagation network parameters were optimized by using simulated data. The effect of random errors in the concentration variables and in the response variables was investigated. The Principal Component Regression (PCR) results for the two preservatives in the concentrated samples were worse when compared to the results obtained with the neural network because the nonlinear response presented in these spectra was inadequately modelled by PCR. This study indicated that when a nonlinear response due to solute interactions or nonlinear instrumental response functions is present, artificial neural networks may be capable of giving superior performance for spectroscopic calibration.

One year later (1991), Gamperline *et al.*[21] used training data which was simulated from experimental data by adding non-linear effects such as stray light, wavelength shifts, and absorption bandwidth changes. They used an orthogonal transformation of the input variables to improve significantly the neural network training speed and to reduce calibration error.

Borggaard and Thodberg[22] announced a new method termed "optimal minimal neural-network interpretation of spectra" (OMNIS). OMNIS was unique in several respects. Firstly, it employed PCA as a preprocessor to the use of the neural network.

The neural network contained direct connections from the input to the output, ensuring that OMNIS was a true generalization of PCR. Secondly, the neural network size was optimized so that the resulting solution contained the minimum of connections necessary to interpret the data. Finally, OMNIS was based on recent insights in neural network research showing that the deliberate search for the minimum network compatible with the data was a way of obtaining the optimal generalization ability. As a result OMNIS gave the best results, where in comparison to using PCR and PLS (Partial Least Squares) on two NIR calibration data sets, OMNIS was demonstrated to reduce the standard error of prediction by 50% to 75%.

Liu et al.[29] reported on chemometric data analysis using artificial Neural Networks in their paper of 1993. An NIR spectrum that contained 3761 points was reduced to 2000 points by selecting the active ranges of the spectra and then further reduced to 190 points by a moving average method. A 0.5%- 5% random noise factor was added to study the effect of input noise on the network performance. The results concluded that the use of neural networks to estimate the concentration of chemical components through spectroscopic signatures provides a high degree of accuracy and the neural network was shown to be more noise tolerant.

Ham et al.[31] improved the method for detection of biological substances by using a hybrid neural network and infrared absorption spectroscopy. The identification of unknown substances was based on the resonance absorption peaks known as *"fingerprints"* that were unique to different molecules. However some complex molecules possessed a large number of absorption peaks in the infrared which could overlap with those of other substances. A neural network detection method had been used to detect and classify concentrations of glucose in a normal saline solution. The testing data were generated by adding Gaussian noise with 0.0008 - 0.0012 variance. The performance showed a 97% accuracy when the Gaussian noise variance was less than 0.001.

Peel et al.[23] suggested a fast procedure for the training of neural networks by applying PCA into the network training philosophy. This involved designing the number of hidden nodes as the number of PCs and initializing the weights between the input and hidden layers as the elements of the corresponding eigenvectors. The performance had been evaluated in an estimation of product conversion in a continuous stirred tank reactor. It confirmed that the Neural network shows sufficient accuracy to approximate the non-linear process.

Sharma et al.[24] compared two learning paradigms: the back-propagation algorithm for supervised and Kohonen's Learning Vector Quantization(LVQ) for unsupervised use in order to classify and cluster amino acid information from Nuclear Magnetic Resonance (NMR) spectroscopic data. As the result of the research it was concluded that back-propagation has a high performance when given patterns similar to those on which it was trained but it does not have ability to recognize new categories of patterns. LVQ is generally useful when the amount of input data is large, relative to the number of clusters required. They also observed that Neural Network tuning was one of the most difficult aspects of the neural network approach.

Azoff [67] used PCA in preprocessing input data for diffraction tomography. He pointed out the advantage of PCA preprocessing and showed that it provides further improvement in convergence, a reduction in the number of input nodes and further the number of patterns required for training may be decreased without risking "over-fitting."

Errington and Graham[27] used a two-step neural network procedure to classify image from chromosomes. The first involved classification of a chromosome, independent of the other chromosomes in a cell. For this task the Multi-Layer Perceptron (MLP) was selected. Outputs from the first network were applied in the second stage. In this second stage classification, a competitive learning method was used as a post-classifier for a separate test set of mis-classified chromosomes. The performance of the neural network classifier was 2%-5% better than that of the statistical classifier, in this example.

Ricard et al.[68] investigated the combination of an expert system and neural network in two distinctly different manners. The first one in which the network was used in cooperation with the expert system to analyze each unknown, and the second was where the network was used instead to help in the creation of the rules that the expert system would use.

In the first case, the neural network was used directly and simultaneously in connection with the expert system. After its training was completed, it was given the spectrum of an unknown compound as an input vector, and the output (real numbers) was categorized to obtain symbolic conclusions (such as definitively absent, probably absent, not classified, probably present, definitively present) relative to the structural components of the unknown. These conclusions were entered into the database of facts that was then used by the expert system. The prepocessing by neural network is well-suited to spectrometric studies because the expert system can use symbolic information instead of whole spectra as facts on which to apply its reasoning.

In the second case, the neural network can be used indirectly to help the expert system identify spectral features by taking information from the trained network into consideration in writing the rules for the expert system. The type of information that was required from the network for this purpose was correlation information from the connection weights obtained by the trained network. With an interface between the neural network and the expert system, the interpretation of infrared spectra that requires the user 'read' the spectrum and make a verbal description to the expert system can be removed.

Sundgren et al.[32] compared two multivariate analysis methods, partial least squares and back-propagation neural network for quantification of individual components in a gas mixture. A gas sensor array with six metal-oxide semiconductor field-effect-transistors was exposed to gas mixture of hydrogen, ammonia, ethanol and ethylene. The neural network appeared to give better results.

Glick and Hieftje[14] used a back-propagation neural network for the classification of metal alloys based on their elemental constituents. Glow discharge-atomic emission spectra obtained with a photodiode array spectrometer were used in multivariate calibrations for 7 elements in alloys. The neural network approach showed a slightly better ability to classify samples when compared with the PCA method. They also concluded that "when noise was added to the input patterns during network training, the network was able to generalize and to assign unknown alloys to the appropriate class, even when determined values and their inherent error were presented. The addition of noise to a training set is analogous to training with a larger data set, as long as the added noise is representative of the real-world noise."

Boger and Karpas[30] derived quantitative information from ion mobility spectra of dimethylformamide, bromine, and hydrogen fluoride. In this case, they found that the training set was smaller than that of the theoretical requirement but also was sufficient for good learning. Theoretically, the size of training set should be at least equal to the number of connections. He explained that the theoretical requirement assumes that no relation exists between inputs, which apparently is not the case for mobility spectra. He also announced that the use of the neural network did not require a detailed knowledge of the ion chemistry of the measured system as is needed to identify peaks using traditional methods.

Orlov et al.[40] applied neural networks to determine organic pollution in natural waters by fluorescent spectroscopy. They formulated three-step procedures for the determination of pollutant concentrations comprising the classification of a pollutant, its identification, and concentration determination. They claimed that the net was capable of giving adequate answers in their work.

Beaverstock[69] published '*It takes knowledge to apply neural network for control*'. This paper described a practical approach for implementing artificially intelligent process control functions based on a unique combination of rule-based expert systems and neural network technology. The system had been successfully applied to a complex pulp and paper process.

The above successful examples show the wide range of possibilities of applying neural networks to spectral analysis. The present study on UV-Vis spectra from nitrate, ammonia and chlorine in aqueous solutions gives rise to a much more difficult data analysis than those considered above. This results from two major factors. First, the UV-Vis absorption peaks of species dissolved in water are very broad and this leads to difficulties in identifying different individual species because they most likely overlap the absorption of another, giving a small signal at a nearby wavelength. Second, there are chemical interactions between the components in the mixtures in the cases considered in this work, again modifying the spectra expected. However, with the prime ability of neural networks that they can take data and learn from it without an explanatory requirement of the source of data, it is worthwhile undertaking a study to apply them, because the alternative requires a detailed knowledge of the varying mechanisms and the dynamic aspects of the chemistry of the range of studied.

The feasibility study on the neural network application to water monitoring will be described in section 6.6, before which the basics of neural networks will be given to establish here the fundamentals of the operation of the neural network, the backpropagation learning rules, and the capabilities and weaknesses of neural networks for this application.

6.2 Neural networks basics

> *"The human brain uses a type of circuitry that is very slow ... at least 10,000 times slower than a digital computer. On the other hand, the degree of parallelism vastly outstrips any computer architecture we have yet to design...For such tasks as vision, language, and motor control, the brain is more powerful than 1,000 supercomputers, yet for certain simple tasks such as multiplying digital numbers, it is less powerful than the 4-bit microprocessor found in a ten-dollar calculator."*
>
> *Raymond Kurzweil,* The Age of Intelligent Machines[41]

Teuvo Kohonen [70] has defined neural networks as : *"massively parallel, interconnected networks of simple elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same ways as the biological nervous systems do*.". This implies that the basic principles of neural computing come from the physical structure of the brain. Because neural network models are inspired by the brain, there is a need to be concerned with an understanding of neuro-physical structure of the brain before attempting to learn more about neural networks.

6.2.1 Physical structure of the brain

The brain is a massive communication network of cells call neurons. Fig. 6.1 shows a diagram of a neuron consists of three major parts: the dendrites, the axons and the cell body.

Dendrites act as the message receivers for the neuron. They receive and interpret chemical messages from the axons of other cells. These chemical messages can either stimulate or suppress a dendrite. Once a dendrite receives a combination of messages, it will send a signal to the cell body.

A Cell Body is the control centre of the neuron. Messages received through the dendrites are interpreted and the response to those messages is sent out through the axon.

A Axon acts as the message transmitter of a neuron. An electrical charge, called the action potential, is sent out through the axon which releases chemicals to stimulate or repress nearby dendrites. The axon of one neuron branches out to communicate with hundreds of other neurons.

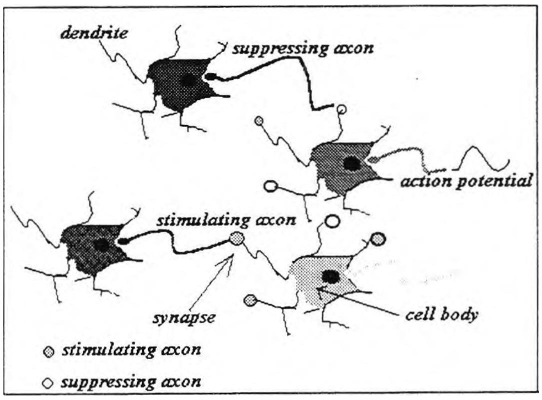


Fig. 6.1 The neuron

Each neuron is an autonomous unit within the brain. It continually receives inputs from other neurons through the dendrites, interprets the inputs in the cell body, and transmits a single response through the axon. The intelligence of the human brain lies in its massively interconnected structure. Thus, each neuron is not intelligent, but the interaction within a network of neurons may be considered to represent intelligence. This intelligence is distributed in several places, mainly in the pattern of connections between neurons, and in the strength of the connection between neurons.

Learning in the human brain takes place through changes in the connections between neurons. New connections may grow or the strength of an existing connector may be altered at the synapse, which is the space between the axon and dendrite. Therefore, changing the pattern and strength of neuronal connections changes the information held by the brain.

It is on the strength of such a simple understanding of neurophysical concepts, that forms an adequate basis to discuss a general model of neural networks.

6.2.2 General neural computing model

The basic unit of a neural computing model is the processing element. As shown in Fig. 6.2, a process element consists of five major parts: inputs, weights, combining function, transfer function, and outputs. Table 6.1 compares the corresponding parts of the human neural system and that of the computing neural system.

Details of the components of a processing element are as follows:

Inputs bring information into the processing element. Inputs can come from other processing elements, or itself, or sources external to the neural network. The inputs can be weighted according to the source of the input.

Weights determine how much influence an input has on the processing element. The strength of a signal from one element to another is modified by the weight of the connection between the elements. Therefore, weights have a direct effect on the degree of influence one element has on another. An input is usually combined with its corresponding weight in the so-called combining function, discussed below.

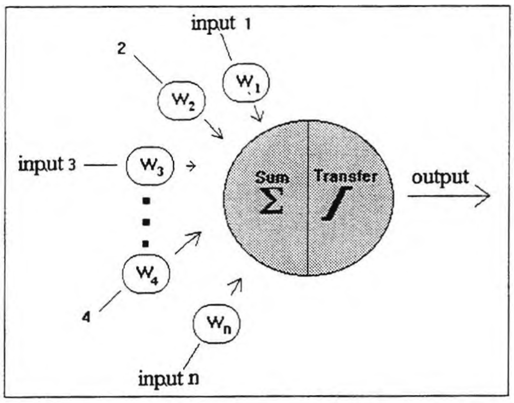


Fig. 6.2 A simple neuron computing processing element

Table 6.1 Comparison between Human and Computing Neural system

Neural System	**Neural Computing System**
Neuron	Processing Element
Dendrite	Combining Function
Cell Body	Transfer Function
Axons	Element Output
Synapses	Weights

A combining function combines the inputs and the weights in the processing element. The result is sent to the transfer function. The most common combining function is a weighted sum of the inputs:

$$A_j = \sum_i W_{ij} X_i$$

where W_{ij} is the weight corresponding to the connection between *ith* and *jth* element,

X_i is the input of *ith* element, and

A_j is the activation of *jth* element.

A Transfer function interprets the result of the combining function and determines the element output. The transfer function depends only on the results of the combining function.

The element output is the result of the transfer function and is spread to other connecting elements or external outputs. There is only one output value for each processing element.

The intelligence and information storage ability of a neural network emerges from its parallel, distributed and interactive structure. The intelligence of a neural network is stored in the pattern of connections between the elements, and the weight or strength of the connections between the elements. In the generalized neural network model, learning takes place through the alteration of the weights between processing elements. A *learning law* for the processing element determines how an element modifies the weights in response to examples and experience. Thus, through the learning law, the intelligence of the network changes.

This section has introduced the processing element which is a basic component of a neural network. The next section will explain how these processing elements can be combined to form a neural network with the exceptional capabilities.

6.2.3 Layers and Networks

Within a neural network, the processing elements or nodes are grouped together to form a structure called a layer. Each processing element in a layer has the same combining function, transfer function, and learning law. Neural networks emerge from the interconnection of one or more layers of processing elements.

A typical network consists of three kinds of layers:

(1) an input layer that receives inputs from the external world,

(2) one or more hidden layers, which are responsible for additional information processing, and

(3) an output layer that delivers the representation of the input after processing has occurred.

Information, or activation, moves between or within layers of a neural network. The output of an element in a layer can flow to any other element including:

- an element in the preceeding layer,
- an element in the following layer (feed-forward),
- an element in the same layer (lateral-feedback),
- itself (feedback), and
- external output.

The actual topology attained is determined by the learning laws associated with each layer. Once a network of processing elements has been formed, in the next paragraph a description will be given of how a network processes information through itself.

A neural network functions by accepting a pattern of activation at its input layer, processing the input through its layer and producing a pattern of activation at the output layer. Each individual processing element functions independently and in parallel with other process elements. The best way to understand how a neural network functions is to consider a relevant example, such as, is it possible with neural networks to answer the question:[71]

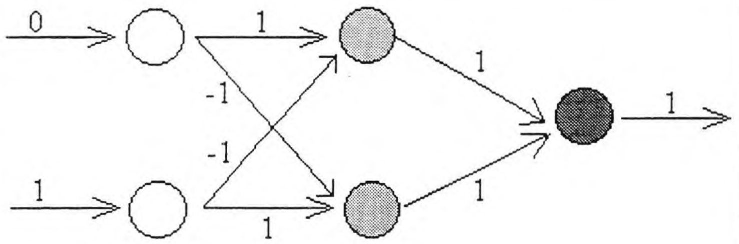
"Given a desired truth table, what values must the weights and transfer function have in order to achieve it?"

An exclusive-or logic (XOR) is explained as " *if X_1 or X_2 but not both X_1 and X_2 are true then Y is true*." This XOR problem can be illustrated by a truth table as shown below:

X_1 :	0	0	1	1
X_2 :	0	1	0	1
Y :	0	1	1	0

where 0 is represented "*false*" and 1 represented "*true*."

A simple network which is ready to determine the input output pairs for this truth table is shown in Fig. 6.3. By following the flow of activation through the network, it can be seen how such a network can process information.



Combination Function : $A_j = \sum_i W_{ij} X_i$

Transfer Function : Threshold Step Function
(if A < 1 then output = 0, otherwise output = 1)

Fig. 6.3 A network with an XOR problem[72]

In order for the network to model the XOR function the correct connections must be made, and they must have the proper weight. The network must be adjusted to fit its purpose. Because of the size and complexity of most practical networks, the connections and weights cannot be pre-set, but a network must be trained or self-organized into a topology in which its weights exhibit the desired characteristics.

To do this, learning rules are required, containing exact algorithms which determine how a processing element will change the weights of its connections in response to differences between actual and desired network outputs from a training pattern. The most well-known learning rule is the backpropagation algorithm, which is used in this study and which will be described in next section.

6.3 Backpropagation

> *"'Backprop' is an abbreviation for 'Backpropagation of error' which is the most widely used learning method for neural networks today. Although it has many disadvantages, which could be summarized in the sentence "You are almost not knowing what you are actually doing when using backpropagation" :-) it has pretty much success on practical applications and is relatively easy to apply."* -- Neural Net-Frequency Ask Question*

The back-propagation algorithm was first proposed by Rumelhart and McClelland.[73] The goal of the back-propagation algorithm is to teach the network to associate specific output patterns (target patterns) by adjusting the connection weights in order to minimize the error between the target output and actual output of the network. The schematic of a backpropagation and an idealize error surface are shown in Fig. 6.4 and Fig.6.5. A gradient descent algorithm which is generally used to perform the optimization is described in the following.

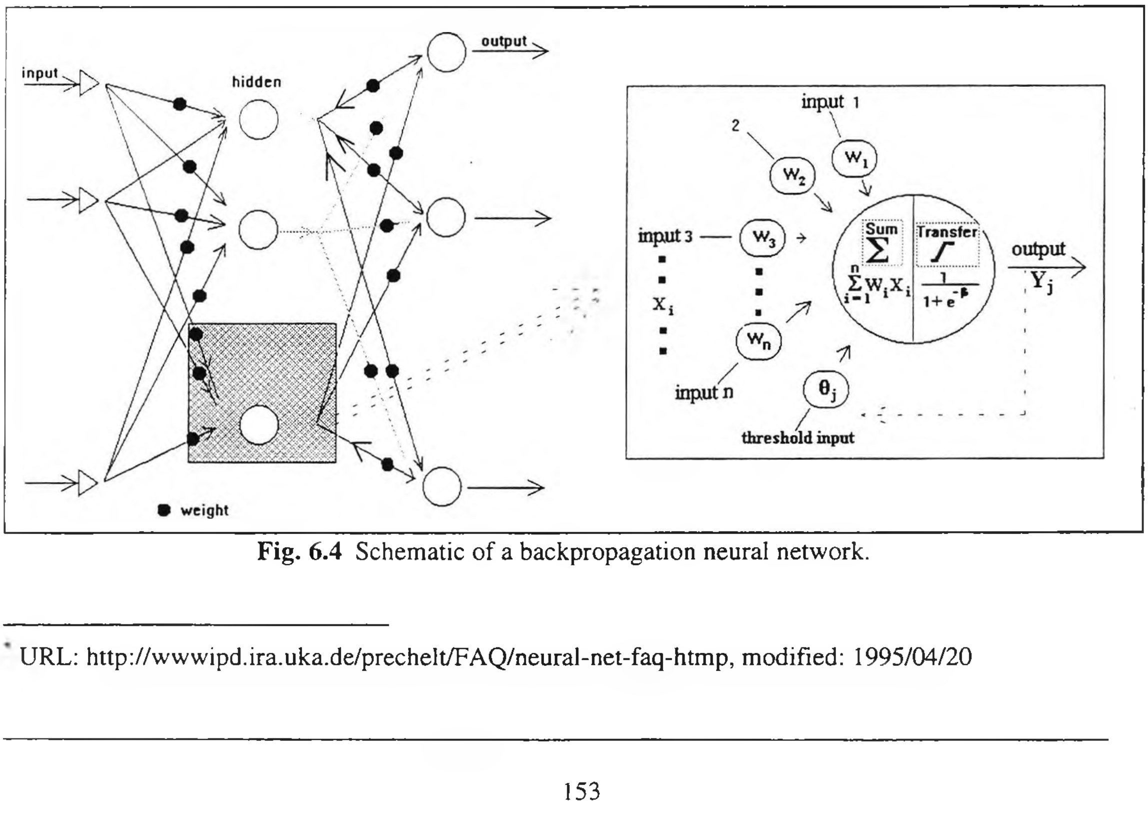


Fig. 6.4 Schematic of a backpropagation neural network.

* URL: http://wwwipd.ira.uka.de/prechelt/FAQ/neural-net-faq-htmp, modified: 1995/04/20

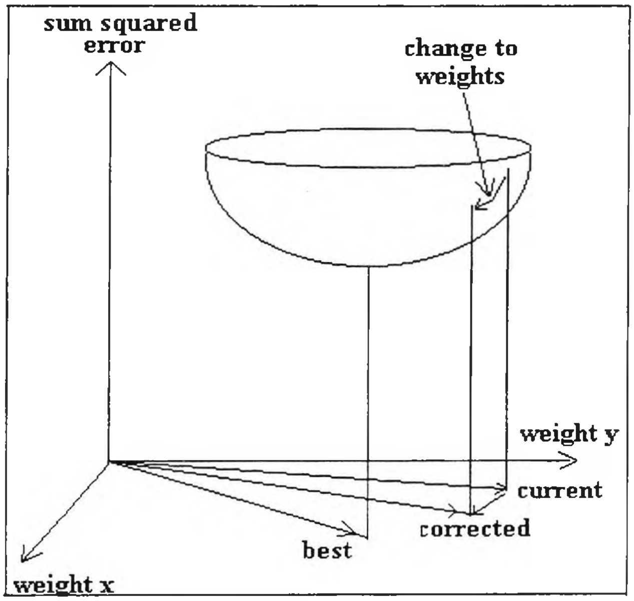


Fig. 6.5 Idealized error surface

During the learning procedure, a series of input patterns (e.g., UV-Vis spectra) with their corresponding output values (e.g., fractional chemical concentrations) are presented to the network in an iterative fashion while the weights are adjusted. The learning procedure is composed of two types of passes* : The forward-propagation (forward pass) and backward-propagation (reverse pass). In the forward-propagation, the network outputs are computed layer by layer, as shown in Fig. 6.4. The output of one layer then serves as the input to the next. The conventional notations for this are described by Eqs. 6.1 - 6.6. In this description, given an input vector **X** (x_1, x_2, x_3, ... , x_n) to the neural network, the $j^{\underline{th}}$ node in the hidden layer receives an input given by the sum of its weighted inputs and a net bias:

$$net_j = \sum_{i=1}^{n} w_{ij} x_i + \theta_j \qquad ....(6.1)$$

*Pass definition : a scan through a body of data -- computer dictionary where w_{ij} is the connection weight between $i^{\underline{th}}$ node in the input layer and $j^{\underline{th}}$ node in the hidden layer;

x_i is the $i^{\underline{th}}$ output from the input layer node; and

θ_j is the $j^{\underline{th}}$ node bias.

The output of the $j^{\underline{th}}$ hidden layer node is evaluated as

$$\mathrm{P}_j = f(net_j) \qquad ....(6.2)$$

where f is an activation function.

These nodes from the inputs to the output layer and the net value for an output layer *kth* node is given by:

$$net_k = \sum_{j=1}^{m} W_{jk} P_j + \theta_k \qquad ....(6.3)$$

The final output of $k^{\underline{th}}$ node is produced as follows:

$$O_k = f(net_k) \qquad .....(6.4)$$

The activation function *f(net)* used in this network is the sigmoidal function given by

$$f(x) = \frac{1}{1+e^{-\beta x}} \qquad \beta > 0 \qquad ....(6.5)$$

Thus, the output of $j^{\underline{th}}$ node in the hidden layer becomes:

$$\mathrm{P}_j = \frac{1}{1+\mathrm{e}^{-\beta(\mathrm{net}_j)}} \qquad ....(6.6)$$

In Eq. 6.6. the parameter β describes the shape of the sigmoidal function which is shown in Fig. 6.6. The bias for the nodes in hidden layer are used to shift the activation function along the x-axis. After the forward-propagation is completed, the error between the network output and the target values is calculated.

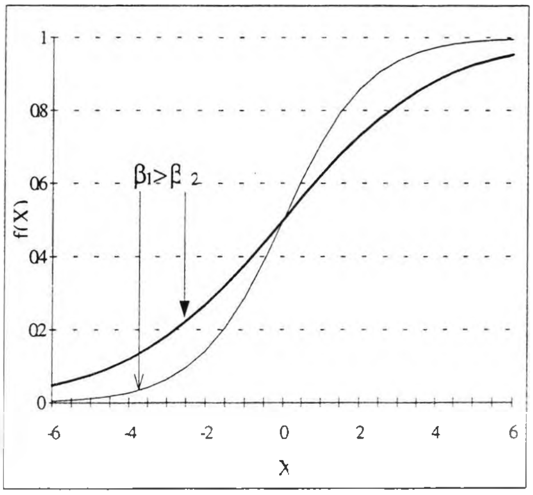


Fig. 6.6 Sigmoidal activation function.

