Bacteria Classification with an Electronic Nose Employing Artificial Neural Networks

Mark Antony Craven

A dissertation submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

University of Warwick
Department of Engineering

June 1997

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank the following:

Dr. J. W. Gardner, Department of Engineering, University of Warwick, for his guidance, advice and encouragement throughout this PhD.

Mr D. Morgan, ENT Department, Heartlands Hospital, Birmingham, for his initiative and vision in getting the project started and providing resources.

Dr C. Dow, Department of Biological Sciences, University of Warwick, for providing the environment for the data collection experiments.

Frank Courtney, Department of Engineering, University of Warwick, for his assistance in the construction of the instrumentation.

Engineering and Physical Sciences Research Council (EPSRC) for providing financial support for this PhD.

Finally (and certainly not least) I would like to thank Sedef Imer (my partner), family, friends, and fellow research students for their support and encouragement during my time of study.

Declaration

This thesis is presented in accordance with the regulations for the degree of doctor of philosophy. All work reported has been carried out by the author unless otherwise stated, including the production of this document.

Glossary of Terms

Antibiotic An antimicrobial agent produced naturally by a bacterium or fungus, see **Antimicrobial agent**.

Antimicrobial agent A chemical that destroys pathogens without damaging body tissues.

Bacteria All living organisms with procaryotic cells, i.e. cells whose genetic material is not enclosed within a nuclear membrane.

Commensal Symbiont organism that lives without causing harm to its host, see **Commensalism** and **Symbiont**.

Commensalism A system of interaction in which two organisms live in association where one organism benefits and the other organism neither benefits nor is damaged.

Electronic Nose Instrumentation that used an array of solid-state gas sensors (where each sensor has overlapping sensitivity) coupled with a suitable pattern recognition sub-system.

ENT Ear, nose and throat. Common medical term.

Epithelium Area in the roof of the nasal cavity where olfactory receptor cells exists.

Flora The microbial population of an area in/on an animal, such as the throat.

Genus The first of the two names used to specify bacteria. It is always capitalised.

Microbial Consisting of microorganisms.

Micro-organism A living organism too small to be seen without visual aid. Microorganisms include bacteria, fungi, protozoans, microscopic algae and viruses.

Mutual Symbiont organism that lives whilst helping its host and being helped by the host, see **Mutualism** and **Symbiont**.

Mutualism A system of interaction in which two organisms live in association where both organisms benefit.

Neuron Elementary processing unit, a network of which constitutes a neural network.

Normal microbiota See **Normal flora**.

Normal flora Microorganisms that colonise an animal without causing disease, see **Flora**.

Odorant A molecule that has particular properties that allow it to be detected by an olfactory system.

Parasite Symbiont organism that lives whilst causing harm to its host, see **Parasitism** and **Symbiont**.

Parasitism A system of interaction in which two organisms live in association where one organism benefits and the other organism neither is harmed.

Pathogen A disease causing organism which can be bacteria.

Pattern Recognition A term that is used to describe the data analysis performed on datasets. It includes pre-processing and classification.

Resident flora See **Normal flora**.

Seeding Introduction of organisms from one growth environment to another. Often this term is used when bacteria from specimens are introduced to growth media.

Species The specific type of a bacteria which is indicated by the nomenclature which comprises of two parts, for example ***Staphylococcus aureus*** refers to a specific bacteria species, see **Genus** and **Specific epithet**.

Specific epithet The second of the two names used to specify bacteria. It is never capitalised.

Swab Pad of surgical wool, usually attached to the end of a thin rod.

Symbiont An organism living in a symbiotic relationship, see **Symbiotic**.

Symbiotic The relationship between two organisms when they live together, i.e. share the same environment.

Transient flora Microorganisms that are present on an animal for a short time without causing a disease, see **Flora**.

Summary

This PhD thesis describes research for a medical application of electronic nose technology. There is a need at present for early detection of bacterial infection in order to improve treatment. At present, the clinical methods used to detect and classify bacteria types (usually using samples of infected matter taken from patients) can take up to two or three days. Many experienced medical staff, who treat bacterial infections, are able to recognise some types of bacteria from their odours. Identification of pathogens (i.e. bacteria responsible for disease) from their odours using an electronic nose could provide a rapid measurement and therefore early treatment. This research project used existing sensor technology in the form of an electronic nose in conjunction with data pre-processing and classification methods to classify up to four bacteria types from their odours. Research was performed mostly in the area of signal conditioning, data preprocessing and classification. A major area of interest was the use of artificial neural networks classifiers. There were three main objectives. First, to classify successfully a small range of bacteria types. Second, to identify issues relating to bacteria odour that affect the ability of an artificially intelligent system to classify bacteria from odour alone. And third, to establish optimal signal conditioning, data pre-processing and classification methods.