In the back-propagation pass, the connection weights are corrected to reduce the error found after the forward propagation. This error-correction procedure is made from the output layer backward to the hidden layer. Here the Generalized Delta Rule is utilized to adjust the interconnection weights so as to reduce the square of the error for each pattern as rapidly as possible. One important parameter in the learning phase is the error value, δ, associated with each processing unit. This reflects the amount of error associated with that unit and is used during the weight-correction procedure. A large value of δ indicates that a large correction should be made to the connection weights.

The parameter δ is defined for the nodes in the output layer by:

$$\delta_{pk} = (t_{pk} - o_{pk}) f'(net_{pk}) \quad \text{....(6.7)}$$

where t_{pk} = target output of node k;

o_{pk} = actual network output of node k;

p = a subscript denoting the pattern number; and

$$f^{'} = \frac{\partial f}{\partial net}$$

The parameter δ is defined for the nodes in the hidden layer,

$$\delta_{pj} = (\sum \delta_{pk} w_{kj}) f'(net_{pj}) \qquad ....(6.8)$$

where w_{kj} = the connection weight between the node k in the output layer and the node j in the hidden layer.

The error parameter, δ_{pk}, can be evaluated in the highest layer of the network by using Eq.6.7. Then the error is propagated in a backward manner to the lower layers. This will allow the calculation of the parameter, δ_{pj}, for each hidden node in terms of the δ_{pk} at the upper layer. According to the generalized delta rule, the weight adjustments are made as follows:

$$\Delta W_{ji} = \eta \delta_{j} o_{i} \qquad ....(6.9)$$

where η is the learning rate parameter defining the step size of training,

δ_j is the error parameter of upper layer node j and

o_i is the active value of lower layer node i.

To improve the training time of the back-propagation algorithm and enhance the stability of the learning process, a momentum term is added to Eq.6.9, as shown in Eq. 6.10:

$$\Delta W_{ji}(n+1) = \eta \delta_{j} o_{i} + \alpha\left[W_{ji}(n) - W_{ji}(n-1)\right] \qquad ...(6.10)$$

The parameter, α, in this equation is the momentum coefficient, which determines the effect of past weight changes on the current weight. The momentum term trends to filter out the high curvature and thus it allows the effective weight steps to be bigger.

Usually α is initialized at a value around 0.9.[73] The integer*(n+1)* in Eq. 6.10 indicates the training iteration number.

Thus, new set of resultant connection weights is computed by using the set of equations:

$$W_{ji}(n+1) = W_{ji}(n) + \Delta W_{ji}(n+1) \qquad ...(6.11)$$

An expression similar to Eq.6.10 is used to adjust the connection weights between nodes in the input layer and the hidden layer. Prior to the start of the training, all the weights in the network are set to random values. Eqs. 6.10 and 6.11 are used to correct the connection weights in each training pattern until the error reaches an acceptable value for the entire training pattern set.

To summarize, the back-propagation learning rule involves a step-by-step procedure given below:

1. Initialize the connection weights to small random values in the range [-1,1].
2. Apply a pattern to the input layer.
3. Propagate the input pattern in a forward fashion through the network using Eqs.6.1-6.4 until the final network outputs are calculated.
4. Compute the error parameter δ for the nodes in the output and hidden layers using Eqs.6.7 and 6.8
5. Adjust the connection weights using the Generalised Delta Rule and Eqs.6.9 and 6.10
6. Repeat step2 → 5 for the next training pattern.

7. Stop training when the root mean square error of network output reaches an acceptable level.

There are several issues that need to be considered when utilizing this algorithm to train a neural network, for example the optimal number of hidden layers required for the corresponding number of nodes in each layer, the optimal values of the learning parameters which improves network training, and the format for presenting training data. These issues are considered in more detail in section 6.6 of this study.

Thus it can be seen how a neural network (back-propagation) can function as an information storage and retrieval tool, by adjusting weights. This self-adaptive ability can provide significant advantages over other techniques as will be described in the next section.

6.4 Neural Network Benefits

Neural networks learn the similarities between patterns directly from instances of the patterns themselves. That is, they infer solutions from data or classify the data without prior knowledge of the regularities in the data: *they extract the regularities empirically by weight adjustment.* This ability gives rise to the unique and powerful advantages of neural networks described below:

First, they are adaptive: they can take data and learn from them. Thus they infer solutions from the data presented to them, often capturing quite subtle relationships. This ability differs radically from standard software techniques because it does not depend on the programmer's prior knowledge of rules. Neural networks can reduce development time by learning underlying relationship even if they are difficult to find and describe. They can also be used to solve problems that lack existing solutions.

Second, neural networks can generalize: they can correctly process data that only broadly resembles the data they were trained on originally. Similarly, they can handle imperfect or incomplete data, providing a measure of fault tolerance. Generalization is useful in practical applications because real world data is on the whole, noisy.

Third, networks are nonlinear, in that they can capture complex interactions among the input variables in a system. In a linear system, changing a single input produces a proportional change in the output, and the effect of an input depends only on the value of that input. In a nonlinear system, on the other hand, the effect depends also on the values of other inputs, and the relationship is a higher-order function. It should be noticed that systems in the real world are often non-linear.

Fourth, neural networks are highly parallel: their numerous identical independent operations can be executed simultaneously.

Training of neural networks is an emerging discipline that involves aspects of programming, statistics, and signal processing. For some problems, using and training neural networks can be faster than writing traditional software. They can also be used to solve some problems that have resisted solution by other methods. Nonetheless, neural networks can be difficult to train and can be unsuitable for some tasks. These difficulties associated with neural networks will be described in the next section.

6.5 Neural network disadvantages

Developing a neural network is unlike developing software, because the network is trained, not programmed. Training one does not, in itself, require defining variables, creating loops, testing for conditions, running a compiler, or debugging code. Instead, the procedure starts with selecting, analyzing, and manipulating data, often using techniques from statistics and signal processing. This first step is usually the most crucial part of a neural network's development. While the network can infer relationships that its developer did not discern, it can find them only if the examples used to train are truly representative and include the effects of all the independent parameters. The acceptance and implementation of neural networks in data analysis has been impeded by the following reasons:

First, it can be difficult to account for their results. Neural networks are like human experts in that their decisions often cannot easily be explained. For one thing, the results depend on thousands of calculations involving the input pattern and the connection weights. Showing how the weights "cause" a result may be more complex than showing how a computer program works. For another, the values of the weights are themselves the result of a complex machine-learning procedure, making their origin hard to explain. For many applications, neural networks are more accurate than other solutions. They are like statistics in their aggregate behaviour, they tolerate imprecision, and are often effective in applications that lack other methods for obtaining solutions.

Second, training methods are imperfectly understood, because few definite rules exist for choosing appropriate values for training parameters. In a back-propagation network, choosing the optimum number of hidden nodes almost always depends on experiment. Still, there are many useful rules of thumb for obtaining reasonable starting values and guidance during training. Continuing research should improve our understanding of how to design and train neural networks. Nevertheless, the development process may always include manual fine tuning.

Third, neural networks can consume huge amounts of data and computer time, especially during training.

Finally, to use neural networks it must be possible to gather a sufficient sample of representative data. Otherwise, it may be difficult or impossible to train a network. It should be remembered that one part of the cost of implementing neural networks comes from the need to collect, analyze, and manipulate training data. The other part comes from the need to experiment with network parameters to find "good" values.

As mentioned above, the first and usually the longest step in neural network development, and also generally the most critical to eventual success, is the creation of a data set. The main tasks involved here in creating a data set include gathering raw data, analyzing them, selecting variables, and preprocessing the data so that the network can learn efficiently. These tasks will be described in detail in the next section discussing the application of a neural network to specific and unique application in this work -- the analysis of water monitoring incidents.

6.6 Neural networks feasibility study for Environment Monitoring of Water Quality

The purpose of the study reported here is to evaluate the use of neural network techniques for extracting information from UV-Vis spectra about chemical species present in water. This evaluation can be compared with that presented in Chapter 5 using PCA as this study used the same training and testing data sets of spectra from nitrate, chlorine, and ammonia in aqueous solution, as presented there. As mentioned, these mixtures are much more difficult to analyse than IR spectra because the broad absorption peaks in the UV-Vis spectrum overlap and because of chemical interactions between the species in the mixtures whereby the conventional method, PCA, was shown to be unsuccessful. Here, neural networks are evaluated because they can, in principle, be used for non-linear data sets and be able to self-organize. This latter ability may be important in this situation where all possible outcomes from the dynamic chemical mechanism cannot be known.

Networks are not programmed like algorithm software: instead they are trained through the repeated presentation of examples so that data are very important for neural networks. They are empirical systems which can recognize new examples of the patterns used to train them, but only if they resemble the training patterns. For example, one trained to recognize A and B can spot new As and Bs, but not Cs and Ds. Therefore the first step in development, and also generally the most critical for eventual success, is the creation of a data set. Tasks here include the gathering of raw data, its analysis, the selection of variables, and the preprocessing of the data so that the network can learn efficiently.

There are many many learning methods for neural networks now in use. Nobody knows exactly how many. New ones (at least variations of existing ones) are invented every week.* These methods can be categorized under two mains heading as:

(a) **supervised learning**, such as; Perceptron, Backpropagation (BP), brain-state-in-a-box (BSB), Boltzmann Machine (BM), adaptive logic networks (ALN), and associative reward penalty (ARP), etc.

(b) **unsupervised learning**, such as; adaptive Grossberg (AG), analog adaptive resonance theory (ART), Kohonen's learning vector quantization (LVQ), competitive learning, fuzzy associative memory (FAM), and counterpropagation (CPN), etc.

In this study, the Kohonen's learning vector quantization based on unsupervised learning and backpropagation based on supervised learning were considered and studied. After a testing trial of these two methods, backpropagation seems to be the better of the two because it is more appropriate to use with training patterns which do not vary greatly in their size or position in the input array as in the case with the data

* URL: http://wwwipd.ira.uka.de/prechelt/FAQ/neural-net-faq-htmp, modified: 1995/04/20

of this study. As a result, the backpropagation approach is more suitable for classification tasks, whereas LVQ is more suitable for clustering problems.

In this Chapter, the backpropagation approach with the alternative preprocessing techniques are discussed for selecting input variables, designing outputs, data preprocessing, and data generating. The section proceeds with the creation of the data set by using original raw data through testing various types of preprocessing: absorbance pre-processing, PCA pre-processing, derivative pre-processing, and derivative encoded pre-processing.

6.6.1 Creating a data set

The goal of this step is to build a matrix formed from a series of input patterns each of which is a set of measured values, taken at one particular instance. The entire pattern is an input vector, and the individual values are vector components. Apart from this series of input vectors, supervised learning networks also require a target result for each input pattern. Building the input patterns generally requires matching choices from among many measurable variables. For instance in this study, the 316 points of each spectrum were represented as a input vector where the presence of each species was designed as a series of output vectors. The task here is to design a suitable match between the input and the output pair. This matching is a criterion for the success of the neural network fit. To do so, three questions by Patrick Henry Winston[66] were determined, these being prerequisites for the successful application:

'To determine if research work in artificial intelligence is successful, you should ask three questions:

- Is the task defined clearly?
- Is there an implemented procedure performing the defined task? If not, much difficulty may be lying "under a rug" somewhere.
- Is there a set of identificable regularities or constraints from which the implemeted precedure gets its power? If not, the procedure may be an 'ad hoc' toy, capable perhaps of superficially impressive performance on carefully selected examples, but incapable of deeply impressive performance and incapable of helping you to solve any other problem.'

Patrick Henry Winston, Artificial Intelligence[66]

Many different alternative techniques including selecting input variables, designing outputs, data pre-processing, and data generating were evaluated. All the techniques studied were implemented by using the Neural Desk version 2.11 neural networks package which provides four different backpropagation algorithms. These algorithms are the Standard Backpropagation, the Stochastic Backpropagation, the Quick propagation and the Weigend Weight Eliminator. These algorithms are implemented as Windows Dynamic Link Libraries (DLLs) and described in Appendix D.

The first of these trial evaluations used an original raw data as the matrix of input vectors, which is described in the next section.

6.6.2 Original Raw Data Trial

> *"The number of input nodes consequently equals the number of measured data values (vector components) presented to the network. In general, the training set must provide a representative sample of the data that the network will process in the finished application*".
>
> *D.M. Hammerstrom,* Neural network at work[74]

In the first trial, the training set consisted of 64 (4^3) patterns of intensity spectra. This is considered enough to represent the signals from three chemical species at 4 levels as absent, low, medium, and high. The 316 intensity values of each spectrum were measured in the range 190 - 820 nm with a 2 nm interval, as described in Chapter 5. These values of 64 patterns were scaled in the range 0 - 1 and fed into a neural network with 316 input nodes. The network had 12 output nodes representing the level of chemical components as absent, low, medium, and high for each of the three species in the matrix. For example, the output "1,0,0,0, 0,0,1,0, 0,0,0,1" represented the mixture of medium concentration of the hypochlorite ion and high concentration of the ammonium ion. The overall network topology used here consisted of 316 input nodes, 10 hidden nodes, and 12 output nodes, as shown in Fig. 6.7.

In this trial, the four different backpropagation algorithms cited above were tried and the learning rate, momentum, and the number of hidden nodes was varied in an attempt to optimise the fit. The result of this trial showed that for all the first-mentioned methods none of the network errors converged satisfactory.

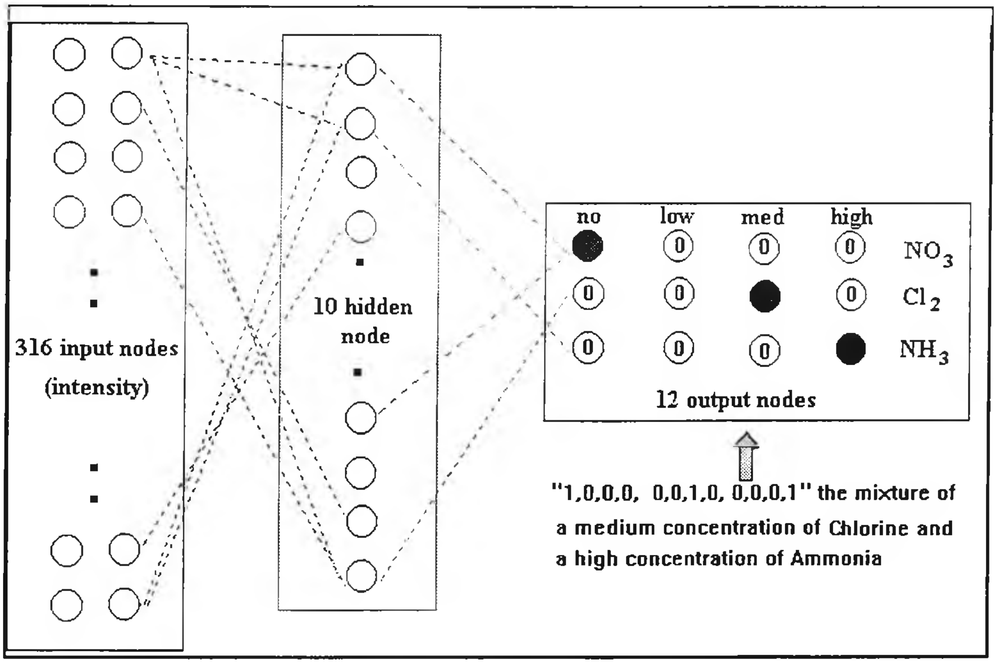

316 input nodes (intensity)
10 hidden node
no
low
med
high
0
NO_3
Cl_2
NH_3
12 output nodes
"1,0,0,0, 0,0,1,0, 0,0,0,1" the mixture of a medium concentration of Chlorine and a high concentration of Ammonia

Fig. 6.7 The network's topology.

One possible reason for the ineffective training in this trial is that the spectral information in the intensity spectrum correlates more with the emission of the lamp source than with the absorption due to the chemicals in the water. To accommodate this fact in the next subsequent trial, absorption spectral data, which shows a better correlation with the absorption species was used for the training and testing sets.

6.6.3 Absorbance pre-processing

In the present case, absorption spectra were used instead of intensity spectra which were too difficult to classify either by the use of neural network techniques or human expert eye observation. As can be seen from in Figs. 6.8a and 6.8b, it is easier to identify components in the mixture related to absorption spectra than intensity spectra.

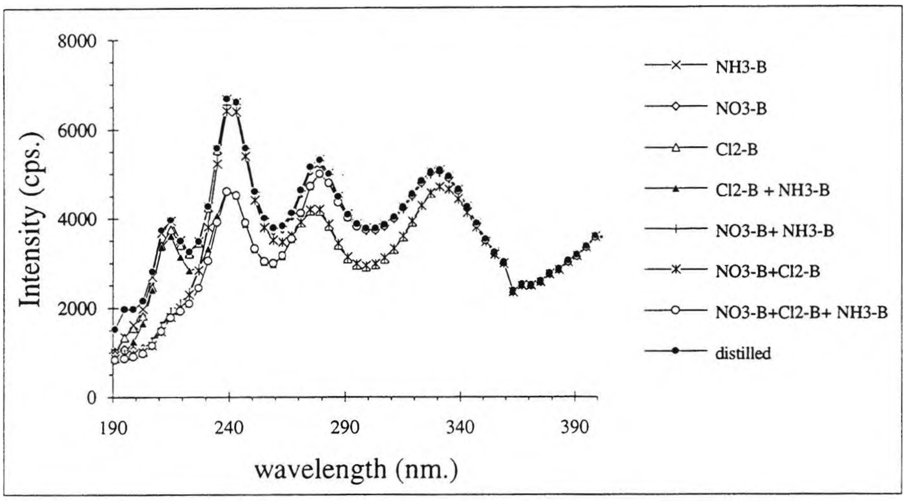


Fig. 6.8a Intensity spectra of mixtures.

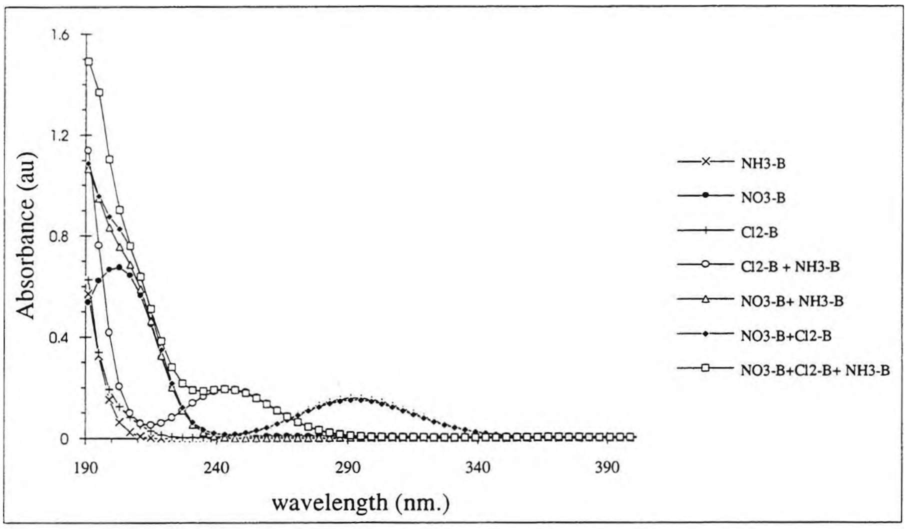


Fig. 6.8b Absorption spectra.

In this second trial, the number of inputs with absorbance spectral data was also reduced from 316 to 64 inputs in the UV range (190-318 nm) where the absorption band of the species of interest mostly occurs. The network topology used here had 64 input nodes, 16 hidden nodes, and 12 output nodes. The Stochastic Back-propagation algorithm was used with a 0.05 learning rate, and 0.5 momentum. This was 'successful' in training with a root of sum square error of 0.025 after 10000 epochs.

However, this trial was less successful with the testing data with a network performance accuracy of 28.22%. This poor performance may have been affected by the fact that the number of weights (64*16+16*12) is much larger than the number of data patterns, while the number of data samples should be, theoretically at least, equal to the number of weights. Therefore in the next trial, the size of the network was reduced by reducing the number of output nodes from 12 to 3.

In this trial, 3 output nodes represent the scale of concentration of the three species of interest. The network used here had 64 input nodes, 8 hidden nodes, and 3 output nodes with the stochastic backpropagation algorithm, a 0.05 learning rate and a 0.5 momentum. However, even with this significant decrease of the number of connection weights, the training error did not converge.

In a further trial to reduce the network size by choosing only the effective wavelength in the UV range around the absorption peaks of interest and reducing the output node, the average training error still did not converge. It was therefore concluded at this stage that backpropagation neural network is unable to classify the data set under these considerations. Therefore the PCA technique, which is commonly employed in data reduction was used for pre-processing, as will be described in the next section.

6.6.4 Pre-processing using PCA

Two examples in which PCA has been used successfully as preprocessing are discussed. Borggard and Thodberg[22], and Azoff[67] used PCA as pre-processing input data to optimize the network size that was able to decrease the number of patterns required for training without risk, whereas Peel *et al.*[23]designed the number of hidden nodes as the number of PCs and initialized weights between the input and hidden layer as the elements of the corresponding eigenvectors for a fast training procedure.

In the next trial, PCA was used to find the first 6 most significant components that accounted for 100% of the variance. The scores of these 6 components were used as the inputs to the neural network. The network size was optimized so that the resulting solution contained the minimum necessary connections to interpret the data, and the resulting network consisted of 6 input nodes, 3 hidden nodes, and 3 output nodes. As a result, the number of weights is reduced to 6*3+3*3=27 which is smaller than the number of data patterns. The concentration outputs were also scaled to keep within the range of 0-1 as the constraint of the neural network package.* Further, in this trial, out of the various algorithms the Stochastic Back-propagation gave the best performance with a 0.5 momentum and a 0.05 learning rate. However even here the training was unsuccessful in that it did not converge below 0.15 average error.

This failure in applying PCA to preprocessing may be because the PCA method reduces the dimensions of the data space without determining the strength of the correlation that exists between the input vector and the output vector. As a result of reducing the size of the network, the training still failed. As a consequence, instead of reducing the network size in the next trial, the number of the training pattern was

* It is important to ensure that all training and query data are scaled to a value between 0 and 1 before the Train or Query operation is commenced.[73].

increased. This was achieved by generating and adding random noise to the original data, as will be discussed in next section.

6.6.5 Simulating Data Patterns by Adding Random Noise

In general, the training set must provide a representative sample of the data that the network will be required to process in the finished application. A large training set reduces the risk of under sampling the underlying function. On the other hand, with a too small, noisy, or skewed training set, the network can learn perfectly but fail in testing with real data in the final application.

In practice, sufficiency of data depends on several factors: network size, testing needs, and input and target distribution. The size of the network matters most. A big network needs more training data than a small one. A rule of thumb suggests it is best to have five to ten training patterns for each weight.

There are many examples in the literature in which data are generated to increase the number of training patterns and amongst these are a study by Gemperline *et al*[21] who simulated a three component UV-Vis data matrix by generating data using three different non-linear effects, a study by Liu *et al*[29] who added 0.5%-5% random noise into NIR and Raman spectra whereas Ham *et al*[31] added Gaussian noise with 0.0008-0.0012 variance into infrared spectra to generate the testing data in order to study the effect of input noise on network performance. Further Glick and Hieftje[14] added noise to a training set and this was analogous to training with a larger data set. However Boger and Karpas[30] report succeeding with a very small training set compared with that theoretically required. The explanation is that the theoretical requirement assumes that no relation exists between inputs, which apparently is not the case for mobility spectra as used in their study.

Thus, in the next trial 1050 training patterns were prepared from the original 64 samples by adding 10% random noise. Also a base-line shift in the distilled water spectrum in the testing set, as shown in Fig. 6.9, was found, which would affect the comparison of absorbance between the training data and the testing data. Therefore, in the next trial a different input selection was used, instead of absorbance spectral data which is related to the distilled water spectra, as used in the previous trials.

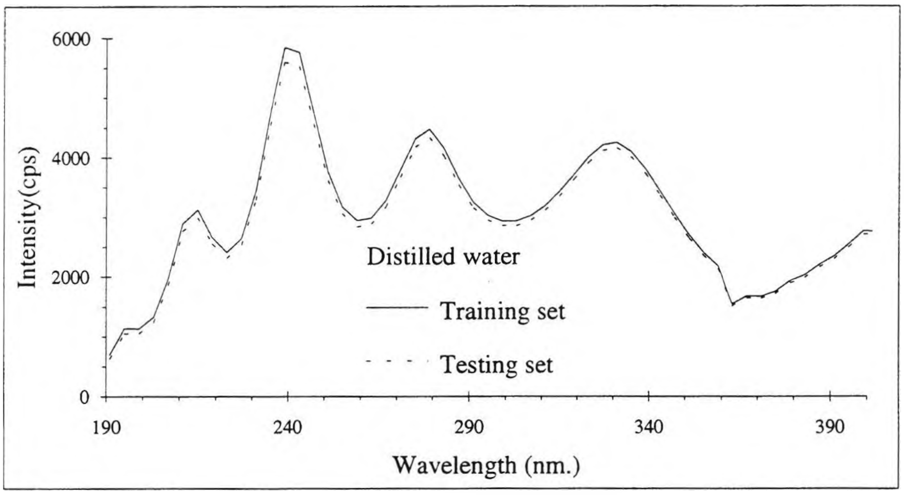


Fig. 6.9 The intensity spectra of distilled water in training set and testing set.

This is done by ratioing the raw data with the intensity at 620 nm, which PCA analysis showed was the least significant wavelength in the spectrum. The 73 inputs were used here consisting of 53 inputs in the UV range and 20 inputs in the visible range. The bias of inputs to the UV reflects the greater significant of this part of the spectrum. The network used here had 3 output nodes which were used to determine whether the nitrate ion, the hypochlorite ion, or the ammonium ion were present. For example, "0, 1, 1" represented the mixture of the hypochlorite ion and the ammonium ion. If the classification were successful, the determination of the concentration would be carried out in the next step.

For all the changes introduced here to improve the possibility of training, i.e. using a larger training set, new representative data and reduced requirements on the output nodes, the network error did not converge in training using any of the algorithms discussed, i.e. Stochastic Backpropagation with a 0.1 learning rate and a 0.9 momentum, Standard backpropagation with a 0.1 learning rate, and Weigend Weight Eliminator with a 0.1 learning rate, a 0.9 momentum. As this result, it can be implied that the correlation of inputs and outputs, as designed, may be too weak which is too difficult for training the network.

This failure in network training and testing with both intensity and absorbance spectral data mean that other pre-processing techniques must be received that are easier for the networks to learn from. In our next attempt, the derivative spectra are used which can help to solve the problem of badly defined absorption peaks that may emerge by the overlapping due to the presence of the mixing of species. Therefore a completely different approach to preprocessing involving a derivative of the spectra was investigated next.

6.6.6 Derivative Preprocessing

Every method reported in the literature review as being successful was tried, but most were found to give different results when applied to the complex examples of this study. These complexities are categorized as:

First, the concentration range of each species is very wide, and the mixing of a high concentration of two species causes very high absorption, resulting in the intensity being lost in the noise signal,

Second, the peak shifts between training and testing sets were measured under different conditions due to the drift in the experimental apparatus,

Third, imprecision in the chemical preparation, and

Finally, chemical reaction between components within the same mixture.

However, the existence of one species or another in the analyte depends less on the value of the absorption, which reflects the concentration, than in the actual wavelength of the absorbance. Since absorption peaks are more easily located in the derivative spectra, using differential spectrometry as a preprocessing stage may help subsequent classification by the use of neural networks. In this context, the two following statements are relevant.

> *"The use of even derivatives (2nd and 4th) is preferable as the best compromise between resolution and sensitivity. Therefore, in the algorithm developed here for resolution of the overlapping absorption bands, their number is determined by derivative spectroscopy."*
>
> *Liudmil Antonov and Stefan Stoyanov* [75]

"It is generally accepted that, for overlapped spectral components, the results are not as accurate as those obtained by curve-fitting of the orignal spectrum. Nevertheless, when peaks are extremely broad and when overlap is severe, quantitation has been routinely accomplished by multivariable calibration methods which use second-derivative transformations as a pre-processing step."

P. R. Bevington [76]

Fig. 6.10 shows the implementation of derivative spectrometry. This figure illustrates an absorption, 1st and 2nd derivative spectrum of mono-chloromine at 17.86 mg/l. It is clear that the first derivative plot can be used to fix accurately the wavelength of the maximum absorption, λ_{max}, (arrow point) where the plot intercepts an axis. The wavelength of the maximum absorption, λ_{max}, can be also fixed by the second derivative plot because it has a central peak which is sharper than the original band.

It is clear that resolution is improved with the use of the higher derivative, and this offers the possibility of separating any two absorption bands which may in fact merge in the zero-order spectrum, as shown in Fig. 6.11. This figure shows the absorbance spectra of a mixture of the nitrate ion at 3.88 mg/l and monochloramine at 17.86 mg/l, in which it is difficult to identify the nitrate ion peak, while the two peaks are clearly seperated by the 2nd derivative spectrum as shown in Fig. 6.11c. However the evaluation of derivative spectrum may give rise to the disadvantage of progressively decreasing of the signal-to-noise ratio at higher derivative order, and therefore only the 1st and 2nd derivative were tried in this example. However as many sharp absorption peaks are not expected, the noise can be minimized by use of a moving average to smooth the spectra.

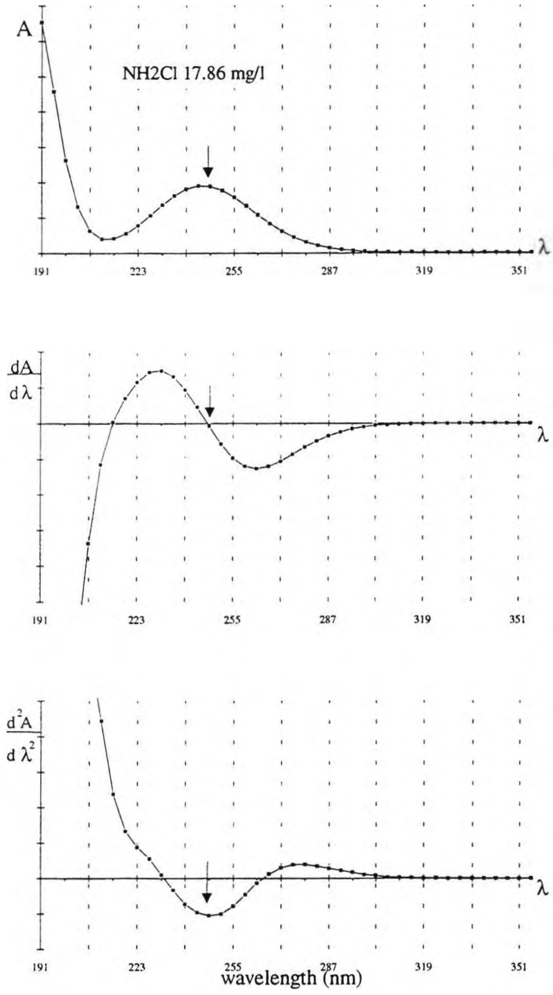


Fig. 6.10 The plot of $A, \frac{dA}{d\lambda}, \frac{d^2A}{d\lambda^2}$ of NH$_2$Cl 17.86 mg/l

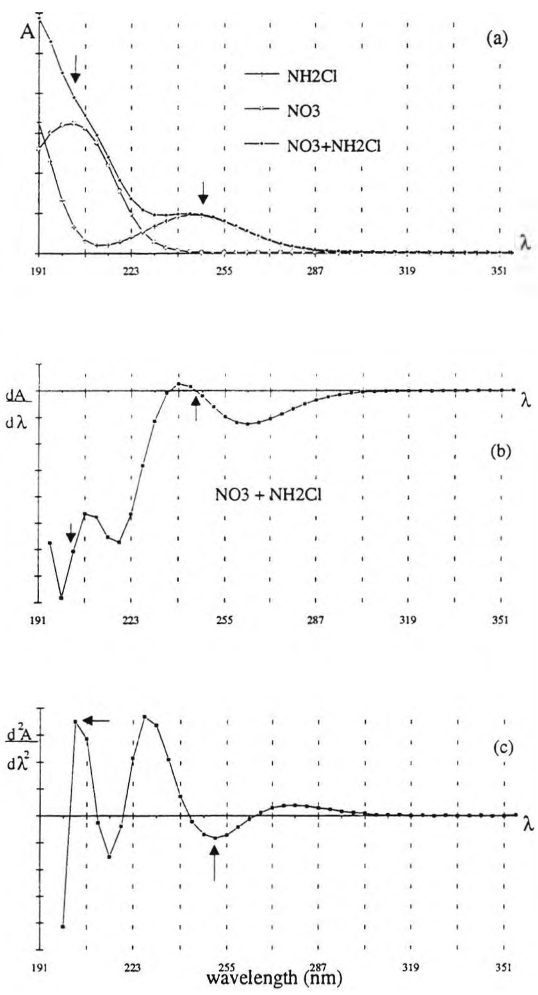


Fig. 6.11 The plot of $A, \frac{dA}{d\lambda}, \frac{d^2A}{d\lambda^2}$ of NH$_2$Cl 17.86 mg/l + NO_3^+ 3.88 mg/l

In this trial, 53 values of original intensity spectra in the range 190-400 nm and 20 values of intensity spectra in range 400-800 nm were averaged, then transformed to absorption spectra with 10% random noise added to generate 1056 data patterns. These 1056 absorption spectra were converted to the first derivative with respect to wavelength ($dA/d\lambda$). The network topology was 73 inputs of $dA/d\lambda$, and 3 outputs, which were just to determine whether the nitrate ion, hypochlorite ion, or ammonium ion were present. Extensive effort was made to find the neural network solution with an acceptable level of minimized error. To optimize the training error, the following procedures were used:

1) varying the number of hidden nodes between 5 - 8 nodes,
2) varying the algorithms used including Standard BP, Stochastic BP, Quick Prop, and Weigend Weight Eliminator,
3) varying the initial learning parameters, such as learning rate between 0.1 - 0.5 and momentum between 0.7 - 0.9,
4) adjusting the learning parameters during training observation by decreasing the learning rate during the fluctuation of the training error and increasing the momentum parameter during the smoothing of training error in order to escape from local minima.

The above procedures involved many trial optimizations of durations of over 20 hours. The smallest training error was given by the use of Stochastic BP with a 0.1 learning rate and a 0.9 momentum, but it was still unsuccessful in converging the training error.

Since the 1st derivative spectrum failed, the 2nd derivative was tried. The 1056 values of 1st derivative spectrum were converted to a 2nd derivative spectrum and then trained with the same topology and algorithm of previous networks. This network also gave a nonconverging training result.

With the further failure of these differential spectra trials, in which classification by eye was easy, a new perspective was required. One idea was that these failures may have been caused by the inclusion of many uncorrelated inputs in the data set which neural networks have to try to classify as being in the noise. For instance, the 20 inputs in the visible range are not necessary to classify the chemical components of this study. Also there are other ranges where the derivative value may be in the noise level because of decreases in the absolute signal level. Therefore a more selective approach was later taken and based on a knowledge of where the expected species absorb and only spectral data in the wavelength range around the expected species were used. This also optimized the size of the network employing fewer inputs, as described in the next section.

6.6.7 Knowledge-based approach to second derivative preprocessing

In general in absorption spectroscopy, the presence of a species is determined by the absorption peak position while the concentration is determined by the absorption values. For example, Fig. 6.12 shows the absorption spectra of the hypochlorite ion at different concentrations, whose absorption peak is at 290 nm

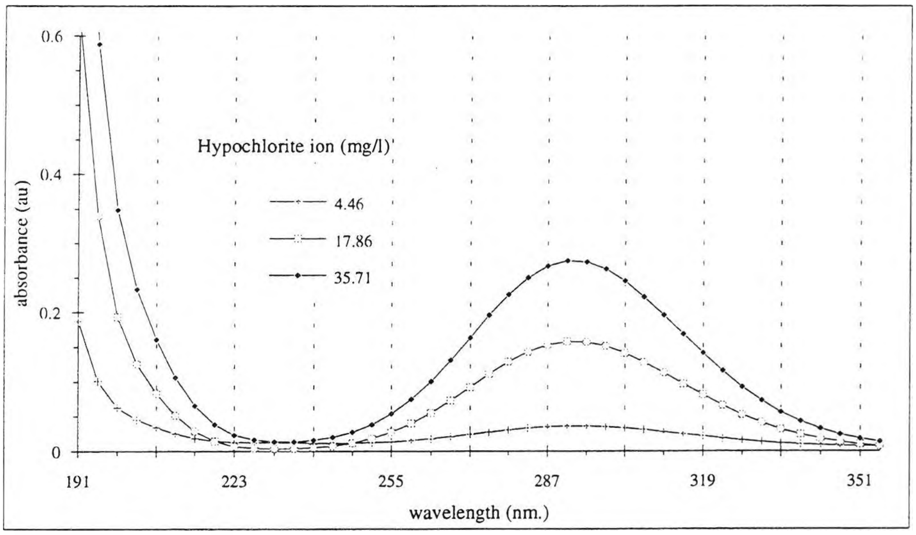


Fig. 6.12 The absorption spectra of OCl^- at 4.46, 17.86 and 35.71 mg/l

In general for Gaussian shaped peaks, classification requires only the shape information where the absorption peak is present. This (λ_{max}) can be found by the intersection on the x-axis of $\frac{dA}{d\lambda}$ plot. However, the spectra of this study were not of a perfect shape: for instance monochloramine has an absorption peak between 220-300 nm and also between 190-220 nm; the hypochlorite ion has absorption peaks between 240-340 nm and 190-240 nm; the ammonium ion has an absorption between 190-210 nm; and the nitrate ion has an absorption peak between 190-240 nm, as shown in Fig. 6.13. Moreover the mixing of the two components makes it more difficult to identify the absorption peaks of the individual components. Theoretically, where a spectrum can be described by an *n* th-order polynomial, interference between peaks will be eliminated in the (*n*+1) derivative, and so a higher order derivative will improve the resolution. In the current spectra, the ammonium ion has not such a peak and thus the 3rd derivative will eliminate the effect of ammonium as a noisy spectrum. With this elimination, the next trial could then use the second-derivative to represent shape information.

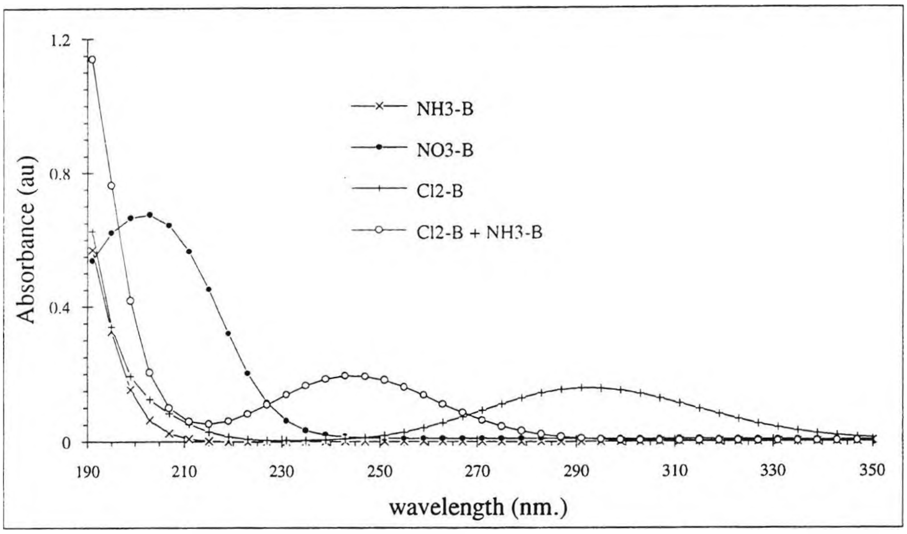


Fig. 6.13 The absorption spectra of mixtures.

Boger and Karpas[30] reported that "*using neural networks did not need detailed knowledge of identified peaks.*" However with the direct application of neural networks, the network has to adjust the connection weight to fit the whole input data set including data at the noise level. For instance, inputs in the visible range are not necessary to classify the chemical components involved in this study and other ranges should be excluded because the derivative value may be in the noise level due to the decreasing signal to noise ratio. By the use of the prior knowledge of some of the parameters involved, such as peak shape, number of components, and peak positions and widths, there may be a further increase in the effectiveness of the fitting procedure. Here, the available knowledge of the individual peaks was applied to select inputs from the 2nd derivative spectra to optimize the training error and network size. The 21 inputs which were selected in the intervals, where each species shows an absorption which occupies three intervals of increasing, convex, and decreasing of absorption spectra. This results in the 21 inputs of the 2nd derivative value, as shown in Fig. 6.14.

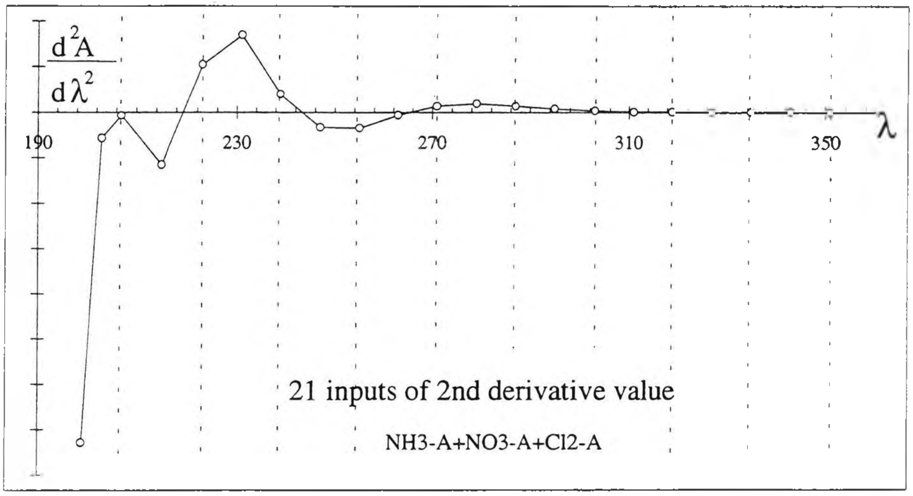


Fig. 6.14 The 21 inputs of 2nd derivative spectra.

Again, to obtain sufficient training sample sets, stray light was added to the raw data in the range 0-5% in intervals of 0.5%, using the formula below. This generated 704 training patterns represented by $A_{i\lambda}$(generate) through the addition of stray light, as shown in Eq. 6.12.

$$A_{i\lambda(\,generate)} = A_{i\lambda} + \log(1+\frac{E}{10^{A_{i\lambda}}}) \qquad ....(6.12)$$

where $A_{i\lambda}$ is the absorption of the $i^{\underline{th}}$ component at the wavelength λ, and E is the fraction of the stray light added.

This trial used stochastic back propagation with a 0.05 learning rate and 0.9 momentum coefficient, a network topology of 21 input nodes, 5 hidden nodes, and 3 output nodes to represent the presence or absence of the three chemical species. The training and testing errors are shown in Fig. 6.15 and 6.16 respectively. Fig. 6.15 shows that the training error was very slow in converging and still a high root of the sum of the square errors remained after 2000 epochs of training. However the testing result was very much improved from the previous trials at about 15-25% error, as shown in Fig. 6.16.

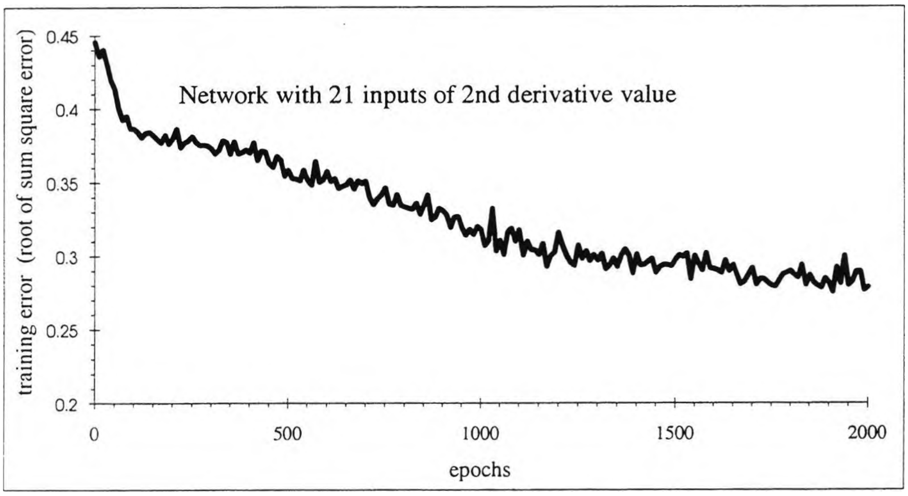


Fig. 6.15 The training error of 2nd derivative input network.

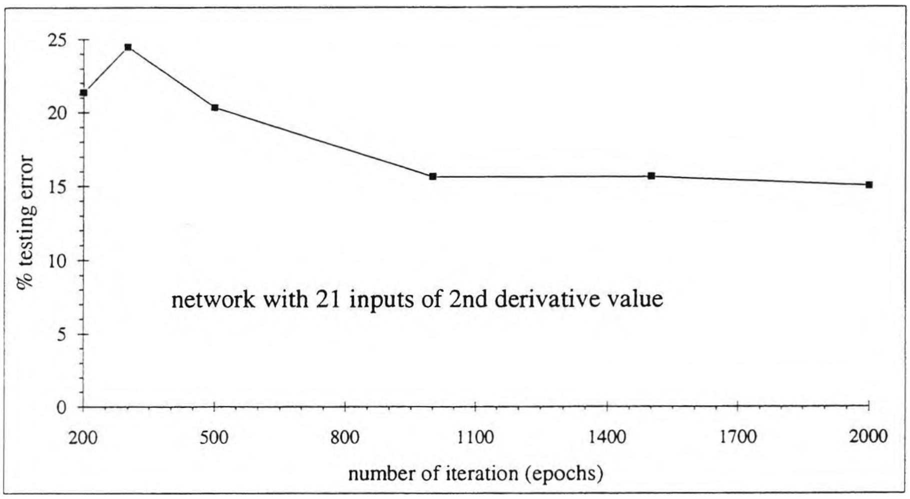


Fig. 6.16 The testing error of 2nd derivative input network.

This training procedure consumed large amounts of time, in taking more than 20 hours for 2000 epochs and resulted in quite a high training error. However the result obtained was very encouraging. Therefore the next trial was arrived at making it is easier for neural networks to learn by encoding these descriptive values to digit numbers, which can represent the shape information.

6.6.8 Binary encoded second derivative preprocessing

The idea behind binary encoding the 2nd derivative spectra is to obtain a code that represents the shape of the spectral data in a way that is as independent as possible from the intensity of the data. The actual procedure is described in detail later, but briefly, it consisted of binary encoding segments of the second derivative of the absorption spectra according to the state of the slope, i.e. of the third derivative. Another important point of the scheme is that only relevant data were used, that is data from around the absorption peaks of the expected species.

In detail, the absorption spectra, and 1st and 2nd derivative spectra in the present experimental data, for example from the hypochlorite ion alone, hypochlorite ion plus monochloramine, and nitrate ion plus monochloramine, as shown in Fig. 6.17, were used. The binary encoding scheme consists of firstly segmenting the 2nd derivative spectra between the wavelength range 190-350 nm which is divided into 10 segments as shown in the Fig. 6.17c., with the segment between 206 and 224 divided into two segments, giving a total of 11 segments. The reason for bisecting this segment is that there is a minimum in this region that would otherwise be missed. Secondly, work is done in encoding the slope in each segment with a 2 bit binary code of 01 for decreasing, 10 for increasing, 11 for convex, and 00 for unchanged, as illustrated in the Fig. 6.17c. and thirdly, using the resulting 22 bit code as an input to a classifying backpropagation neural network.