The Electronic Nose consisted of a gas sensor array with temperature and humidity sensors, signal conditioning circuits, and gas flow apparatus. The bacteria odour was analysed using an automated sampling system, which used computer software to direct gas flow through one of several vessels (which were used to contain the odour samples), into the Electronic Nose. The electrical resistance of the odour sensors were monitored and output as electronic signals to a computer. The purpose of the automated sampling

system was to improve repeatability and reduce human error. Further improvement of the Electronic Nose were implemented as a temperature control system which controlled the ambient gas temperature, and a new gas sensor chamber which incorporated improved gas flow.

The odour data were collected and stored as numerical values within data files in the computer system. Once the data were stored in a non-volatile manner various classification experiments were performed. Comparisons were made and conclusions were drawn from the performance of various data pre-processing and classification methods. Classification methods employed included artificial neural networks, discriminant function analysis and multi-variate linear regression. For classifying one from four types, the best accuracy achieved was 92.78%. This was achieved using a growth phase compensated multiple layer perceptron. For identifying a single bacteria type from a mixture of two different types, the best accuracy was 96.30%. This was achieved using a standard multiple layer perceptron.

Classification of bacteria odours is a typical 'real world' application of the kind that electronic noses will have to be applied to if this technology is to be successful. The methods and principles researched here are one step towards the goal of introducing artificially intelligent sensor systems into everyday use. The results are promising and showed that it is feasible to used Electronic Nose technology in this application and that with further development useful products could be developed. The conclusion from this thesis is that an electronic nose can detect and classify different types of bacteria.

Preface

A brief description of the contents of each chapter is given below along with the amount of content of novel material.

Chapter 1 is background information to the project and discusses, briefly, related biological issues. This chapter contains virtually no original material. However a novel (i.e. the application of new technology) solution is outlined for this research project. The possible role of an Electronic Nose in current clinical practice is proposed.

Chapter 2 is a review of previous research performed in the area of Electronic Noses and bacteria classification using odour signatures. This chapter contains no original material.

Chapter 3 details improvements made to the Electronic Nose. The design and testing of a new and novel odour delivery system is documented. Characterisation, testing and experimental procedure development are described. The initial data collection experiments, which analysed two different bacteria types, are documented. Larger amounts of electronic nose data were collected, for each odour class, than has previously been documented.

Chapter 4 details the techniques and algorithms used for initial exploration of the data collected in the experiments detailed in chapter 3. Both old and new data pre-processing methods are investigated. Similarly, both established and novel classification techniques were employed. Comparisons are drawn from the performance of the various methods. The chapter contains a mixture of novel and

established techniques.

Chapter 5 details further modifications to the Electronic Nose. This includes the implementation of new and novel ideas, such as a gas temperature control sub-system and a modified gas sensor chamber. Characterisation, testing and experimental procedure development are described. Also further data collection experiments are documented, which analysed four different bacteria types and mixtures of bacteria types. The analysis of mixtures of bacteria types has not to the author's knowledge, been reported. Also the large amounts of bacteria odour data collected are the largest so far reported.

Chapter 6 details the techniques and algorithms used for data analysis of the data collected in the experiments detailed in the chapter 5. Similar to chapter 4, both old and new pre-processing and classification methods are described. In this chapter the novel use of dynamic information from gas sensor responses are described, the method of extracting the dynamic information, using feature extraction models, is original. A new and original neural net is documented that consists of a growth phase compensation element. Finally, novel analysis of bacteria type mixtures is described.

Chapter 7 contains the discussion and conclusions drawn from the results achieved. Briefly, the major conclusions were: the modifications to the electronic nose were successful and improved final performance, large data sets are necessary for significant classifier research, bacteria growth phase significantly effects performance, weak odours are significantly 'masked' by stronger ones and that the research was successful. Also possible future explorations are discussed.