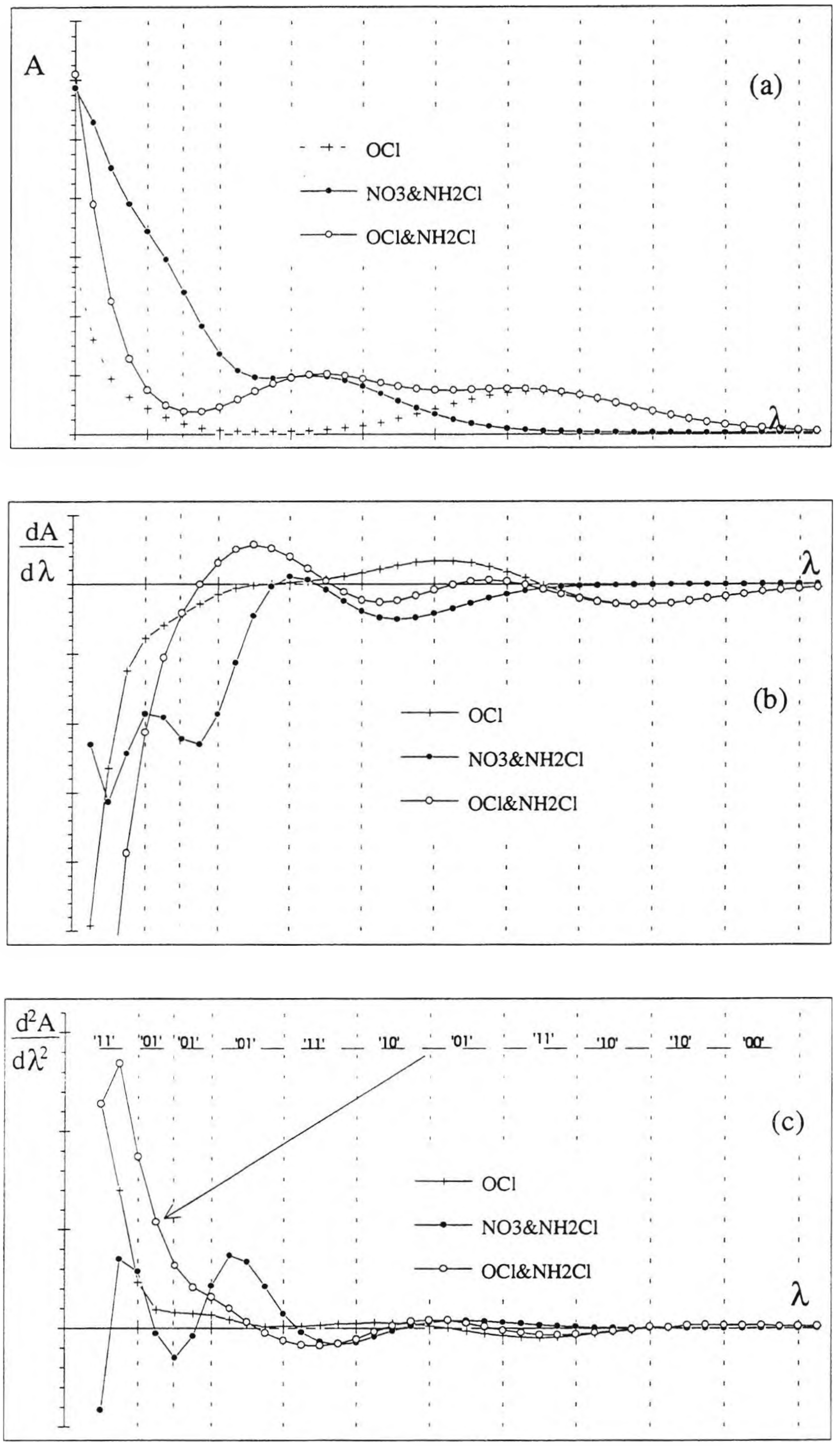


Fig. 6.17 Plot of A, $\frac{dA}{d\lambda}$, $\frac{d^2A}{d\lambda^2}$ of *OCl^-*, *NO_3^+& NH_2Cl*, and *OCl^-& NH_2Cl* and encoding of *OCl^- & NH_2Cl.*

The topology of the network used here is 22 input nodes, 5 hidden nodes, and 3 output nodes. The three outputs determine whether the nitrate ion, hypochlorite ion, or ammonium ion were present. The use of the classification network is compared with previous preprocessing methods, as summarized in Table 6.2.

Table 6.2 The topology of networks in three different preprocessing methods

Stochastic back-propagation: *learning rate = 0.1, alpha = 0.9*	*Network Topology*		
Pre-processing	*inputs*	*hiddens*	*outputs*
Score of the first 6 principal components of absorption spectra	6	3	3
2nd derivative values	21	5	3
Encode the shape of 2nd derivative spectra	22	5	3

As can be seen in Fig. 6.18 and Fig. 6.19, the classification from this scheme is much more accurate than that from previous attempts and the training error is seen to have converged to a small value very quickly.

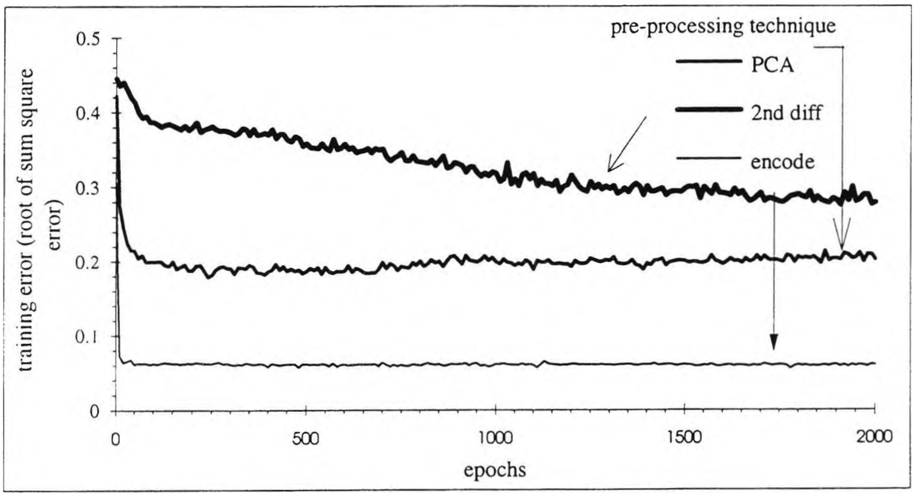


Fig. 6.18 The training error obtained using three different preprocessing techniques.

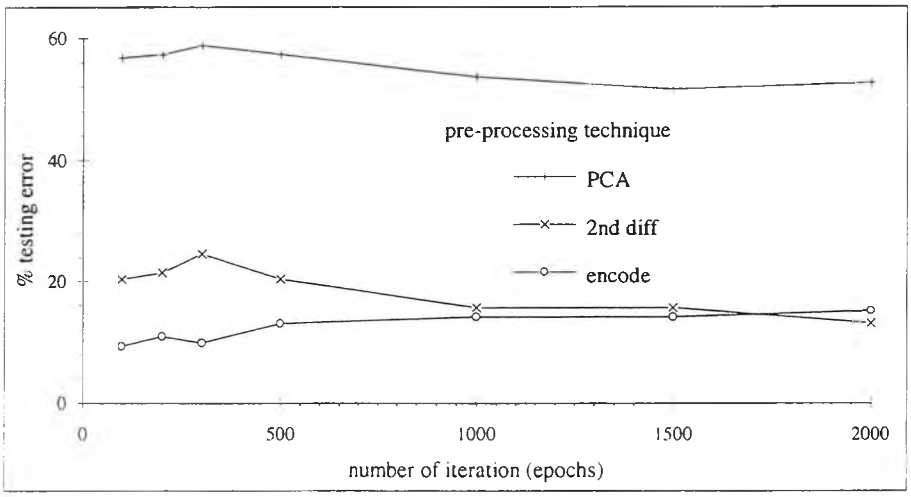


Fig. 6.19 The testing error of three different techniques.

This resulted in a 93.75% prediction confidence overall for the classification. However an error of 6.25% is obtained for the case of a mixture of the ammonium ion and nitrate ion which is mis-predicted as a mixture of ammonium ion, nitrate ion and hypochlorite ion. However, as will be shown in the next section, when this classification was followed up by the third stage of estimation, the mis-identified chlorine is predicted as occurring at a low, and insignificant level. Thus, overall, the two stages tends to cancel out the error.

6.6.8 Estimation of Concentration

Following on from the binary encoded second derivative preprocessing, the scheme for estimation of concentration also uses a knowledge of components expected in the spectra. Here one of seven possible networks which predict concentration is chosen, depending on the output of the classification network. The seven networks cover all the possibilities of the use of the following:

1. hypochlorite ion,
2. nitrate ion,
3. hypochlorite ion and nitrate ion,
4. ammonium ion,
5. hypochlorite ion and ammonium ion,
6. nitrate ion and ammonium ion, and
7. hypochlorite ion, nitrate ion and ammonium ion.

The results obtained are described in the next section.

In this scheme, the absorption peak shape data for the various components; nitrate at 210 nm, ammonium ion at 180 nm, hypochlorite at 290 nm and monochloramine at 245 nm are used to determine the input variables for the seven BPNNs used for estimating the concentration. The inputs used are the 7 values of absorbance centred around the peak region of each species as depicted in Fig. 6.20 and tabulated in Table 6.3.

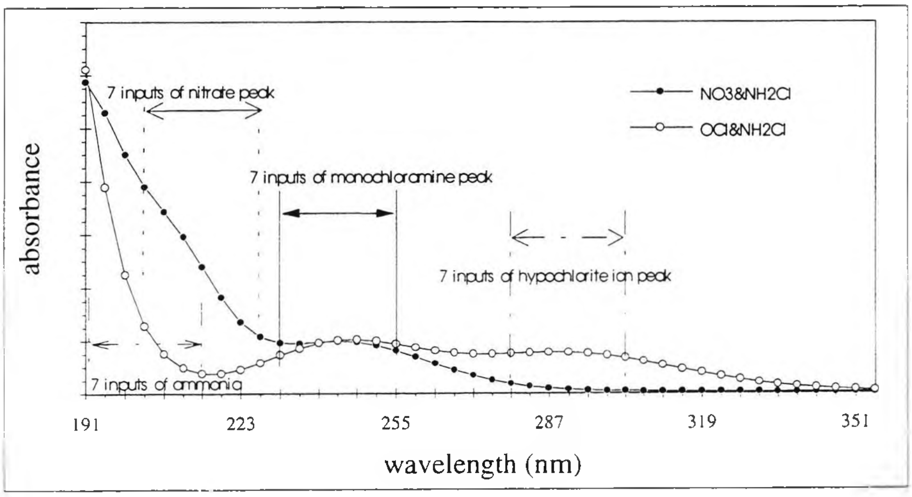


Fig. 6.20 Inputs of the concentration networks.

Table 6.3 Selecting of inputs for each networks.

Network	INPUTS : absorption values measured over wavelength range
1. Chlorine	279-303 nm (7 inputs)
2. Nitrate	203-227 nm (7 inputs)
3. Chlorine and Nitrate	279-303 & 203-227 (14 inputs)
4. Ammonia	191-219 nm (7 inputs)
5. Chlorine and Ammonia	191-219 & 231-255 & 279-303 nm (21inputs)
6. Nitrate and Ammonia	191-227 nm(9 inputs)
7. Chlorine, Nitrate and Ammonia	191-227 & 231-255 & 279-303 nm (23inputs)

Training and testing data for these seven networks were generated, using extinction coefficient data for each of the expected components and varying the concentration of each species over the ranges shown in Table 6.4. The extinction coefficients data sets required for this were obtained by a partial least squares analysis of the absorption spectra using the combined original raw training and testing data sets.

Table 6.4 Variety of the concentration of each species used in generating training patterns.

Concentration range (mg/l)			
Network	Chlorine	Nitrate	Ammonia
1. Chlorine	4.5 - 40.0	------	------
2. Nitrate	------	1.2 - 8.0	------
3. Chlorine and Nitrate	4.0 - 40.0	1.2 - 8.0	------
4. Ammonia	------	------	4.5 - 40.0
5. Chlorine and Ammonia	4.0 - 40.0	------	4.0 - 38.0
6. Nitrate and Ammonia	------	1.2 - 8.0	4.0 - 40.0
7. Chlorine, Nitrate and Ammonia	5.0 - 40.0	1.0 - 8.0	5.0 - 40.0

Information from this new training data set was used with one of the following network topologies, depending on the input-output of the classification network. The various networks were, 3 hidden nodes for 7 inputs for the cases with 1 outputs node, 4 hidden nodes for 14 inputs for the cases with 2 outputs, and 5 hidden nodes for 21 inputs for the cases with 3 outputs. The algorithm used here was the Stochastic BP with a 0.1 learning rate and 0.9 momentum. It is worth noting that these estimation networks are linear since they were trained using data from a linear model of absorbance.

Testing data for the estimating networks were generated by selecting the values of absorption halfway between the values of absorption used for the training set and adding 5% stray light, as shown in Eq. (6.3).

$$A_{i\lambda}(\,generate) = A_{i\lambda} + \log\left(1 + \frac{0.05}{10^{A_{i\lambda}}}\right) \qquad .....(6.3)$$

The graphs of training and testing error for the ammonium ion and hypochlorite ion network in Fig. 6.21 are typical of the results obtained for all the estimation networks. As can be seen from Fig. 6.21, the network converges very quickly from an error of 0.189 at 1st epoch to less than 0.003 after only 100 epochs. This figure also shows that the testing error also reduced to a low level of 0.015 after 500 training epochs.

The overall high accuracy of the estimating networks is shown in Table 6.5. This figure tabulates the number of training patterns and the resulting error for the networks.

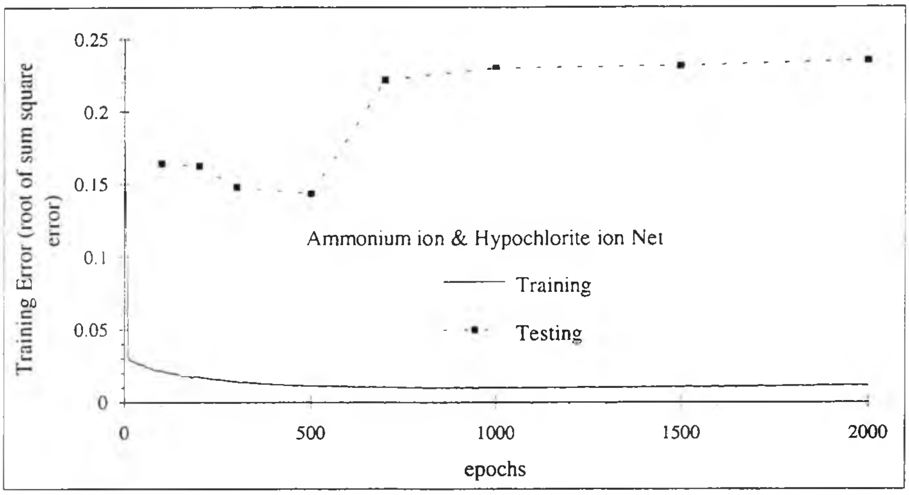


Fig. 6.21 Training and testing error for the ammonium ion and hypochlorite ion estimation network.

Table 6.5 Network Topology and Performance.

ALGORITHM	Stochastic BP with 0.1 learning rate, and 0.9 momentum		
Network	pattern	hidden	Result Error
1. Hypochlorite ion	356	3	0.03 %
2. Nitrate ion	137	3	0.05 %
3. Hypochlorite ion and Nitrate ion	350	4	0.18 % & 0.20 %
4. Ammonium ion	356	3	0.04 %
5. Hypochlorite ion and Ammonium ion	350	4	0.72 % & 0.17 %
6. Nitrate ion and Ammonium ion	350	4	0.68 % & 0.56 %
7. Hypochlorite ion, Nitrate ion and Ammonium ion	512	5	0.13 % & 1.72 % & 1.03%

Finally, to check that the combined performance of the classification and estimation schemes performed satisfactorily, species and concentrations predicted by the scheme were used as an input to a model, based on absorption data information, to generate spectra that could be compared statistically with the original sample spectra, and hence determine whether the overall scheme was self-consistent.

The result of this is shown in Fig. 6.22 which illustrates the satisfactory nature of the comparison of the raw data and those predicted for the ammonium ion and hypochlorite ion.

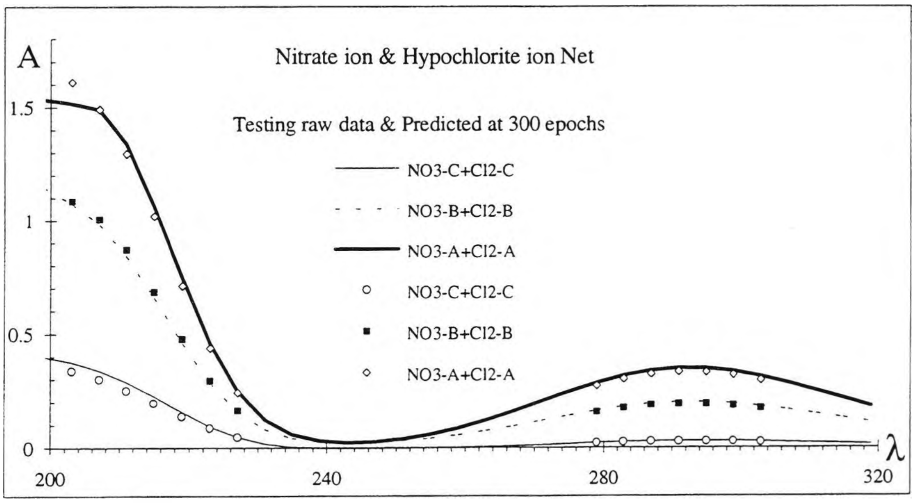


Fig. 6.22 Plot of raw vs predicted data of hypochlorite ion and ammonium ion network

6.7 Conclusions

In laboratory-based trials, PCA and 2nd derivative analysis of UV-Vis absorption spectra in preprocessing spectra before BPNN classification are found to be ineffective as means to classify the composition of chemical species in water samples from UV-Vis spectroscopic data. This follows from earlier work in which various neural network algorithms including BPNN were evaluated for classification and estimation and also found to be ineffective. It can be concluded here that such self-learning approaches to classification may in general be unsuccessful and that choosing a few strong correlation inputs based on background knowledge of the expected species and their absorption spectra may give a better performance than the methods above which use the whole data set.

Subsequently, a more knowledge-based approach has been formulated which restricts itself to determining the presence and concentration of a range of expected species. This scheme involves a three stage process. In the first stage shape information is derived by binary encoding segments of the second derivative of the absorption spectra according to their shape. The rationale of this stage is to reduce the spectral information to shape-sensitive factors. This is found significantly to ease the classification of the spectra by a second stage of BPNN analysis. For the estimation of the concentraton of the species, absorption data for the expected species are used to train a second stage of BPNN. Segmentation of the spectra and selection of relevant inputs for the second stage BPNN are determined from the absorption data for the expected species, to give the best segmentation pattern and the minimum number of network inputs.

This overall scheme is shown in the study to have a good predictive ability when presented with spectra from a mixture of contaminants. The combination of networks with the observation of only the portions of the spectral data that relate to the expected chemical components performs better than one network when used in estimating all the chemical components simultaneously.

The two-step approach taken to classification and then estimation yields a better result than a one step approach. The first-step network specifies which species are likely to occur and the second-step network can then be used to focus on a few inputs that strongly correlate with the presence of the expected species. Also the second-step provides a filter that compensates for the classification of species at low concentration levels or the incorrect identification of species due to low level signals in the presence of noise, as shown in Fig. 6.23.

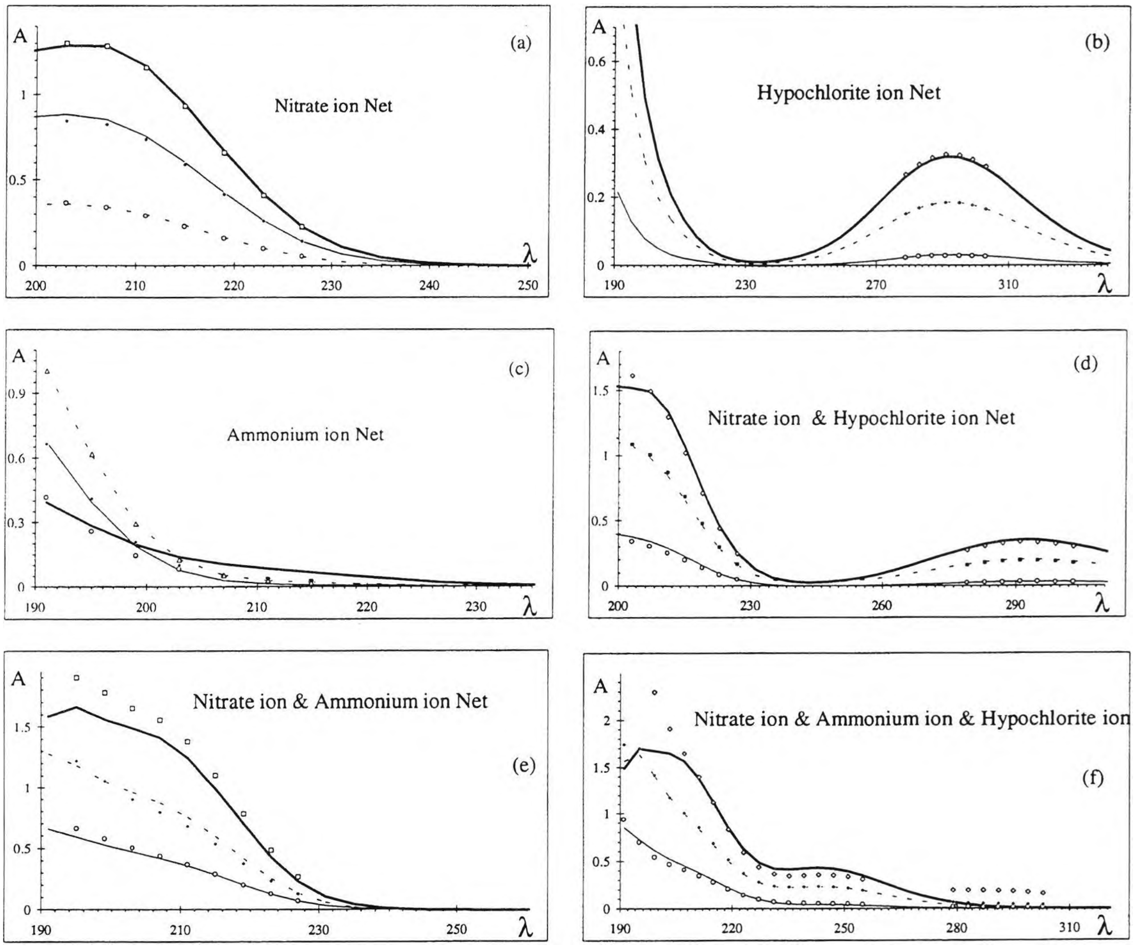


Fig. 6.23a-f Sample absorption spectra compare with spectra predicted by the concentration estimation network for the six other combinations of species at three different concentration levels

As neural networks imitate function on the basis of an optimised mapping of inputs to outputs and this is reflected in the following points that should be considered in the preprocessing using the binary encoded of the $2^{\underline{nd}}$ derivative:

1) The correlation between inputs and outputs must be mapping functions which reduce the number of independent variables.
2) The segments of the $2^{\underline{nd}}$ derivative spectrum must be carefully divided to reflect knowledge of the absorption spectra of the chemical species of interest.

Otherwise valuable information may be lost and the mapping may not reflect the change in species.

3) Where spectral features shift in wavelength in the spectra (due to insufficient light of chemical interferences) the choice of the 2nd derivative segments should be modified to reflect this.
4) As it stands, if unknown species are in the spectra, they will be classified at a characteristics of 'undeterminable data.'

Although the above criteria will yield estimates for most applications, there is another criteria the acceptability of a lack of precision in the estimates. As neural networks are like statistics in their effectiveness is in applications that lack other solutions and tolerate imprecision. They are suitable for tasks that have few obvious rules, deal with imperfect data, or optimise many constraints simultaneously. For example, controlling an industrial waste process can be a good task for a neural network, since often rules are difficult to define, historical data are plentiful but noisy, and perfect numerical accuracy is unnecessary.

Many advantages of the preprocessing scheme developed here could be summarized as the following:

(1) As far as the author is aware, this is the first reported use of neural network to classify and estimate aqueous borne species with the occurrence of chemical interactions between the components using the UV-Vis absorption spectra This reflects the difficulty of analyzing affected from the non-linear approach due to the modification of the spectra features caused by the chemical interaction and overlapping features in UV-Vis spectra.

(2) The success obtained when neural network is combined with knowledge based pre-processing in the form of the 2nd derivative binary encoding of the spectral results in a wide range of applications not otherwise possible in monitoring waters with various expected constituents. This scheme could be applied in the analysis of unresolved spectral features in UV-Vis absorption and fluorescence spectrometry. A possible example for the application of these methods is in the better determination of the total organic carbon (TOC) concentration in water, which absorbs UV light at 254 nm[5] from complex background spectra including nitrate ions. Moreover, it can be applied to colorimetric measurement where it may help discern spectral features, for example, in the colour change of dyes or fluorophores used in detecting the presence of important metals such as Hg, Cd and Cr(VI).

(3) Significant possibilities exist for future development of this processing scheme to so as it becomes more automatic and capable of determining the existence of unknown species.

(4) The difficulties of using neural network in process control loops are reduced with the three-stage scheme developed here. This is because the three-stage scheme is a 'well structured' and 'more understandable' approach to analysing spectra than the 'black box' approach of using neural network alone. In particular, it is probable that the classification scheme based on binary encoded 2nd derivative spectra is more verifiable and may lead to the use in process control schemes where the straight use of neural network would not be possible.

(5) The scheme is capable of being easily extended to the measurements of other parameters within the same measurement scheme. For instance the measurement of temperature, pH and other transient information from dynamic measurements can be incorporated within the scheme.

(6) The scheme provides a systematic approach to the application of UV-Vis absorption spectroscopy in water analysis. Although the implementation requires knowledge of the expected chemistry in the application, the implementation is expected to be relatively rapid and easily adapted to changing requirements within an overall instrument design.

CHAPTER 7

Conclusion

"The problems of the world cannot possibly be solved by skeptics or cynics whose horizons are limited by the obvious realities. We need men who can dream of things that never were."

John. F. Kennedy, The Age of Inteligence Machine[1]

7.0 Review of Work Carried Out

In order to fulfill the need that has been identified for a generic approach to the design of intelligent sensor systems in the water and environmental monitoring area, the work described in this thesis reports an investigation of the techniques developed in the analysis of the absorption of light in the UV-Vis part of spectrum, for the monitoring of the concentrations of chemical species dissolved in the effluent outflow analyzed from the industrial sites discussed.

This thesis has concerned itself with resolving a major problem in applying UV-Vis spectrometry to the on-line analysis of water, namely the lack of spectral resolution caused by the presence of wide absorption peaks, which are typically 20 nm, of the important and relevant species absorbing in the UV-Vis region.

In the thesis the limitations of the applicability of UV-Vis spectroscopy are closely defined and an approach developed based on the use of artificial intelligent techniques whereby UV-Vis spectroscopy can be applied within an integrated software scheme, that can be easily adapted to different applications, is discussed.

To achieve this two main problems have had to be addressed. There were firstly, the inability to determine the exact nature of the polluting constituents due to the unpredicable nature of the source water of interest. Secondly, the problem of classification and estimation of the species in the analyte required to be advanced. These two problems, especially the latter, are not trivial and a significant effort has been required and reported upon to find what would be a practical solutions to them.

The end results and principal output of the work is an intelligent software monitoring system for use in association with the central of on-line UV-Vis spectrometers, used as the essential monitoring tools, together with a development of the necessary procedures for implementing the monitoring of specific species of interest in aqueous-based industrial process and outflows. The software developed specifically for the purpose during the work will also be applicable in future research within sensor systems for the determination of specific species in the analysis of less specified open water, i.e., specifically in the measurement of chlorine, nitrate, ammonia, organics, and in COD estimation.

The work of the thesis to achieve the above has been divided into three parts, which were discussed, i.e.:

First, development of the "front end" sensing system: This consisted of developing novel software to control a UV-Vis spectrometer, software for data acquisition, and software to provide a basic analysis of the data, such as a spectrum plot, average, standard deviation, and an assessment of simple errors. This completed system was automatic and remotely controlled by a readily available computer (IBM-PC) using a modem for a telemetry.

Second, the conducting of feasibility studies of the newest appropriate data analysis methods: Here two data analysis methods frequently employed in analytical chemistry, PCA and artificial neural networks, were studied and compared in their ability to classify and estimate factors in the data available, for use in the system designed.

Third, implementation of the above in waste water monitoring: A GUI-software that integrates the above functions was designed and implemented for a host machine to develop an integrated automated monitoring system. The approach taken in this part has used neural networks for data analysis which in the second task was shown to have distinct advantages over the PCA technique for this application.

7.1 Discussion of the progress of the work

In first part of work, after construction the front end sensing machine, it was operated and tested continually over a trial two week period with no operational problems. The controlling software and communication package employed were also tested in a laboratory-based trial involving the measurement of tap water samples at City University without any problems being experienced. However when the system was used at industrial sites, several more serious problems arose because of the more adverse operating conditions. These conditions not only revealed limitations in the machine itself but in other actual systems involved in the process including sample conditions, public communications and the harshness of the industrial environment. After these initial trials, the software for the remote system was redesigned to work without intensive human operation. The resulting system, following that redesign, has shown itself capable of detecting most possible error conditions and dealing with them, including a system halt which occurred where the software was shown to have been successful in having coped with an automatic rebooting of the computer system. This upgrade system was then installed and used to collect a large amount of data in an on-line measuring industrial outflow.

Unfortunately, it was not possible to classify the data in these measurments in the way anticipated, even with over two months of data received. In order to classify these data, fully, several other steps would have been necessary. It is clear that, since it is impossible to control the phenomena involved in an industrial process to generate deliberately a series of pollution incidents, therefore the "polluted samples" have had to be prepared in the laboratory. It is also difficult to evaluate the performance of the system fully without prior reference knowledge of the chemical composition of the sample, which was not available in this case. Also the process of collecting and analyzing the solution samples from the industrial sites would involve a significant cost, for any analysis at about 10 pounds per sample, a large sum when all the sample involved are taken into consideration. However, it was also noted that there exists a wide variety of industrial processes which discharge effluents often of known chemical composition. One common factor among many effluent discharges is that there are chemical species used in the industrial processes concerned that are dissolved or suspended in the discharge. Therefore these processes are more amenable to simulation in the laboratory, using chemical species which are expected to be present in the effluent of industrial sites. The simulation model used has been developed after consultation with process engineers working within the food industry, where the discharge of the hypochlorite and ammonia were identified as specific pollutants whose concentration is of interest for regular monitoring.

The second part of the work consists of the preparation of a data set from laboratory-based experiments, and the evaluation and development of appropriate data analysis. To generate the data set, samples were prepared with a wide range of concentrations, representative of those generally encountered in waste water treatment. This preparation was deemed to be not so dependent on the preciseness of the chemical preparation because the waste water measurement does not require the high degree of accuracy required in the pharmaceutical or food industries. The chemicals used in the study were sodium hypochlorite and liquid ammonia, which were pollutants typically expected in the industrial process, and sodium nitrate, which occurs at relatively high concentrations in supplied water and which has an absorption band overlapping those of the other two species. For the data analysis, the samples were prepared in two sets, a training and a testing set, this test representing a specific measurement application. In the data analysis evaluation, two analysis methods, both PCA and neural networks, were investigated. The data set for this evaluation was produced by transforming the intensity spectra obtained from the spectrometer to absorbance spectra which are used to relate the concentration of the chemical species using the Beer-Lambert law.

In reviewing the value of these techniques, the first, the PCA approach, is a linear technique that reduces the number of variables in a data set without loss of information. The advantage of this over the neural network technique is that it provides statistically verifyable results of known confidence. Another advantage is that it has been widely used, so that a considerable background of literature and operational suggestions exists. Although the technique has been reported in many articles, to be successful there is a strict need for linearity and operation under precise chemical preparation and ideal operating conditions in the laboratory, and thus this technique has serious limitations for waste water applications. Factory waste water is a very complex and varying mixture composed of the range of chemical substance used in factory and chemicals in the water supply. There are certainly significant interactions between these components, such as in the case of ammonia with chlorine to form new species such as monochloramine. In the analysis with PCA, the results showed the achievement of a good performance where exact linear responses were observed, such as in the prediction of the presence of nitrate. In contrast, a poorer performance in predicting of results was shown where non-linear responses of the ammonia and chlorine mixtures were observed, even though some attempt to linearize the data was made by assuming that the reactions were completed, whereby the new species, monochloromine, could be accounted for by the reduction in the amount of ammonia and chlorine measured. This assumption resulted in a poor performance in the data analysis because the chemical mechanism involved depends on several factors such as temperature, time, pH, etc. However, in reality, linearity is not always possible, so that the second approach, using neural networks based on non-linear techniques, was required.

Neural networks do give methods for analyzing problems that are not otherwise tractable and where there is promise that they may provide a method for classifying data without the need for a detailed knowledge of the chemical mechanism involved. It should be remembered that the alternative to such a measurement approach may require expensive chemical sampling schemes with storage and transport of batch samples to orthodox analytical laboratories for analysis, as is routinely done within the water industry today. However neural networks have one very serious problem in that solutions found with neural networks are not, in general, unique and there is no definite procedure for ensuring that the best solution or classification of data has been found in any specific case. To address this problem, several alternative preprocessing techniques were used in which various neural network algorithms including backpropagation neural networks were evaluated for classification and estimation.

Earlier work in which the whole data set was used for training the network were found ineffective in classifying the composition of the chemical species present in the water samples. Also, subsequent work where choosing a few strong correlated inputs by using the PCA technique was tried, or by choosing the absorption spectra in the UV range with their derivatives were also found ineffective. However the choosing of a few inputs based on a background knowledge of the expected species and their absorption spectra gave a better performance than the methods above which selected the network inputs by using mathematical methods alone.

Subsequently a more knowledge-based approach has been formulated which restricts itself to determining the presence and concentration of a range of expected species. This scheme involves a three stage process. In the first stage, shape information is derived by binary encoding segements of the second derivative of the absorption spectra according to their shape. The rationale for this stage is to reduce the spectral information to shape-sensitive factors. This is found significantly to ease the classification of the spectra by a second stage of backpropagation neural network analysis. For the estimation of the concentration of species, absorption data for the expected species are used to train a second stage of a backpropagation neural network. Segmentation of the spectra and the selection of relevant inputs for the second stage network are determined from the absorption data for the expected species, to give the best segmentation pattern and to optimize the number of network inputs. The combination of networks, with the observation of only the portions of the spectral data that relate to the expected chemical components, performs better than one network estimating all the chemical components simultaneously. As a result, the first-step network specifies which species are likely to occur and the second-step network can then focus on a few inputs that strongly correlate with the presence of the expected species. Also the second-step provides a filter that compensates for the classification of species at low concentration levels or incorrect identification of the species due to the low level signals in the presence of noise.

This three stage scheme forms a well structured and 'more understandable' approach to analyzing spectra than the 'black box' approach of the direct application of a Neural Network. Implementation of this scheme within an in-line monitoring system will require knowledge of the expected chemistry in the application. However this implementation could be relatively rapid and easily adapted to changing requirements, within a specific instrumental design.

The Intelligent User Interface, described in Chapter 4, completes and integrates the system software. This coordinates the collection of the data by the front end sensing system, and data analysis and presentation of results at the host computer. With the development of the successful technique in Chapter 6, the fully automated system with user friendly interface which was required was developed under the most popular graphical user interface, MS-WINDOWS. This GUI-software is very easy to use for those familiar with MS-Windows. The entry, dialog, menu, windows, buttons, and help are similar to most other applications running under the MS-Windows. The screen display is designed to be user-friendly and to present the data obtained in a real time animation. Also the use of Windows allows the use of dynamic data exchange with other applications such as Dynacomm and Neural Desk to make automatic real time data analysis. Data and result files are stored in CF_TEXT format which is able directly to be opened in most windows applications such as Excel, MS-word, Matlab, and etc. for further analysis or implementation.

The growing richness of the technological opportunities, the greater awareness of water pollution, and the individual nature of the outflow of specific industries means there is no panacea for user-computer interaction, and no single "right answer." It is appropriate to develop the right system to fit a specific purpose. This software system in this work is developed under C language which provides a flexible approach suitable, for adapting to the future needs in specific industry sites. The C language is also flexible enough to meet the requirements of almost any systems which require the integration of several front end sensors. Moreover, this GUI-software is developed under a Windows environment where hundreds of Windows applications are currently available. Thus this program can engage in dynamic data exchange with these hundreds of applications. For example, the Neural package can be modified to incorporate new algorithms from expert teams, or a new application involving a different set of chemical species can be studied, and all of that can be easy manipulated in this program. Furthermore the screens that are created by a Graphic tool as bitmap files that can be easily redesigned and also with the use of an editable set-up file, this makes it is easy to change the controlling parameters. Further, Windows has the reputation of being easy for users so that the aim of this application is for it to be widely used, although Windows programming is difficult even for experienced programmers.

To summarize, a critical artificial intelligence-based software system has been developed for the control and data analysis of UV-Vis spectrometer-based sensor systems which can be remotely controlled and provide real-time data analysis. The system was shown to be adaptable by being capable of detecting pollution incidents and of being configured to address a wide range of measurement tasks in chemical process and experimental monitoring and control.

7.2 Future Directions

Possibilities of the future direction of the work are determined by using the principle of encoding the second derivative spectra and the knowledge-based approach considered in data preprocessing for neural network analysis. With this, it provides a generic approach with the major application to the analysis of unresolved spectral features in UV-Vis absorption and fluorescence spectrometry. A possible example for application is in the better determination of the total organic carbon (TOC) in water which absorbs UV light at 254 nm from complex background spectra that from including nitrates. Moreover, it can also be applied to colorimetric measurement where it may help discern spectral features in the colour change of dyes or fluorphores used in detecting the presence of important metals such as Hg, Cd and Cr(VI). The three stage scheme is a well structured and 'more understandable' approach to analyzing spectra than the 'black box' approach of direct application of Neural Networks. Further, the scheme is also capable of incorporating other parametric data such as the pH of water and temperature and transient information from dynamic measurements. It should be stressed that implementation of this scheme within an in-line monitoring system will require a knowledge of the expected chemistry involved in the application. In many instances, knowledge can be obtained from consulting with an experienced chemist at the operational site to determine what chemical species are being discharged. A knowledge of the chemistry of each expected species will be used to model the most suitable sensors, and determine the appropriate network topology.

Work in this thesis is based on the use of a diode array spectrometer that scans through the UV, Visible and near Infrared spectrum. The chemical species used in this work directly absorbed UV-light while in other cases, chemical pollutants may need the use of reagents to form substances that absorb UV-Vis light. For example, the colorimetric determination of Ca and Mg[77] uses Arsenazo III and a buffer of 0.5 TRIS (Tris hydroxymethyl aminomethane), whose pH is adjusted to 8.5 and whose spectrum is registered between 450 and 650 nm.

The work is also applicable in the development of sensors based on differential absorption where this may be combined with the ability to control directly the concentrations of reagents through electro-chemical methods to give sensors which operate in a novel fashion. For example, this could occur using methods whereby the existence of species present in the water are inferred from the dynamic response of the sensor, employing signals giving changes in the electrical stimulus to reagent electrodes.

It is envisaged that the combining of electro-chemical and spectrometric methods will provide a definite advantage in the design of monitoring equipment, where a number of different variables may be determined using the same "front-end" instrument. However the control of optical spectrometers, electro-chemical sensors, and other instrumentation within a single sensing instrument will require the application of considerable effort in designing and implementation. It will also require a systematic approach to computer hardware interfacing and the development of associated control software that was undertaken in this work. Hence the success of this approach will greatly ease the potential for future developments such a those involving the addition of electro-chemical methods for solving other monitoring problems.

The more complex systems that are discussed would result in the requirement of an in-depth study of appropriate modelling and related mathematical and/or neural network techniques used for data analysis. This would include an integration of data from several sensor heads, and feedback to control the reagent supply.

For such more complicated systems, this thesis has given methods for the analyses of spectral data from UV-Vis sensor systems. However, these need to be applied to each particular industrial process under consideration. This study gives a methodology for applying data analysis successfully.

Most importantly, the techniques developed can be implemented using knowledge that focuses more on the individual factory site. Once a network is trained, it is able to be implemented rapidly in the determination of the compositions of the outflow stream.

Further development of this methodology, using a combination of a knowledge base in an expert system with the generalization of a Neural Network could be used to obtain a overall better performance. Some situations where it is too complicated to design expert system rules can profit from the generalization of the Neural Networks. The work here suggests that a knowledge-based approach, which offers the possibility of extracting knowledge directly from the time-series data, combined with a Neural Network analysis, and current information, can be extended to forecast the trend that is much more of value for the control of the process and to protect against any accident incidents.

Finally, the development of the GUI-software under C language is flexible enough to be adapted for the new sensing systems using the communication protocol under MS-Windows which provides for the integration of several Windows applications into one software system. With this facility, the software can be more easily adapted for data analysis and control of more complex systems in future work.

In conclusion, overall the work described in this thesis demonstrates the value of an automatic sensing system with remote control, real-time analysis, and graphical user interface in the field of monitoring chemical species in aqueous environments, and reveals, it is believed, the extent to which such methods have the potential for wider application in the future in the highly relevant and economically very important field of water quality and monitoring.

List of Publications by the author

1. Benjathapanun N., Boyle W. J. O. and Grattan K. T. V., (1995), "*The Application of Binary Encoded 2nd Differential Spectrometry in Preprocessing of US-Vis Spectral Data in the Estimation of Species Type and Concentration by Neural Networks*", Conference on: Mathematics of Neural Networks and Applications, Oxford, July 3-7, 1995.

2. Benjathapanun N., Boyle W. J. O. and Grattan K. T. V., (1995), "*Binary Encoded 2nd Differential Spectrometry using US-Vis Spectral Data and Neural Networks in the Estimation of Species Type and Concentration*", Paper submitted to IEE Proceeding A, Science, Measurement & Technology, 1995.

References

[1] Preece J., Rogess Y. , Sharp H. , Beneyon D., Holland S. and Carey T., (1994), *Human-Computer Interaction*, Addison-Wesley, Wokingham, U.K.

[2] Briggs R. and Grattan K.T.V., (1990), Instrumentation and Control in the U.K. Water Industry: A Review in Instrumentation Control and Automation of Water and Wastewater Treatment and Transport systems, Advances in Water Pollution Control, p. 27-38, Pergamon Press, Oxford.

[3] Briggs R., (1993), Status of Instruments, Unpublished manuscript.

[4] Boyle W.J.O., Han L., Briggs R., Mouaziz Z., Grattan K.T.V., Carr-Brion K.G. and Dawdswell R., (1993), "Optical Based Sensor System for Water Pollution Monitoring Incorporating Sensor Elements and Intelligent Processing", Accepted for the 6th IAWQ Automation Workshop, Banff, Canada, June 17-25, 1993.

[5] MacGraith B., Grattan K.T.V., Connally D., Briggs R., Boyle W.J.O. and Avis M., (1993), "Cross-comparison of Techniques for monitoring Total Organic Carbon (TOC) in water sources and supplies." Accepted for the 6th IAWQ Automation Workshop, Banff, Canada, June 17-25, 1993.

[6] Briggs R. and Melbourne K.V., (1972), "Ion-Selective Electrodes in Water Quality Monitoring Techniques in Air and Water Solution", Institute of Mechanical Engineers, London, U.K., p. 37-64.

[7] Snell F.D. and Snell C.T., (1954), “Colorimetric Methods of Analysis”, 3rd ed. Vol. II, D. Van Norstrand, NewYork, p. 797-799.

[8] Mouaziz Z., Briggs R., Hamilton I., Grattan K.T.V., (1993), “Design and Implementation of Fibre-Optic based Residual Chlorine Monitor”, Sensors & Actuators-B Chemical, Vol.11, p. 431-440.

[9] Danigel H., Jeltsch K. and Obergfell P., (1993), “Computer-controlled waste water monitoring for industrial purposes”, Chemometrics and Intelligent Laboratory Systems, Vol.19, 1993, p. 181-185.

[10] Pjovoso M. J., Kosanovich K. A., and Yuk J. P., (1991), "Process Data Chemometrics", IEEE, 1991, p. 608-612A.

[11] Saaksjarvi E., Khalighi M. and Minkkinen P., (1989), "Waste Water Pollution Modelling in the Southern Area of Lake Saimaa, Finland, by the SIMCA Pattern Recognition Method", Chemometrics and Intelligent Laboratory Systems, Vol. 7,1989, p. 171-180.

[12] Dale J. M. and Klatt L. N., (1989), "Principal Component Analysis of Diffuse Near-Infrared Reflectance Data from Paper Currency", Applied spectroscopy, Vol. 43, no.8, 1989, p.1399-1405.

[13] Robert P., Bertrand D., Crochon M. and Sabino J., (1989), "A new mathematical procedure for NIR analysis: The Lattice Technique. Application to the prediction of sugar content of apples", Applied Spectroscopy, Vol. 43, no. 6, 1989, p. 1045 - 1049.

[14] Glick M. and Hieftje G. M., (1991), "Classification of Alloys with an Artificial Neural Network and Multivariate Calibration of Glow-Discharge Emission Spectra", Applied Spectroscopy, Vol, 45, no. 10, 1991, p. 1706 - 1716.

[15] Donahue S. M., Brown C. W. and Scott M. J., (1990), "Analysis of Deoxyribonucleotides with Principal Component and Partial Least-Squares Regression of UV Spectra after Fourier Preprocessing", Applied Spectroscopy, Vol. 44, no.3, 1990, p. 407 - 413.

[16] Gemperline P. J. and Salt A., (1989), "Principal Components Regression for routine Multicomponent UV Determinations: A Validation Protocol", Journal of Chemometrics, Vol. 3, 1989, 343-357.

[17] Pizarro C., Sarabia L. A. and Palacios J. L., (1988), "Multivariate Calibration in UV-Vis Spectroscopy", Journal of Chemometrics, Vol.3, 1988, p. 241-247.

[18] Zhongxiao P., Siqing X., Shengzhu S., Maosen Z., Leming S. and Xinan L., (1990), "Target Factor Analysis: UV Spectrophotometry for the Simultaneous Determination of six Amino Acids", Journal of Chemometrics, Vol. 4, 1990, p. 323-330.

[19] Blanco M., Coello J., Iturriaga H., Maspoch S. and Redon M., (1994), "Principal Component Regression for Mixture Resolution in control analysis by UV-Visible Spectrophotometry", Applied Spectroscopy, Vol.48, 1994, no.1, p.37-43.

[20] Cadet F., Bertrand D., Robert P., Maillot J., Dieudonne J. and Rouch C., (1991), "Quantitative Determination of Sugar Cane Sucrose by Multidimensional Statistical Analysis of their Mid-Infrared Attenuated Total Reflectance Spectra", Applied Spectroscopy, Vol.45, no.2, 1991, p.166 - 172.

[21] Gemperline P. J., Long J. R. and Gregoriou V. G., (1991), "Nonlinear Multivarate Calibration Using Principal Components Regression and Artificial Neural Networks", Anal. Chem., Vol. 63,1991, p. 2313-2323.

[22] Borggaard C. and Thodberg H. H., (1992), "Optimal Minimal Neural Interpretation of Spectra", Anal. Chem., Vol. 64, 1992, p. 545-551.

[23] Peel C., Willis M. J. and Tham M. T., (1992), "A fast Procedure for the training of neural networks", J. Proc. Cont., Vol.2, 1992, No.4, p. 205-211

[24] Sharma A. K., Sheikh S., Pelczer I. and Levy G. C., 1994, "Classification and Clustering: Using Neural Networks", J. Chem. Inf. Comput. Sci., Vol. 34,1994, p.1130-1139.

[25] Mackenzie M. D., (1994), "Counterpropagation Networks Applied to the Classification of Alkanes through Infrared Spectra", Neural Comput & Applic, Vol.2, 1994, p. 111-116.

[26] Lee S. E. and Holt B. R., (1992), "Regression Analysis of Spectroscopic Process Data Using a Combined Architecture of Linear and Nonlinear Artificial Neural Networks", IEEE, 1992 ,p. iv 549 - iv 554

[27] Errington P. A. and Graham J., (1993),"Classification of Chromosomes using Combination of Neural Networks", IEEE, 1993, p. 1236 - 1241.

[28] Long J. R., Gregoriou V. G. and Gemperline P. J., (1990), "Spectroscopic Calibration and Quantitation Using Artificial Neural Networks", Anal. Chem., Vol. 62, 1990, p.1791-1797.

[29] Liu Y., Upadhyaya B. R. and Naghedolfeizi M., (1993), "Chemometric Data Analysis using Artificial Neural Networks", Applied Spectroscopy, Vol.47, 1993, no.1, p. 12-23.

[30] Boger Z and Karpas Z., (1994), "Use of Neural Networks for Quantitative Measurements in Ion Mobility Spectrometry (IMS)", J. Chem. Inf. Comput. Sci., Vol. 34, 1994, p. 576-580.

[31] Ham F. M., Cohen G.M. and Cho B., (1991), "Improved Detection of Biological Substances Using A Hybrid Neural Network and Infrared Absorption Spectroscopy", IEEE, 1991, p. I-227 - I-232.

[32] Sundgren H., Winquist F., Lukkari I and Lundstrom I, (1991), "Artificial neural networks and gas sensor arrays: quantification of individual components in a gas mixture", Meas. Sci. Technol.,Vol.2, 1991, p. 454-459.

[33] Broten G. S. and Wood H.C., (1992), "A Neural Network Approch to Analyzing Multi Component Mixtures", IEEE, 1992, p. II-957 - II-962.

[34] Broten G. S. and Wood H. C., (1993), " A neural network appreoach to analysing multi-component mixtures", Meas. Sci. Technol., Vol.4,1993, p. 1096-1105.

[35] McAnany D. E., (1993), "Practical applications of artificial neural networks in chemical process development", ISA Transactions, Vol.32, 1993, p. 333-337.

[36] Tanabe K., Tamura T. and Uesaka H., (1992), "Neural network system for the identification of infrared spectra", Applied Spectroscopy, Vol 46, 1992, No. 5, 807-810.

[37] Melssen W. J., Smits J. R. M., Rolf G. H. and Kateman G., (1993), “Two-dimension mapping of IR spectra using a parallel implemented self-organising feature map”, Chemometrics and Intelligent Laboratory Systems, Vol.18, 1993, p. 195 - 204.

[38] Minderman P. A. and McAvoy T. J., (1993), "Neural Net Modeling and Control of a Municipal Waste Water Process", Proceeding of the American Control Conference, San Francisco, California, June 1993, p. 1480 - 1484.

[39] Boger Z., (1992), "Application of Neural Networks to Water and Wastewater Treatment Plant Operation", ISA, Vol.31, 1992, no.1, p.25 - 33.

[40] Orlov Y. V., Persiantsev I. G. and Rebrik S. P., (1993), "Application of neural networks to fluorescent diagnostics of organic pollution in natural waters", IEEE, 1993, p. 1230 - 1235.

[41] Kurzweil R, (1990), *The Age of Intelligent Machines*, MIT Press, Cambridge, p.

[42] Calvert J. G. and Pitts, Jr, J. N., (1967), *Photochemistry*, John Wiley & Sons Inc., USA.

[43] Atkins P. W., (1978), *Physical Chemistry*, Oxford University Press, U.K.

[44] Graybeal J. D., (1988), *Molecular Spectroscopy*, McGraw Hill International Editions.

[45] Sommer, L., (1989), *Analytical Absorption Spectrophotometry in the Visible and Ultraviolet; The Principles*, Elsevier Science publisher, Netherlands.

[46] Mouaziz Z., Briggs R. and Grattan K. T. V., (1993), “Multi-parameter Fibre Optic Chemical Sensors for the Measurement of Nitrate Ion, Ammonia and Organic Matter”, in Proceedings of the Symposium on Chemical SensorsII, Edited by Buttler M., Ricco A. and Yamazoe N., The electrochemical Society Inc., New Jersey, U.S.A., Vol. 93-7, p. 303-316.

[47] Briggs R., Mouaziz Z. and Grattan K. T. V., (1991), “Design and Implementation of a Prototype Optically Based On-line Residual Chlorine Monitor”, Internal Report to Severn Trent Plc.

[48] Jobin Yvon Instruments S.A., (1992), SPECTRAVIEW 1D,Linear Diode Array Detection System & QuickView Software, Manaul, Jobin Yvon Instruments S.A., France.

[49] *Odyssey*, Manual.

[50] Dowdeswell R. M., (1994), "*Design of a Pollution Breakthrough Monitor Using Optical Sensing and an Artificial Neural Network Decision System*", PhD thesis : Cranfield University, School of Mechanical Engineering.

[51] Petzold C., (1992), *Programming Wondows 3.1*, Microsoft Press, U.S.A.

[52] Clark J. D., (1992), *Windows programmer's guide to OLE/DDE*, 1st ed., Sam Carmel Ind.

[53] Farrell T. and Connally R., (1992), *Programming in Windows 3.1*, 2nd ed., Que Corporation, U.S.A.

[54] Miller I. and Freund J. E., (1985), *Probability and Statistics for Engineers*, chapter 14: Applications to Quality Assurance, 3rd ed., Prentice Hall, London.

[55] Future Soft Engineering Inc., (1993), *DynaComm 3.1: User's Manual.*

[56] Neural Computer Sciences, (1992), *NeuralDesk: User's Guide.*

[57] Borland International Inc., (1992), *Borland C++ 3.1: Reference Manual.*

[58] Microsoft Corporation, *Microsoft Windows for Workgroups 3.1: User's Guide.*

[59] National Instruments, (1994), LabWindows/CVI Seminar, February 1994 ed., National Instruments Corporation.

[60] Pearson K., Philos. Mag., 1901,Vol. 2, p. 559.

[61] Wiley-Interscience, New York

[62] Aris R. E., Lidiard D. P. and Spragg R. A., (1991), “Principal Component Analysis”, Chemistry in Britain, September 1991, p. 821-824.

[63] Sharaf M. A., Illman D. L. and Kowalski B. R., (1986), *Chemical Analysis: Chemometrics*, John Wiley & Sons, New York.

[64] Marten H. and Naes T., (1989), *Multivariate calibration*, John-Wiley, Chichester.

[65] Hewlett-Packard, (1989), UV-Vis Operating Software and Handbook for the HP 8452A Diode Array Spectrophotometer, Hewlett-Packard, Germany., Hewlett-Packard, Germany.

[66] Winston P. H., (1992), Artificial Intelligence, 3rd ed., Addison-Wesley, New York.

[67] Azoff E. M., (1993), "Neural Network Principal Components Preprocessing and Diffraction Tomography", Neural Comput Applic, Vol.1, 1993, p.107-114.

[68] Ricard D., Cachet C. and Bass D.C., (1993), "Neural Network Approach to Structural Feature Recognition from Infrared Spectra", J. Chem. Inf. Comput. Sci., Vol. 33, 1993, p.202-210.

[69] Beaverstock M. C., (1993), "It takes knowledge to apply neural networks for control", ISA Transactions, Vol.32, 1993, p.235-240.

[70] Kohonen T., (1974), “An adaptive association memory principle”, IEEE Transactions, C-33, 444-445.

[71] Aleksander I. and Morton H., (1992), *An Introduction to Neural Computing*, Chapman & Hall, London.

[72] Karna K. N. and Breen D. M., (1989), "An artificial neural networks tutorial: part1 - basics, Neural Networks, Vol.1, No.1, January 1989, p.4-22.

[73] Rumelhart D. E. and McClelland J. L., (1992), *Parallel distributed processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, MIT Press, London, Chapter 8.

[74] Hammerstrom D., (1993), "Neural networks at work", IEEE Spectrum, June 1993, p. 26-32.

[75] Antonov L. and Stoyanov S., (1993), "Analysis of the Overlapping Bands in UV-Vis Absorption spectroscopy", Applied Spectroscopy, Vol.47, 1993, No.7, p.1030-1035.

[76] Bevington P. R., (1969), *Data Reduction and Error Analysis for the Physical Sciences*, McGraw-Hill, New York, 1969.

[77] Ruisanchez I., Rius A., Larrrechi M.S., Callao M.P., and Rius F.X., (1994), "Automatic simultaneous determination of Ca and Mg in natural waters with no interference seperation", Chemometrics and Intelligent Laboratory Systems, Vol.24, 1994, p. 55-63.

Appendix A

File Layout

CF_TEXT is a format that contains any numeric and decimal point data in which each number in a record seperated by TAB (ascii no. 9). Each record is ended by Carriage RETURN (ascii no.13) and Line Feed (ascii no. 10).).

Most files that run under MS-Windows can be interchanged between applications in CF_TEXT format. In this study, most files are also recorded in CF_TEXT format.

File name : **SETDATA.*****

Description : The spectrophotometer set up parameters of the front end sensor

Format : CF_TEXT

Record Length : vary

Total number of record : 1 record

Item.	Description	format
1.	S-DATA is number of files to get and record data in 60 seconds	integer
2.	L-DATA is number of files to get and record in 60 minutes	integer
3.	M-DATA is number of files to get and record in 24 hours	integer
4.	Temperature setting point to diode array	integer
5.	Exposure time	integer
6.	Time between scan is duration time between each scan	integer
7.	Time between M-DATA is duration time to record M-Data	integer
8.	Time between H-DATA is duration time to record for an hour.	integer
9.	Time between L-DATA is duration time to record for a day	integer
10.	Time between D-DATA is not used for now	integer

File name : **SETUP.TXT**

Description : Operational and Alarm Limit parameters

Format : CF_TEXT

Record Length : vary according to the numbers that user defined for each parameter.

Total number of record : 1

Item.	Description	format
1.	Cycle Time is a duration time to gather a new spectral data in seconds unit.	integer ≥ 5 (default =5)
2.	Record Time is a duration time to store the data and results	integer ≥ 5
3.	No. File Recycle is a maximum number of hours that can restore experimental data and results. Spectral data and results will be continually stored in RAM-DRIVE for an hour then transfer to hard disk with numeric extension filenames. These extension filenames are given a running number which is incremented from 001 until it reaches the number that is user defined and then starts at 001 again that means the previous one will be overwritten.	integer ≥ 1 and ≤ 999 (default =168)
4.	Turbidity alarm limit is a turbidity level that will be alarm if the measurement is exceed than this value.	decimal
5.	Hypochlorite ion limit is a concentration of hypochlorite ion that will be alarm if the measurement is exceed than this value.	decimal
6.	Nitrate limit is a concentration of nitrate that will be alarm if the measurement is exceed than this value.	decimal
7.	Monochloramine limit is a concentration of monochloramine that will be alarm if the measurement is exceed than this value.	decimal
8.	Current file number is a number of extension filename that is according to the last file number. This number is updated when the WWM is stoped. Once WWM is started again, this number will be increased to stored the data continuously.	integer

File name : **SPECDATA.TXT**

Description : absorption value from a remote front end sensing which pass through DDE with DYNACOMM

Format : CF_TEXT

Record Length : 2846 bytes

Total number of record : 1 record

Item.	Description	format
1.	absorption unit at 190 - 820 nm with 2 nm interval (316 values)	decimal x.xxxxxx

File name : **REFDATA.TXT**

Description : average of absorption unit from spectral data that is collected in 24 hours

Format : CF_TEXT

Record Length : 2846 bytes

Total number of record : 1 record

Item.	Description	format
1.	absorption unit at 190 - 820 nm with 2 nm interval (316 values)	decimal x.xxxxxx

File name : **SPECDATA.*****

Description : absorption data that gathered and stored for an hour

Format : CF_TEXT

Record Length : 2857 bytes

Total number of record : vary according to duration time to record which is equal to 3600 divided by duration time to record

Item.	Description	format
1.	Time which is recieved the data	hh:mm:ss PM or AM
2.	absorption unit at 190 - 820 nm with 2 nm interval (316 values)	decimal x.xxxxxx

File name : **OUTPUT.*****

Description : The turbidity level and concentration of chemical species that estimated by neural network approach then stored in an hour period.

Format : CF_TEXT

Record Length : 40 bytes

Total number of record : vary according to duration time to record which is equal to 3600 divided by duration time to record

Item.	Description	format
1.	Time which is recieved the data	hh:mm:ss PM or AM
2.	Turbidity level	decimal
3.	Hypochlorite ion concentration	decimal
4.	Nitrate concentration	decimal
5.	Monochloramine concentration	decimal

File name : **ALARM.*****

Description : The alarm message and the concentration that excees alarm limits

Format : CF_TEXT

Record Length : 21 bytes

Total number of record : vary according to number of alarm events in an hour.

Item.	Description	format
1.	Time which is alarm	hh:mm:ss PM or AM
2.	Alarm Code which is 0 = Instrument malfunction 1 = Turbidity alarm 2 = Hypochlorite ion alarm 3 = Nitrate alarm 4 = Monochloramine alarm	integer
3.	Value that exceed a limit	decimal

Appendix B

Graphical User Interface (GUI)

> "*Windows has the reputation of being easy for users but difficult for programmers. If at first you find Windows programming to be difficult, awkward, bizarrely convoluted, and filled with alien concepts, rest assured that this is a normal reaction. You are not alone.*"
>
> *Charles Petzold.* Programming Windows[67]

Microsoft Windows has emerged as the most popular graphical user interface (GUI) environment for MS-DOS. For the program developer, Windows provides a wealth of built-in routines that allow the use of menus, dialog boxes, and other components of a friendly user interface. Windows also contains an extensive graphics programming language that includes the use of formatted text in a variety of fonts. Programmers can treat the keyboard, mouse, video display, printer, and RS-232 communication port in a device-independent manner. Windows programs run the same on a variety of hardware configurations.

In earlier days, the video display was used solely to echo text that the user typed using the keyboard. In a GUI on the other hand, the video display itself becomes a source of user input. All GUIs make use of graphics on a bitmapped video display which requires a large memory space. Typically, the video display shows various graphical objects in the form of icons and input devices such as buttons and scroll bars. Using keyboard or mouse, the user can directly manipulate these objects on the screen, Graphics objects can be dragged, buttons can be pushed, and scroll bars can be scrolled. Rather than the one way cycle of information from the keyboard to the program to the video display, the user directly interacts with the objects on the display and the interaction between the user and a program becomes more intimate.

Of course, all of this user-friendliness does not come free. GUI system requires more expensive graphics-- higher resolution-- display systems. They also require more memory, more disk space, and faster processors to work efficiently. Windows and other GUI shells require even more resources because they are built on top of operating systems that were not designed with GUIs in mind. MS-DOS and other command-line operating systems have minimal hardware requirements when compared to just about any GUI operating system or shell. However, as hardware becomes cheaper and faster, efficiency arguments concerning memory and processing requirements become less convincing. In contrast, with MS-DOS command line, programming for GUIs like Windows is very difficult. Programming for windows is an all-or-nothing proposition. For example, you cannot write an MS-DOS application--even a well behaved one-- and use Windows only for some graphics. If you want to use any part of Windows, you must make the commitment to write a full-fledged Windows program. Everything in Windows is interconnected. For example, if you want to draw some graphics on the video display, you need something called a "handle to a device context." To achieve that, you need a "handle to a window." To get that, you must create a window and be prepared to receive "messages" from the Windows. To receive and process messages, you need a "window procedure." And at that point you are writing a Windows program.

Windows programming is certainly different from programming for a conventional environment like MS-DOS. Nobody will claim that Windows programming is easy. What a developer has to ask himself is: "Do I want my programs to use a more modern and productive user interface, one that includes menus, dialog boxes, scroll bar, and graphics?" If the answer is 'No,' readers can skip to section 6.2 on the waste water monitoring system. If the answer is 'Yes!', some very strange concepts requires mental reorientation will be encountered in the next section that almost every programmer who begins writing code for Windows must experience. These concepts are multi-tasking, Graphical Device Interfacing, Dynamic linking, and Memory management. These will be introduced in the following sections to familiarize comers to the Windows environment. For more details, ones can read several standard textbooks in Windows programming[67-69].

B.1 Multi-Tasking

Although some people continue to question whether multitasking is really necessary on a single-user computer, users definitely are ready for multitasking and can benefit from it as demonstrated by the popularity of MS-DOS RAM-resident programs such as Norton Commander or SideKick which allow fast switching. Under Windows, every program in effect becomes a RAM-resident popup. Several Windows programs can be displayed and run at the same time. Each program occupies a rectangular window on the screen. The user can move the windows around on the screen, change their size, switch between different programs, and transfer data from one program to another.

Everything that happens to a window is relayed by Windows to the window procedure of the user program in the form of a message. These messages can be posted to a message queue or sent directly to the windows procedure. The windows procedure then responds to this message in a manner defined by the programmer or by default processing. In most cases, when a program calls a function in Windows, the function will be processed and returned control to the program within a reasonable period of time. Sometimes, however, a program has a long job to do, and all other programs running under Windows seem to stop running during this time. In this case, Windows can be made to switch from then program by use of a high-priority messages such as *'TIMER'* etc. present in the queue. Therefore the programmer can program for the application to interrupt the queue or to give up the process to the waiting queue.

The first concept of Windows based, '*Multitasking*', has been introduced above. The next Windows concept, graphical user interface, that provides graphical devices without a hardware specification for the application program will be described in the next section.

B.2 Graphics Device Interface (GDI)

Programs written for windows do not directly access the hardware of the graphics display devices such as the screen and printer. Instead, Windows includes a Graphical Devices Interface (GDI) programming language that allows for hardware independent and easy to display of graphics. A program written for windows will run with any video board or any printer for which a Windows device driver is available. The program does not need to determine what type of device is attached to the system. Graphics in Windows are handled primarily by functions exported from the GDI.EXE module. The GDI.EXE module calls routines in the various driver files basically a file with an DRV extension for the video display screen and possibly one or more other driver files that control printers. The video driver accesses the hardware of the video display. Different video display adapters and printers require different driver files. Because a large number of different display devices can be attached to PC compatibles, one of the primary goals of GDI is to support device-independent graphics on output devices such as video displays, printers, plotters. Windows programs should be able to run without problems on any graphics output device that Windows supports. GDI accomplishes this goal by providing facilities to insulate application programs from the particular characteristics of different output devices. Windows can run on either a monochrome display or a color display. A program can be written without worrying very much about color. If a program is developed in color display and the program, later runs on a monochrome display adapter, Windows will use a shade of gray to represent each color. However, a program can also determine how many colors are available on the particular display device and take the best advantage of the hardware. To display graphics, a program can use a device context to create a graphical pictures or use bitmap command to represent each pixel in the picture.

Device Context: Windows gives a permission to an application program to use the *device context* such as pens, brushes, fonts, colors, icons, menu, scroll bars, windows caption bar and etc. A program must obtain a handle to a device context with a parameter in the GDI functions to identify to Windows the device you want to use. The device context contains many current "attributes" that determine how the GDI functions work on the device. For example, when a *TextOut* is called, the program need only specify in the function the device context handle, the starting coordinates, the text, and the length of the text. The program does not need to specify the font, the color of the text, and the intercharacter spacing, because these attributes are part of the device context. When the program wants to change one of these attributes, the program calls a function that changes the attribute in the device context.

Bitmaps and **Bit-block-transfer (BitBlt) :** The bitmap is a complete digital representation of the picture. Each pixel in the image corresponds to one or more bits in the bitmap. Monochrome bitmaps require only one bit per pixel; color bitmaps require additional bits to indicate the color of each pixel. A bitmap can be construct "manually" using the PAINTBRUSH program included with Windows. Then it can be included as a resource and loaded into a program using the *LoadBitmap* function.

Bitmaps have two major drawbacks. First, they are highly susceptible to problems involving device dependence. The most obvious device dependency is color. Display a colour bitmap on a monochrome device is often unsatisfactory. Another problem is that a bitmap implies a particular resolution and aspect ration of an image. Although bitmaps can be stretched or compressed, this process generally involves duplicating or dropping rows or column of pixels and can lead to distortion in the scaled image. The second major drawback of bitmaps is that they require a large amount of storage space. For instance, a bitmap representation of an entire 640-by-480-pixel, 16-color VGA screen requires over 150 KB.

Computer graphics involves writing pixels to a display device. The powerful pixel manipulation procedures in window are *BitBlt, PatBlt* and *StretchBlt.* The parameters that define a rectangle of a source and a destination are needed in these functions. The *PatBlt* ("Pattern block transfer") procedure alters a rectangular area of the device context destination. It performs a logical operation involving the pixels in this rectangle and a "pattern" which is simply another name for a brush. For this pattern, *PatBlt* uses the brush currently selected in the device context. The *BitBlt* ("bit-block-transfer") procedure modifies the destination device context. It performs a logical combination of three elements: the brush selected into the destination device context, the pixels in the source device context rectangle, and the pixels in the destination device context rectangle. *BitBlt* becomes most valuable in working with bitmaps that have been selected into a memory device context. When a program performs a "bit-block-transfer" from the memory device context to a device context for a client area, the bitmap selected in the memory device context is transferred to the client area. An animation program can use this procedure to draw a graphic into a memory device (RAM) and then uses BitBlt to transfer this bitmap to its destination in the client area (on the screen) and then changes the destination position this bitmap has moved to a new destination thus providing an animation graphic. The *StretchBlt* procedure allows programmer to flip an image vertically and/or horizontally. *StretchBlt* has some problems related to the inherent difficulties of scaling bitmaps. When expanding a bitmap, *StretchBlt* must duplicate rows or columns of pixels. If the expansion is not an integral multiple, then the process can result in some distortion of the image. When shrinking a bitmap, *StretchBlt* must combine two or more rows or columns pixels into a single row or column.

The aim of Windows based is not only for graphical tool. Other concepts which are make Windows much more powerful than just a resident program under MS-DOS is that Windows allows access to the application program interface of Windows or other applications through a dynamic linking process. The details of dynamic linking will be discussed in the next section.

B.3 Dynamic Linking

A Windows program interfaces to Windows through a process called "dynamic linking." Windows programs use a '.EXE' format called the New Executable file format. Whenever a Windows program calls a Windows function, the C compiler generates assembly-language code for a far call. A table in the '.EXE' file identifies the function being called using a dynamic link library name and either a name or a number of the function in that library. Windows itself consists largely of three dynamic link libraries, call KRNL386 (responsible for memory management, loading and executing programs, and scheduling), USER (the user interface and windowing), and GDI (the graphics). These libraries contain the code and data for most Windows functions. When a Windows program is loaded into memory, the far calls in the program are resolved to point to the entry of the function in the dynamic link library, which is also loaded into memory. However, since WINDOWS itself is a very large system, integrating it with other windows applications may be very difficult. This requires very good memory management. It is assumed that data is stored in each application. When an application is updated, the application data is changed, but it can not assume that the visible portion in the window of applications reflects the changes automatically. In many cases, this requires a large number of complex operations to transfer the data between applications such as Dynamic Data Exchange (DDE), DDE Management Libraries (DDEML), Object Linking and Embedding (OLE), Dynamic Link Libraries (DLL), and the Clipboard.

Dynamic Data Exchange (DDE): DDE is Microsoft's communication protocol by which it is possible to exchange data between applications in real time. DDE is based on the messaging system built into Windows. Two windows programs carry on a DDE "*conversation*" by posting messages to each other. These two programs are known as the "*server*" and the "*client.*" Although many programs that run under Windows support DDE, they each tend to have their own set of user commands and procedures. DDE conversations are set up between applications which take the role of *client* and *server*. A client can initiate and control a conversation whereas a server can only respond to data received from a client. With a warm link, the server advises the client that it has some information to transmit but does not send that data unless requested to do so by the client. With a hot link, the server sends both the advice and the data to the client application. The structure of any DDE communication comprises an address followed by a data block containing data and/or instructions to the server. The communication can cause data to be sent back to the instigating application giving a real-time type communication and processing ability between applications.

Dynamic Link Libraries (DLL): DLLs are files containing functions that can be called by programs and other DLLs to perform certain jobs. A DLL is brought into action only when another module calls one of the functions in the library. The term *dynamic linking* refers to the process that windows uses to link a function call in one module to the actual function in the library module. The code, data, and resources in a dynamic link library module are shared among all programs using the module. We will generally find that DLL make most sense in the context of a large application. For instance, The HELP About screen is a huge bitmap file which every module in WWM programs can call to display. These common routines could be put in a normal object library (.LIB) and then added to each of the program modules during static link. But this approach is wasteful, because if this routine is altered, every module would have to be relinked. If however, this routine is put in a dynamic link library (.DLL), only the library module need contain the routines required by all the application thus less memory space when running two or more of the applications simultaneously, and the library module can be changed without relinking any of the individual program. The routines that are contained in DLLs are capable of being added and removed while an application is running.

As mention before, Windows' itself is a very large system, therefore integrating it with other Windows applications requires very good memory management, as discussed next.

B.4 Memory Management

> *"Multitasking without memory management is like having a party in a closet: You may be able to accommodate some of the earlier arrivals, but once everybody starts mingling some toes are going to get smashed."*
>
> *Charles Petzold.* Programming Windows[67]

An operating system cannot implement multitasking without doing something about memory management. As new programs are started up and old ones terminate, memory use can become fragmented. The system must be able to consolidate free memory space. This requires the system to move blocks of code and data in memory. Programs running under Windows can overcommit memory; a program can contain more code than can fit into memory at any one time. Windows can discard code from memory and later reload the code from the program's .EXE file. Programs running in Windows can share routines located in other .EXE or .DLL files called "dynamic link libraries." Windows includes a mechanism to link the program with the routines in the dynamic link libraries at run time.

Memory Organization in Windows: The entire memory area that Windows controls is called "*global memory*" or the "*global heap*." This area begins at the location where MS-DOS first loads Windows into memory and ends at the top of available memory, which most often is the top of physical memory. Every block of memory allocated from the global heap is a *segment*. When Windows loads a program into memory, it allocates at least one segment from the global heap for code and one segment for data. When the program begins to execute, the microprocessor's Code

Segment (CS) register is set to the segment address of the code segment that contains the entry point of the program. The Data Segment (DS) and Stack Segment (SS) registers are set to the segment that contains the program's default data segment, which is the data segment that contains the stack. When loading a program, Windows also allocates two other segments from the global heap for program overhead. One of these segments contains the header portion of the program's .EXE file. This segment is used for all instances of a program, so it is allocated only for the first instance. The other segment contains information unique to each instance, such as the program's command-line string and the program's current subdirectory. When a program loads resources (such as icons, cursors, or menu templates) into memory, each resource gets its own segment in the global heap. A program may itself also allocate some memory from the global heap.

If a program has only one code segment, any calls it makes to functions within the program are compiled as a near call. The CS code segment register remains the same. However, when a Windows program passes a pointer to Windows function, the pointer must be far pointer, otherwise, the code that contains the Windows function would use its own data segment. Far pointers are required so that the Windows function can access the data within your program's data segment.

Fixed and Moveable Segments: Every segment in Windows' total memory space is marked with certain attributes that tell Windows how to manage the segment. First and foremost, segments are marked as either "fixed" or "moveable." Windows can move moveable segments in memory if necessary to make room for other memory allocations. A fixed segment always resides at the same physical memory location where it was first allocated. Two ways that Windows deals with moveable segments. First, Windows maintains a segment called BURGERMASTER that contains a master handle-to-memory table. The handle points to a small area of memory within BURGERMASTER that contains the segment address of the item that the handle references. When Windows moves the segment that contains the item, it can adjust the address in BURGERMASTER without invalidating the handle. BURGERMASTER is itself a moveable segment.

Second, Windows does not directly make calls to windows procedures, dialog procedures, or call-back functions. Instead, Windows builds a small piece of code called a "thunk," which assigns the segment address of the default data segment before entering the program. (This is the purpose of the *MakeProcInstance* function.) When Windows moves your default data segment, all it needs to do is change the thunk.

A programmer should try very hard to ensure that the code and data segments of his Windows programs are moveable segments. Fixed segments stand like brick walls in memory space and clog up Windows' memory management.

Discardable Memory: Moveable segments can also be marked as discardable. This means that when Windows needs additional memory space, it can free up the area occupied by the segment. Windows uses a "least recently used" (LRU) algorithm to determine which segments to discard when attempting to free up memory.

Discardable segments are almost always read-only segments that do not change after they are loaded. Code segments of Windows programs are discardable because (in most cases) programs do not modify their code segments. When Windows discards a code segment, it can later reload the code segment by accessing the .EXE file. Most of Windows' own code in the USER and GDI modules and in various driver libraries is also discardable. (The Kernel module is an exception because this is the module responsible for Windows' memory management.) Resources (such as dialog box templates, cursors, and icons) also are often marked as discardable. Again, Windows can simply reload the resource into memory by accessing the .EXE file that contains the resource.

Discardable segments must also be moveable segments because discardable segments can be reloaded in a different area of memory than the area they occupied earlier. However, movable segments are not always discardable segments. This is usually the case with data segments. Windows cannot discard a program automatic data segment because the segment always contains read-write data and the stack.

To summerize, some important concepts on Windows based system have been introduced to exemplify of how the Waste Water Monitoring (WWM) software performs multi-tasking and dynamic linking to other application programs. This will show that how the Windows environment is used to implement software for monitoring and at real-time analysis industrial effluents in our sensor systems as described in chapter 4.

Appendix C
WWM Reference Manual

C.1 MS-Windows and MS-DOS

MS-Windows is started up as a normal application program under MS-DOS but as windows loads, it becomes almost a full-fledged operating system. It is not quite an operating system because it runs on top of MS-DOS. While Windows is running, it shares responsibility with MS-DOS for managing the hardware resources of the computer. Basically, MS-DOS continues to manage the file system, while windows does everything else. Windows commands the video display, keyboard, mouse, printer, and serial ports. It is also responsible for memory management, program execution, and scheduling.

Every window that a program creates has an associated window procedure. This window procedure is a function that could be either in the program itself or in a dynamic link library. MS-Windows sends a message to a window by calling the window procedure. The window procedure does some processing based on the message and then returns control to windows. This window procedure located in the Windows USER.EXE dynamic link library.

A window procedure processes messages to the window. Very often these messages inform a window of user input from the keyboard or mouse. This is how a push-button windows that it's being "Pressed," for example. Other messages tell a window when it is being resized or when the surface of the window needs to be repainted.

When a Windows program begin execution, Windows creates a "message queue" for the program. This message queue stores messages to all the various windows a program may create. The program includes a short chunk of code called the "message loop" to retrieve these messages from the queue and dispatch them to the appropriate window procedure. Other messages are sent directly to window procedure without being placed in the message queue.

C.2 The WWM Files List

Normally, it needs at least three files to create a executable program. These are:

- a make file (.MAK),
- a C source code file (.C), and
- a module definition file (.DEF).

It may include a resource file(.RC) and a header file(.H).

The table C.1 is a list of all files which are used in the WWM software system.

item	filename	Description
1.	*.BMP	These following are bitmaps files which are created by PAINTBRUSH: • LOGO1.BMP - LOGO8.BMP for display a title screen • 0_0.BMP, 50_50.BMP,100%.BMP for display the results from NeuRun which are presence or absent or under determine • ONLINE.BMP for display on-line screen • STATIS.BMP for display statistics screens • ABOUT.BMP,ABOUT2.BMP for display help about.
2.	LOGO.DLL ABOUT.DLL, ABOUT2.DLL	This dynamic link library contains bitmaps for the title screen and the help about screen.

3.	BITLIB.MAK BITLIB.C BITLIB.RC BITLIB.DEF	These are used to create library files called LOGO.DLL and ABOUT.DLL.
4.	WWM.MAK WWM.C WWM.RC WWM.DEF WWM.H WWM.EXE	This file group is used to create executable file called WWM.EXE which is the main program of this system.
5.	DYNACOMM with WWM.DCT	After Dynacomm is started up, the WWM.DCT script file has to execute. This script file will wait for data from communication line and transfer the data to WWM.EXE
6.	NEURUN.EXE and *.NCS	The run-time processor NeuRun-classify.ncs will be invoked by WWM.EXE and start DDE communication.

Resources File: Icons, cursors, menus, and dialog boxes are all examples of "resources." Resources are data and are included in a program's .EXE file, but they do not reside in a program's normal data segment. When Windows loads a program into memory for execution, it usually leaves the resources on disk. Only when Windows needs a particular resource does it load the resource into memory. If Windows runs out of memory, segments occupied by resources can be freed up. If the resource is required again later, Windows reloads it from the .EXE file.

Make File: The WWM.MAK make file contains three sections. The first section runs the linker if either WWM.OBJ, WWM.DEF, or WWM.RES has been altered more recently than WWM.EXE:

```
wwm.exe : wwm.obj wwm.def wwm.res
    $(WINLINK) wwm, wwm, NUL, $(WINLIB), wwm
    rc -t wwm.res
```

The second section runs the C compiler if WWM.C has been changed more recently than WWM.OBJ:

```
wwm.obj : wwm.c  wwm.h
    $(WINCC) wwm.c
```

The third section compiles the .RC resource script into a binary .RES file:

```
wwm.res : wwm.rc wwm.h
    $(WINRC) wwm.rc
```

Definition File: In addition to the C source code, another file is required for Windows programs. It is called a "module definition file" and has the extention .DEF. The module definition file aids the linker in creating the .EXE file by telling it the characteristics of the program's code and data segments, the size of the program's stack. This information becomes part of the header section of the New Executable file format. The WWM.DEF file is shown here.

```
;-----------------------------------
; WWM.DEF module definition file
;-----------------------------------
NAME         WWM
DESCRIPTION   'Waste Water Monitoring (c) N. Benjathapanun, 1995'
EXETYPE       WINDOWS
STUB         'WINSTUB.EXE'
CODE         PRELOAD MOVEABLE DISCARDABLE
DATA         PRELOAD MOVEABLE MULTIPLE
HEAPSIZE      4096
STACKSIZE     8192
```

DYNACOMM Script: The WWM.DCT is the script file that recieve data from communication line and transfer data into SPECDATA.TXT for WWM.EXE to analyte and store it later. The script commands in WWM.DCT are shown in table C.2. This script file is shown here:

```
when string "EOF" resume
File Receive text 'd:\text.txt'
wait resume
file close
FILE COPY 'd:\text.txt' TO 'd:\specdata.txt'
hangup
execute "WWM"
```

Table C.2 DYNACOMM Script Langauge.

Command	Explanation
WHEN STRING	is activated when the specified string is received through the communications port for a terminal window.
RESUME	causes script execution to resume at the command following the most recently executed WAIT command.
FILE RECEIVE TEXT	prepares DynaComm to receive the specified file using the text transfer protocol specified by the active settings file.
WAIT RESUME	pauses execution until Script:Resume is selected or the RESUME command is executed.
FILE COPY	copies the contents of the source file to the destination file, creating the destination file if it does not exist.
HANGUP	directs the modem to disconnect the telephone line.
EXECUTE	causes script execution to restart at the specified target.

NeuRun and DDE: The structure of any DDE communication comprises an address followed by a data block containing data and/or instructions to the server. The communication can cause data to be sent back to the instigating application giving a real-time type communication and processing ability between applications.

NeuRun will always be the background activity so its operation is not usually apparent to the user except in the form of results presented back in the application.

<u>Address:</u> Initially, messages are broadcast to all applications, but only the address comprises three separate parts: APPLICATION NAME, TOPIC and ITEM. During initialisation, when an attempt is made to start a conversation, the Application name of the conversee is transmitted to all applications together with a Topic name which specifies the subjecr of the conversation. Once a conversation has started, the individual BITs of data are communicated as Items. The following sections give example of communication between Borland C and NeuRun. It also gives details of the information that needs to be communicated to NeuRun.

```
hConv = DdeConnect (idInst, hszService, hszTopic, NULL) ;
```

The DdeConnect function establishes a conversation with a server application that supports the specified service name and topic name pair. This is "NeuRun" and "classify.ncs" for hszService and hszTopic.

```
DdeClientTransaction ((void FAR*) hData, -1, hConv,NULL, CF_TEXT,
                                    XTYP_EXECUTE ,
                                    DDE_TIMEOUT,NULL) ;
```

The DdeClientTransaction function begins a data transaction between a client and a server. Only a dynamic data exchange (DDE) client application can call this function, and only after establishing a conversation with the server. This command is executed four times by changing the hData to:

- [SetFile(InterrogStimulus,d:\\test.txt)]
- [Process(Relate)]
- [SetFile(InterrogResponse,d:\\out.txt)]
- [Process(Quit)]

These are according to tell NeuRun that

- the data in file TEST.TXT is for query process,
- sets the processing state of the run time network to the interrogate state,
- the results from query process are saved in OUT.TXT file and
- closes the NeuRun window.

```
DdeDisconnect (hConv) ;
```

The DdeDisconnect function terminates a conversation started by either the DdeConnect and invalidates the given conversation handle. It terminates a conversation with NeuRun-classify.ncs and then start conversation with NeuRun-ocl.ncs, or other concentration networks.

```
hConv = DdeConnect (idInst, hszService, hszTopic, NULL) ;
DdeClientTransaction ((void FAR*) hData, -1, hConv,NULL, CF_TEXT,
                                        XTYP_EXECUTE ,
                                        DDE_TIMEOUT,NULL) ;
DdeDisconnect (hConv) ;
```

The one of following files is selected according to the results from the classify network and added one more command SCALING correspond to the concentration of each network by DdeClientTransaction commamnd. Table C.3 shows the concentration networks and corresponding scale.

Table C.3 File name and scale of each concentration network.

Network	Scale
OCL.ncs	[Scaling(-0.03,3.125,-4,0.02439)]
NO3.ncs	[Scaling(-0.03,0.75188,-1,0.133333)]
OCLNO3.ncs	[Scaling(-0.025,0.597015,-1,0.022727)]
NH2CL.ncs	[Scaling(-0.035,1.941748,-4,0.02439)]
OCLNH2CL.ncs	[Scaling(-0.03,1.923077,-3,0.02381)]
NO3NH2CL.ncs	[Scaling(-0.03,0.581395,-1,0.022727)]
OCLNONHC.ncs	[Scaling(0,0.5,0,0.022222)]

Appendix D

Principal Component Analysis Mathematics Procedure

D.1 Principal Component Analysis

In PCA the eigenvectors are consecutively calculated so as to minimize the residual error in each step. For example, $d_{ik}(m)$ is the reproduced data point in the *i* th row and *k* th column calculated from the first *m* principal factors. Hence we write

$$d_{ik}(m) = \sum_{j=1}^{m} r_{ij} c_{jk} \qquad \text{....(D.1)}$$

where the sum is taken over the first *m* principal factors. To obtain the first principal component, we proceed as follows. First, we define the residual error, $e_{ik}(1)$

$$\begin{aligned} e_{ik}(1) &= d_{ik} - d_{ik}(1) \\ &= d_{ik} - r_{i1} c_{1k} \qquad \text{....(D.2)} \end{aligned}$$

To minimize the residual error, we take the derivative of the square of each residual error and set equal to zero, finding

$$\sum_{i=1}^{r} \frac{de_{ik}^{2}(1)}{dc_{1k}} = 2c_{ik} \sum_{i=1}^{r} r_{i1}^{2} - 2\sum_{i=1}^{r} r_{i1} d_{ik} = 0 \Rightarrow\Rightarrow \sum_{i=1}^{r} r_{i1} d_{ik} = c_{ik} \sum_{i=1}^{r} r_{i1}^{2} \qquad \text{....(D.3)}$$

Since k varies from 1 to c equations of this form, which in matrix notation can be expressed as follows:

$$R_1^{'}[D] = C_1^{'} R_1^{'} R_1 \qquad \text{....(D.4)}$$

We now define λ_1 by

$$\lambda_1 = R_1^{'} R_1 = \sum_{i=1}^{r} r_{i1}^{2} \qquad \text{....(D.5)}$$

Inserting this into (D.4) and taking the transpose, we find that

$$[D]^T r_1 = \lambda_1 C_1 \qquad \text{....(D.6)}$$

According to [R][C] ^{T}, the complete data matrix can be written as

$$[D] = R_1C_1' + R_2C_2' + \cdots + R_cC_c' \qquad \text{....(D.7)}$$

Postmultiplying (D.7) by C_1 and setting $C_i'C_j = \delta_{ij}$, so that the eigenvectors are orthonormal, we obtain

$$[D]C_1 = R_1 \qquad \text{....(D.8)}$$

Inserting (D.8) into (D.6), we conclude that

$$[D]^T[D]C_1 = \lambda_1 C_1 \Leftrightarrow [Z]C_1 = \lambda_1 C_1 \qquad \text{....(D.9)}$$

Now we get the first principal eigenvectors,C_1 ,and its associated eigenvalue,λ_1 .

To obtain the second principal component, we consider the second residual error,

$$e_{ik}(2) = d_{ik} - d_{ik}(2)$$

$$= e_{ik}(1) - r_{i2}c_{2k} \qquad \text{....(D.10)}$$

To minimize the error in the second principal component we apply the method of least squares to $e_{ik}(2)$ while keeping $e_{ik}(1)$ constant. Thus we find an exression analogous to (D.3) and (D.4)

$$R_2'[E]_1 = C_2'R_2'R_2 \qquad \text{....(D.11)}$$

where[E]$_1$ is an *r* x *c* error matrix composed of the first residual errors. Now λ_2 is defined as

$$\lambda_2 = R_2'R_2 = \sum_{i=1}^{r} r_{i2}^2 \qquad \text{....(D.12)}$$

From (D.11) and (D.12) we learnt that

$$[E]_1 R_2 = \lambda_2 C_2 \qquad \text{....(D.13)}$$

$$[E]_1 = [D] - R_1C_1' = R_2C_2' + R_3C_3' + \cdots + R_cC_c' \qquad \text{....(D.14)}$$

Postmultiplying (D.14) by C_2 and recalling that the eigenvectors are to be orthonormal, we obtain

$$[E]_1 C_2 = R_2 \qquad \text{....(D.15)}$$

Inserting this equation into (D.13), we conclude that

$$[E]_1^T[E]_1 C_2 = \lambda_2 C_2 \qquad \text{....(D.16)}$$

from (D.4), (D.5) and (D.14), we can show that

$$[E]_1^T[E]_1 = [D]^T[D] - \lambda_1 C_1 C_1' \qquad \text{....(D.17)}$$

The first residual matrix is defined as

$$[\Re]_1 = [Z] - \lambda_1 C_1 C_1' \qquad \text{....(D.18)}$$

Hence we conclude from (D.16), (D.17) and (D.18) that

$$[\Re]_1 C_2 = \lambda_2 C_2 \qquad \text{....(D.19)}$$

To obtain the third principal component we apply the method of least squares to $e_{ik}(3)$. In this case we obtain

$$[\Re]_2 C_3 = \lambda_3 C_3 \qquad \text{....(D.20)}$$

where the second residual matrix is defined as

$$[\Re]_2 = [Z] - \lambda_1 C_1 C_1' - \lambda_2 C_2 C_2' \qquad \text{....(D.21)}$$

We continue in this fashion to successively extract the remaining eigenvectors. We find that

$$[\Re]_m C_{m+1} = \lambda_{m+1} C_{m+1} \qquad \text{....(D.22)}$$

where

$$[\Re]_m = [Z] - \sum_{j=1}^{m} \lambda_j C_j C_j' \qquad \text{....(D.23)}$$

Principal component analysis yields a unique set of mutually orthonormal eigenvectors which represents the coordinate axes of the data space. The first eigenvector that emarges from the iteration is associated with the largest eigenvalue and accounts for most of the variance of the data. This vector is oriented in a direction that maximizes the projections of the data points onto this axis. Each eigenvector that emerges from the iteration is orthogonal to all the previous eigenvectors and is oriented in the direction that maximizes the sum of squares of all projections on the axis

Here we see that the eigenvectors, which results from the iteration process, constitute the respective rows of the column matrix,[C].

$$[C] = \begin{bmatrix} C_1' \\ C_2' \\ \vdots \\ C_c' \end{bmatrix} = [C_1 \quad C_2 \quad \cdots \quad C_c]^T \qquad ....(D.24)$$

Having obtained matrix[C], we can calculate the complete set of numerical values for the elements of matrix [R] by carrying out the multiplication shown as

$$[R] = [D][C]^{-1} = [D][C]^T \qquad ....(D.25)$$

Short-Circuit Reproduction

Although there are c eigenvectors, only n principal eigenvectors are required to span the factor space. Consequently,

$$[R_1 \quad R_2 \quad \cdots \quad R_n] \begin{bmatrix} C_1' \\ C_2' \\ \vdots \\ C_n' \end{bmatrix} = [R^\tau][C^\tau] = [D^\tau] \approx [D] \qquad ....(D.26)$$

$[D^\tau]$ is the reproduced data matrix, using *n* eigenvectors. The minimum number of eigenvectors, n, required to reproduce the data within experimental error represents the number of factors involved. This number also represents the "dimensionality" (rank or size) of the factor space. We call the abstract factors are produced by *abstract factor analysis* (AFA).

D.2 Transformation

In principle there are an infinite set of axes which can be used to define the plane and locate the data points. Similarly, in factor analysis we may rotate the reference axes as long as we keep them distinct and as long as they adequately span the space. *Target transformation* is unique because, in spite of the complexity of the data space, it allows us to search for the basic factors *individually*. This can be seen by examining (D.41),which concerns the mathematical operation involved in transforming the eigenvector axes.

$$\left[\overline{R}\right]=\left[R^{\tau}\right]\left[T\right] \qquad \text{....(D.27)}$$

$$\overline{R}_l=\left[R^{\tau}\right]T_l \qquad \text{....(D.28)}$$

We call $\overline{R}_l$ the predicted vector (the *l* th column of newly transformed row matrix) and T_l the associated transformation vector. We wish to find the transformation vector thet yields an $\overline{R}_l$ most closely matching $\overline{\overline{R}}_l$, the *test vector* that we suspect is a basic factor. This test vector is our "target." To do this we carry out a least-squares procedure that minimizes the deviation between the test vector and the predicted vector. The difference between each element of the predicted vector and test vector is gevin as

$$\Delta r_{il}=\bar{r}_{il}-\bar{\bar{r}}_{il}=r_{i1}t_{1l}+r_{i2}t_{2l}+\cdots+r_{in}t_{nl}-\bar{\bar{r}}_{il} \qquad \text{....(D.29)}$$

To find the best T_l, the deviation between the test vector and the predicted vector is minimized by setting the sum of the the derivatives of the squares of the differences equal to zero.

$$\sum_{i=1}^{n}\frac{d(\Delta r_{il})^2}{dt_{1l}}=0=t_{1l}\sum_i r_{il}^2+t_{2l}\sum_i r_{i1}r_{i2}+\cdots+t_{nl}\sum_i r_{i1}r_{in}-\sum_i r_{i1}\bar{\bar{r}}_{il} \qquad \text{....(D.30)}$$

In order to express these equations in matrix form, we define the two vectors A_l , T_l

and
$$A_l=\begin{bmatrix}\sum r_{i1}\bar{\bar{r}}_{il}\\ \sum r_{i2}\bar{\bar{r}}_{il}\\ \vdots\\ \sum r_{in}\bar{\bar{r}}_{il}\end{bmatrix} \quad \text{and} \quad T_l=\begin{bmatrix}t_{1l}\\ t_{2l}\\ \vdots\\ t_{nl}\end{bmatrix} \qquad \text{....(D.31), (D.32)}$$

$$\text{matrix[B\}} \quad \left[B\right]=\begin{bmatrix}\sum r_{i1}^2 & \sum r_{i1}r_{i2} & \cdots\cdots & \sum r_{i1}r_{in}\\ \sum r_{i1}r_{i2} & \sum r_{i2}^2 & \cdots\cdots & \sum r_{i2}r_{in}\\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \vdots\\ \sum r_{i1}r_{in} & \sum r_{i2}r_{in} & \cdots\cdots & \sum r_{in}^2\end{bmatrix} \qquad \text{....(D.33)}$$

In matrix notation, the equations in (D.44) now become

$$A_l=\left[B\right]T_l \Leftrightarrow T_l=\left[B\right]^{-1}A_l \qquad \text{....(D.34)}$$

Upon examining (D.33) and (D.5), we see that

$$[B]=[R^{\tau}]^{T}[R^{\tau}]=[\lambda^{\tau}] \qquad ....(D.35)$$

$$A_{l}=[R^{\tau}]^{T}\overline{\overline{R}}_{l} \qquad ....(D.36)$$

$$T_{l}=[\lambda^{\tau}]^{-1}[R^{\tau}]^{T}\overline{\overline{R}}_{l} \qquad ....(D.37)$$

where λ^{τ} is a diagonal matrix composed of the primary eigenvalues only, $\overline{\overline{R}}_{l}$ is the test vector composed of the suspected parameters associated with the row designees and T_{l}, a column of [T] is the least-squares vector transformer.

Having obtained T_{l}, we use (D.52) to obtain numerical values for the elements of $\overline{R}_{l}$. We can then ascertain whether or not the following equation is obeyed within experimental error:

$$\overline{R}_{l} \overset{?}{=} \overline{\overline{R}}_{l} \qquad ....(D.38)$$

If our suspected test vector $\overline{\overline{R}}_{l}$ is a factor, each element of $\overline{R}_{l}$ will equal the corresponding element of $\overline{\overline{R}}_{l}$, within experimental error. It is possible to find a sufficient number of acceptable test vectors, but still not span the factor space, because some of the test vectors lie in a common subspace. To ascertain whether or not a particular set of test vectors adequately spans the space, we perform the following calculation:

$$[\overline{\overline{R}}][\overline{C}]=[D^{\tau}]_{TFA} \overset{?}{=} [D] \qquad ....(D.39)$$

$$[\overline{R}][\overline{C}]=[R^{\tau}][T][T]^{-1}[C^{\tau}]=[R^{\tau}][C^{\tau}]=[D^{\tau}]\equiv[D] \qquad ....(D.40)$$

D.3 Key combination set

Basic Factors. To find the key set,various combinations of acceptable basic vectors are formed into row-factor matrices, $[\overline{\overline{R}}]_{basic}$, and the following calculation is carried out:

$$[\overline{\overline{R}}]_{basic}[\overline{C}]_{basic}=[D^{\tau}]_{basic} \qquad ....(D.41)$$

where $$\left[\overline{\mathrm{C}}\right]_{\text{basic}} = \left[\mathrm{T}\right]^{-1}\left[\mathrm{C}^{\tau}\right] \qquad \text{....(D.42)}$$

A key set of basic vectors is found when $\left[\mathrm{D}^{\tau}\right]_{\text{basic}}$, the combination reproduced data, adequately equals the original data matrix:

$$\left[\mathrm{D}^{\tau}\right]_{\text{basic}} \cong \left[\mathrm{D}\right] \qquad \text{....(D.43)}$$

<u>*Typical Factors*</u>. Since the columns of the data matrix lie in the factor space, a judicious choice of n data columns can be used to describe the n-dimensional factor space. Such a combination is called a key set of typical vectors. The mathematical steps involved in this process are follows. The *l* th column of the data matrix can be looked upon as a vector $\overline{\overline{\mathrm{D}}}_l$ with the data points in the column representing the elements of the vector. In other words, $\overline{\overline{\mathrm{D}}}_l$ is used as a test vector. Letting $\overline{\overline{\mathrm{D}}}_l = \overline{\overline{\mathrm{R}}}_l$ and using (D.51), we find that

$$\mathrm{T}_l = \left[\lambda\right]^{-1}\left[\mathrm{R}\right]^{\mathrm{T}}\overline{\overline{\mathrm{D}}}_l \qquad \text{....(D.44)}$$

$$\left[\mathrm{T}\right] = \left[\mathrm{T}_a \quad \mathrm{T}_b \quad \cdots \quad \mathrm{T}_n\right] \qquad \text{....(D.45)}$$

Various combinations are employed in an attempt to find the key combination set that best reproduces the data matrix. Hence we search for a set of *n* data columns that best satisfies

$$\left[\overline{\overline{\mathrm{D}}}\right]_{\text{key}}\left[\overline{\mathrm{C}}\right] - \left[\mathrm{D}\right] = \text{minimum} \qquad \text{....(D.46)}$$

where

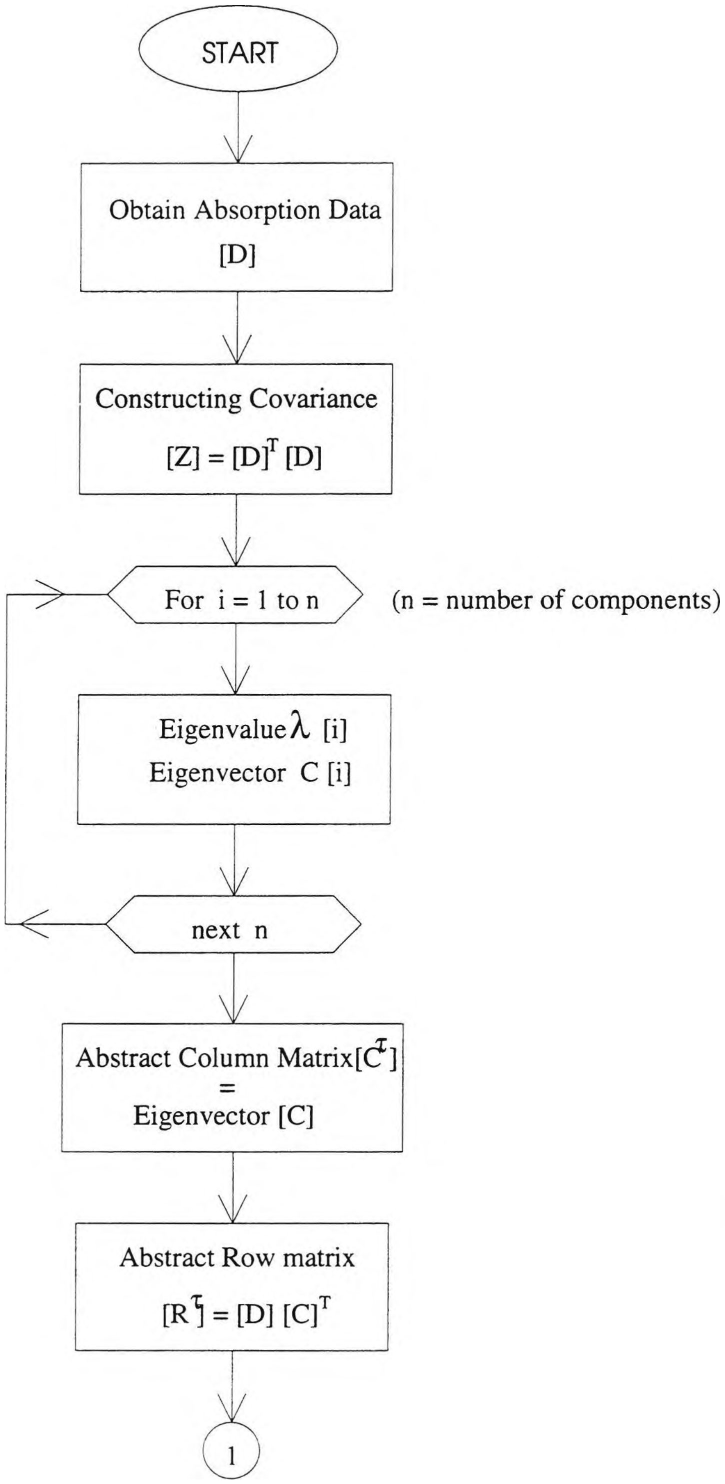
$$\left[\overline{\overline{\mathrm{D}}}\right]_{\text{key}} = \left[\overline{\overline{\mathrm{D}}}_a \quad \overline{\overline{\mathrm{D}}}_b \quad \cdots \quad \overline{\overline{\mathrm{D}}}_n\right] \qquad \text{....(D.47)}$$

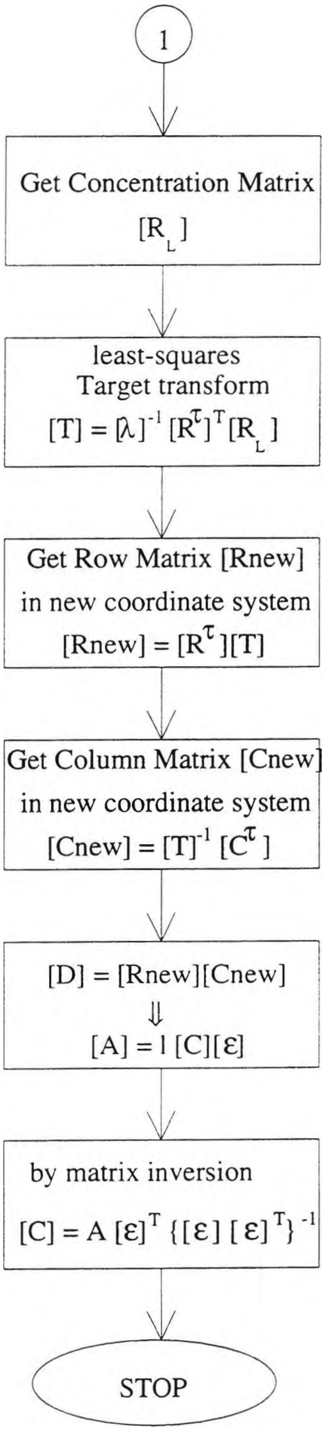
and $$\left[\overline{\mathrm{C}}\right] = \left[\mathrm{T}\right]_{\text{key}}^{-1}\left[\mathrm{C}^{\tau}\right] \qquad \text{....(D.48)}$$

There are several advantages to using data columns as factors of the space. First, they are more easily visualized than the abstract factors produced by factor analysis. Second, their use precludes the need to identify the true controlling factors. Third, empirical predictions can be made quickly from the resulting equations.

Appendix E

PCA Flow Chart





(Rnew identical to R_L)
except computational error

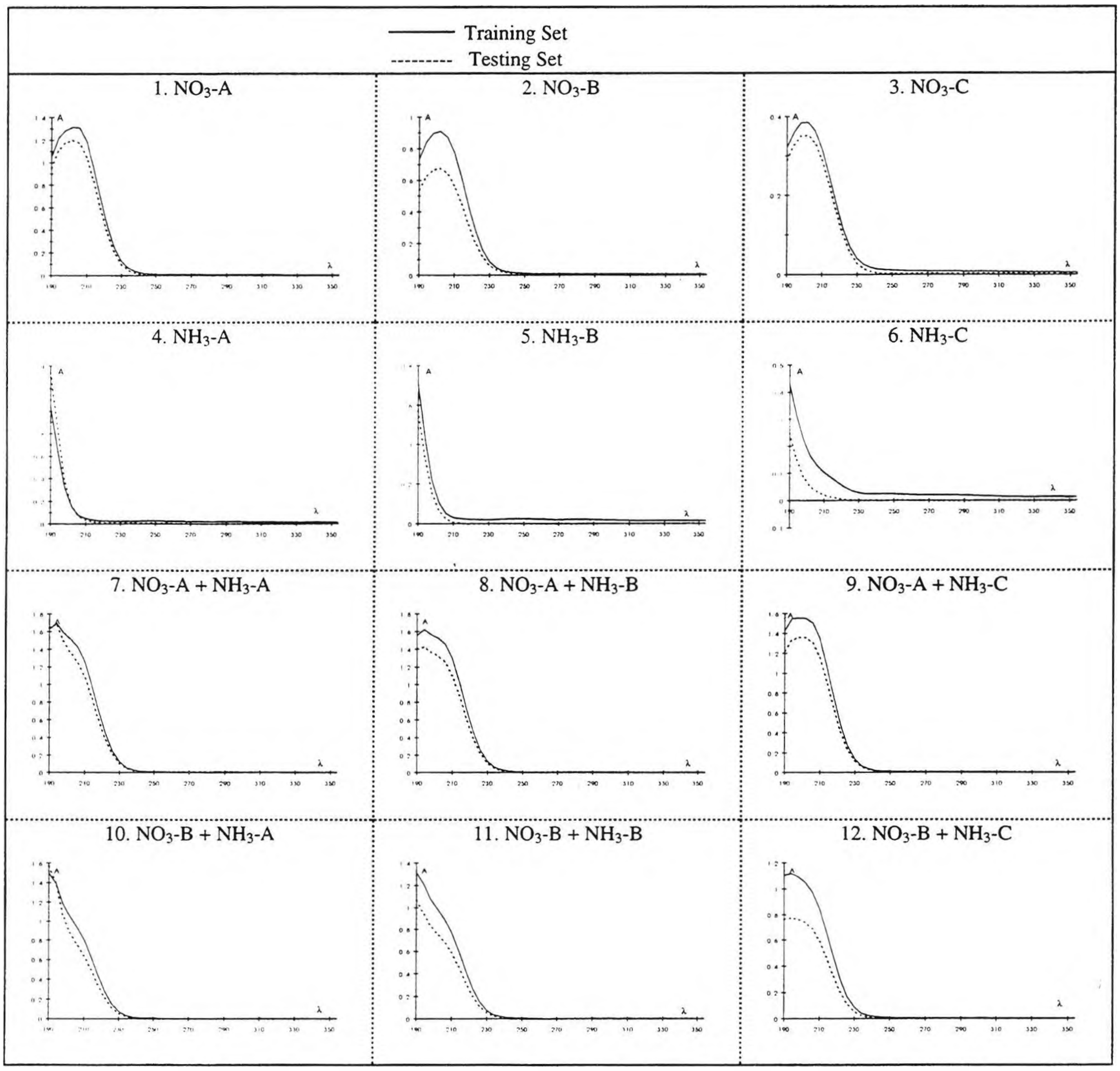
(Cnew is extinction coefficient matrix)

[A] = absorbance
[C] = predicted concentration
[ε] = extinction coefficient
l = path length constant = 1 cm

Appendix F

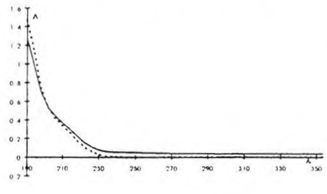
Training and Testing Set Comparison

The comparison between the training and testing set was used to evaluate the performance of two analysis methods (see Chapter 5 and 6).

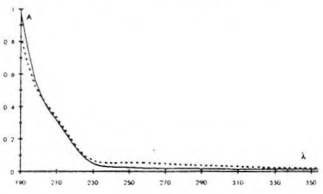


1. NO_3-A
2. NO_3-B
3. NO_3-C
4. NH_3-A
5. NH_3-B
6. NH_3-C
7. NO_3-A + NH_3-A
8. NO_3-A + NH_3-B
9. NO_3-A + NH_3-C
10. NO_3-B + NH_3-A
11. NO_3-B + NH_3-B
12. NO_3-B + NH_3-C

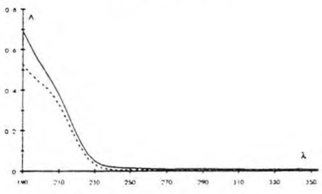
13. NO_3-C + NH_3-A



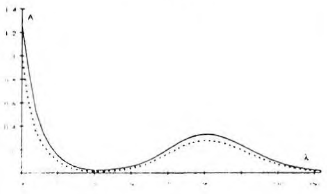
14. NO_3-C + NH_3-B



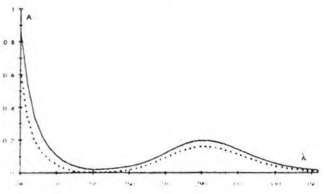
15. NO_3-C + NH_3-C



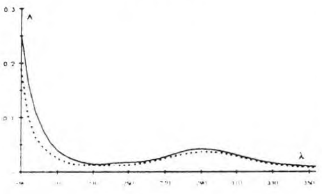
16. Cl_2-A



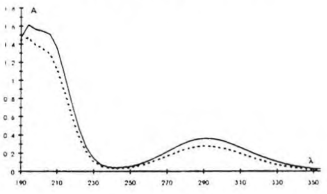
17. Cl_2-B



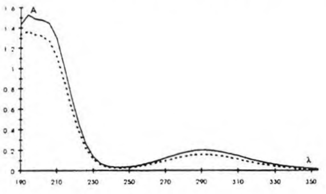
18. Cl_2-C



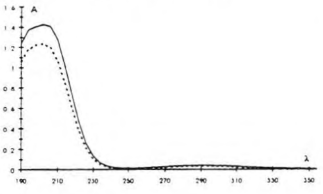
19. NO_3-A + Cl_2-A



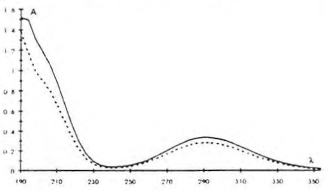
20. NO_3-A + Cl_2-B



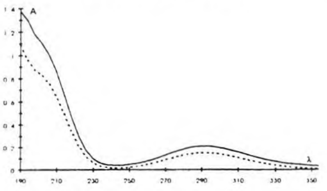
21. NO_3-A + Cl_2-C



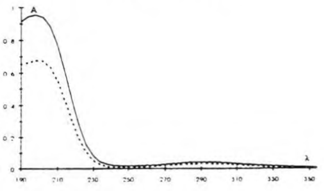
22. NO_3-B + Cl_2-A



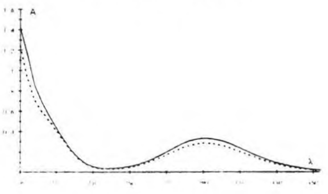
23. NO_3-B + Cl_2-B



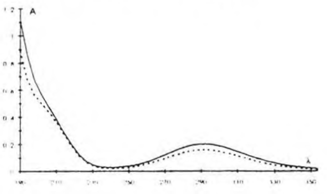
24. NO_3-B + Cl_2-C



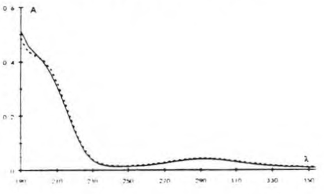
25. NO_3-C + Cl_2-A



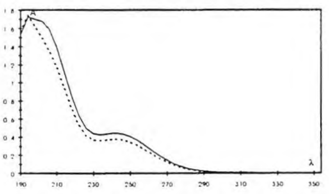
26. NO_3-C + Cl_2-B



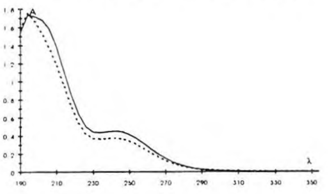
27. NO_3-C + Cl_2-C



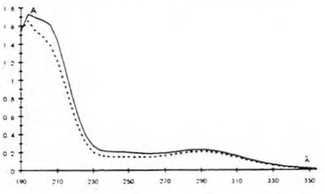
28. NO_3-A + Cl_2-A + NH_3-A



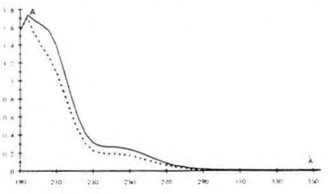
29. NO_3-A + Cl_2-A + NH_3-B



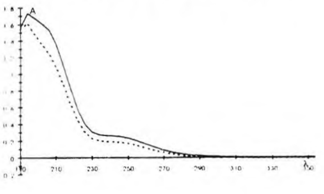
30. NO_3-A + Cl_2-A + NH_3-C



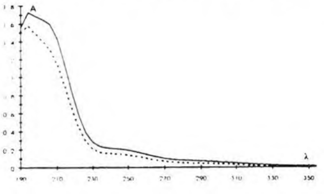
31. NO_3-A + Cl_2-B + NH_3-A



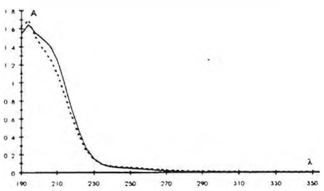
32. NO_3-A + Cl_2-B + NH_3-B



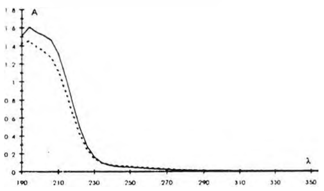
33. NO_3-A + Cl_2-B + NH_3-C



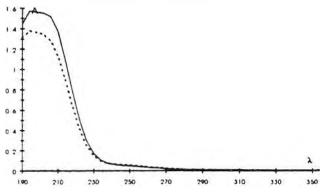
34. NO_3-A + Cl_2-C + NH_3-A



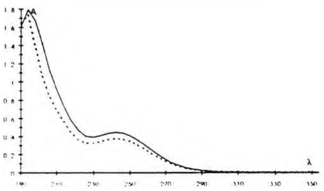
35. NO_3-A + Cl_2-C + NH_3-B



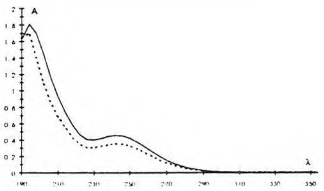
36. NO_3-A + Cl_2-C + NH_3-C



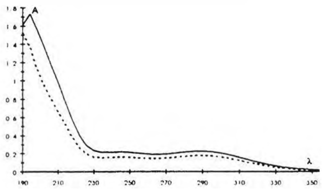
37. NO_3-B + Cl_2-A + NH_3-A



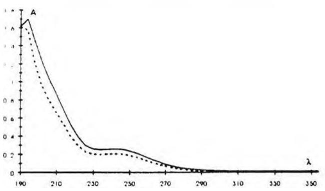
38. NO_3-B + Cl_2-A + NH_3-B



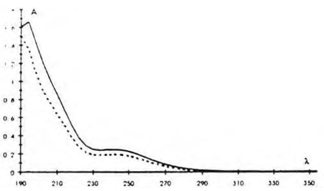
39. NO_3-B + Cl_2-A + NH_3-C



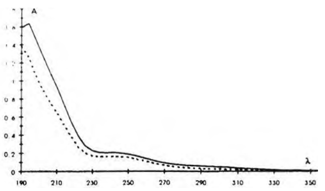
40. NO_3-B + Cl_2-B + NH_3-A



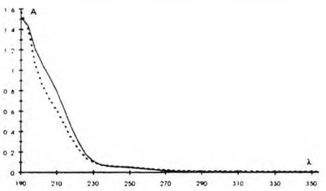
41. NO_3-B + Cl_2-B + NH_3-B



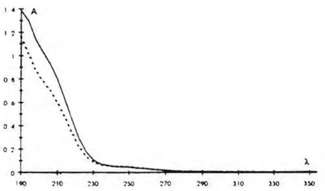
42. NO_3-B + Cl_2-B + NH_3-C



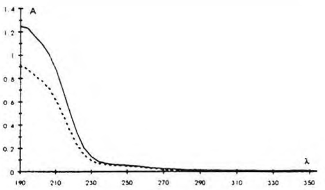
43. NO_3-B + Cl_2-C + NH_3-A



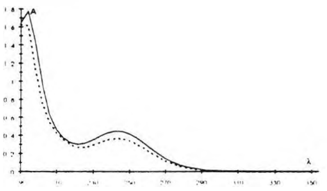
44. NO_3-B + Cl_2-C + NH_3-B



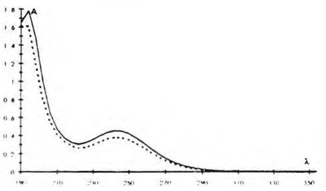
45. NO_3-B + Cl_2-C + NH_3-C



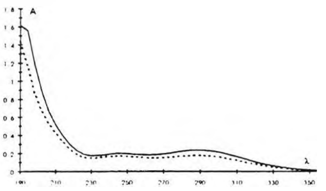
46. NO_3-C + Cl_2-A + NH_3-A



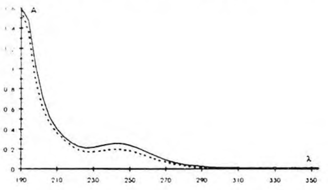
47. NO_3-C + Cl_2-A + NH_3-B



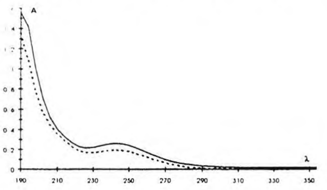
48. NO_3-C + Cl_2-A + NH_3-C



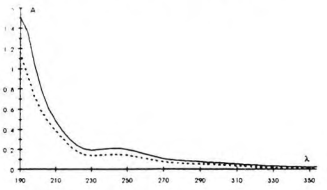
49. NO_3-C + Cl_2-B + NH_3-A



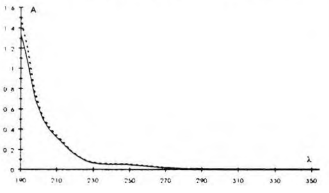
50. NO_3-C + Cl_2-B + NH_3-B



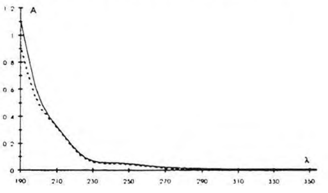
51. NO_3-C + Cl_2-B + NH_3-C



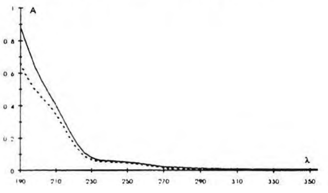
52. NO_3-C + Cl_2-C + NH_3-A



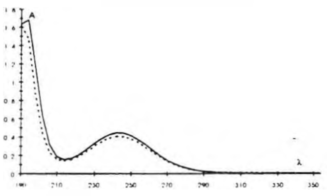
53. NO_3-C + Cl_2-C + NH_3-B



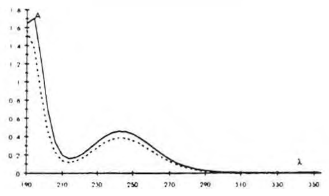
54. NO_3-C + Cl_2-C + NH_3-C



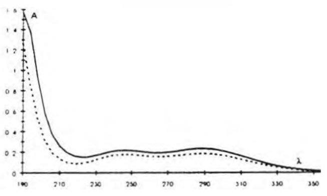
55. Cl_2-A + NH_3-A



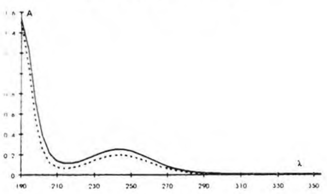
56. Cl_2-A + NH_3-B



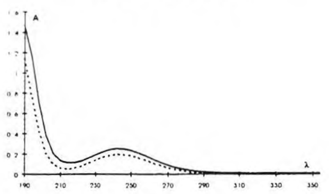
57. Cl_2-A + NH_3-C



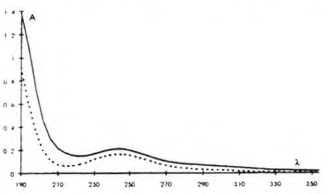
58. Cl_2-B + NH_3-A



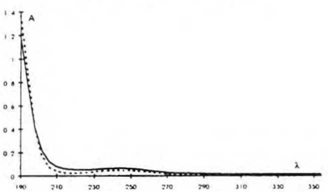
59. Cl_2-B + NH_3-B



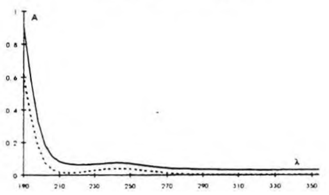
60. Cl_2-B + NH_3-C



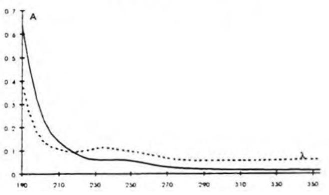
61. Cl_2-C + NH_3-A



62. Cl_2-C + NH_3-B



63. Cl_2-C + NH_3-C



Appendix G

Neural Net Algorithms Supplied with NeuDesk

G.1 Standard Back Propagation

The fundamental concept behind Back Propagation(Back Prop) is that of gradient descent down the walls of an error surface. In Back Prop, the steps taken down this surface are proportion and opposite to the gradient of the surface. Thus, if the walls are steep, large steps are taken; if they are gently sloping, smaller steps are taken.

Since the error surface represents the network error for all the combinations of weights, movement on the surface implies a modification of the weights as follows:

$$W_{change(t)} = -\eta\left(\frac{dE}{dW}\right)_t + \alpha W_{change(t-1)} \tag{G.1}$$

η is learning rate and α is a constant specifying the influence of the momentum. The last term is a momentum term that includes a proportion of the last weight update in the current one. This has the effect of preventing thrashing due to ripples in the error surface. Training proceeds much faster with this term.

$\frac{dE}{dW}$ is the rate of change of error with respect to weight for a particular weight. Standard Back Prop differs from Stochastic by the point at which the weight changes are made. With standard back prop the $\frac{dE}{dW}$'s are summed for every pattern in an epoch and then the weight changes are made based on this sum. Both η and α are limited between 0-1.0. For modelling applications smaller values of η may be more appropriate.

G.2 Stochastic Back Propagation

Stochastic Back Prop differs from Standard Back Prop by the point at which the weight changes are made. With Stochastic Back Prop, the weight changes are made after each pattern presentation. Because the order of the patterns may influence learning, the patterns are presented in random error.

G.3 Quick Propagation

One method to speed up the learning is to use information about the curvature of the error surface. This requires the computation of the second order derivatives of the error function. Quickprop assumes the error surface to be locally quadratic and attempts to jump in one step from the current position directly into the minimum of the parabola.

Quickprop computes the derivatives in the direction of each weight. After computing the first gradient with regular backpropagation, a direct step to the error minimum is attempted by

$$\Delta(t+1)w_{ij} = \frac{S(t+1)}{S(t) - S(t+1)}\Delta(t)w_{ij} \tag{G.2}$$

where: w_{ij} weight between units i and j,

$\Delta(t+1)$ actual weight change,

$S(t+1)$ partial derivative of the error function by w_{ij} and

$S(t)$ the last partial derivative.

G.4 Weigend Weight Eliminator

This algorithm attempts to decrease the sum squared error and decrease the values of less important weights simultaneously during training. This one is of greater importance as training progresses. One of the tenets of neural net faith is that smaller nets optimise better, and one method of achieving this is to push unwanted weights towards zero, thereby effectively eliminating them. The algorithm is similar to Stochastic Back Prop apart from the weight reducing function. The inventors claim that this algorithm will learn associations too difficult for unaided back-prop, and generally perform much better owing to the tighter representation the ‘pruning’ has produced.