Publications from this thesis

The following is a list of publications, in chronological order, of publications from this thesis:

M. A. Craven, E. L. Hines, J. W. Gardner, P. Horgan, D. Morgan and I. A. Ene. Bacteria detection using artificial neural networks in conjunction with an electronic nose. *Proceedings of Int. Conf. on Neural Networks and Expert Systems in Medicine and Healthcare*, Plymouth, England, 23-26 August 1994, eds. E. C. Ifeachor and K. G. Rosen, Bluestone Design Ltd, Plymouth, pp 226-234. (ISBN 0-905227 301)

M. A. Craven, E. L. Hines, J. W. Gardner, P. Horgan, D. Morgan and I. A. Ene. Application of an Artificial neural network based electronic nose to the classification of bacteria. *Proceedings of 2nd Int. Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, 20-23 September, 1994, pp 768-774.

M. A. Craven, J. W. Gardner and P. N. Bartlett. Electronic noses - developments and future prospects. *Trends in Analytical Chemistry*,Vol 15, No. 9, 1996, pp 486-493.

J. W. Gardner, M. A. Craven and E. L. Hines. Classification of bacteria age and type using an array of metal oxide sensors and pattern recognition techniques. *3rd Int. Symposium on Olfaction and Electronic Noses* , Miami, US, 3-6 November 1996

E. L. Hines, J. W. Gardner, F. Molinier, M. A. Craven and P. N. Bartlett. Can we mimic human sensory systems? *Royal College of Anaesthetists, Symposium*, London, England, 7-8 November 1996.

M. A. Craven and J. W. Gardner. Rapid Static Headspace Sampler for Automated Odour Analysis. *Measurement and Control*, Special Issue, 8 May 1997.

Chapter 1

Clinical Background

This research project was conceived by an ENT (Ears, Nose and Throat) consultant who identified a problem in the current methods of diagnosing infections. Today the treatment of infection of the upper respiratory system involves taking a sample (or specimen) of infected matter from a patient usually by means of a swab (used for collection of material from mucus membranes, e.g. lining of the throat) or screw-capped jar (used for collection of sputum resulting from a deep cough). The sample is then sent to a micro-biology laboratory for analysis. The patient is unable to receive effective treatment before the analysis is completed, which can take up to 3 or 4 days. Once the sample has been analysed and the bacteria types that are responsible for the infection (pathogens) identified, the patient can begin to receive treatment in the form of antibiotics (antimicrobial agents) that are targeted to kill the pathogens.

There is a need for a system that will allow early detection of bacteria types. It is proposed that this system should be based on electronic nose technology and involves the analysis of odours from specimens taken from patients suspected of suffering from an infectious disease. Subsequently the system will identify if there are any abnormal bacteria present, and if so, whether they are the cause of the disease (i.e. pathogens). This system does not necessarily need to be as accurate as the standard clinical techniques, however, it does need to be accurate enough to enable some early targeting of antibiotics[1]. Therefore this system is not seen as a replacement for existing techniques

[1]Some antibiotics can be targeted to a genus of bacteria types rather than a specific species

of bacteria classification (or any other treatment technique), but rather as a system that would complement and enhance existing practices in this area. For example, this system could be installed either in an operating theatre or in a consultation room (i.e. the location of the ENT consultant) rather than in a microbiology laboratory. The specimens of infected matter could be analysed by this system within minutes, at the point of diagnosis and preliminary treatment can be started. Samples could then be sent on to the laboratory as usual, and at a later date when a more detailed report is available a better targeted antimicrobial agent can be administered.

The concept of using smell to identify bacteria is not new, however, it is yet to be characterised and quantified. The experience of various members of staff in the medical profession, over a number of years, leads them to be able to recognise certain bacteria types by their characteristic odour which can originate from a variety of sources such as puss from infected wounds or swabs containing infected matter. Indeed, in the past (before micro-biological methods were developed), the sense of smell of a nurse or doctor was the primary method used to determine the nature of the infection. There are many difficulties that arise from using odour to classify as the bacteria, these are:

- Given identical circumstances, a particular bacteria type may not produce a unique odour signature but more probably a range of similar odour signatures. This may reflect the chaotic processes that occur within the bacteria cell itself.

- The effect of a single environmental parameter, such as growth medium pH, upon the odour signature may not be sufficient for prediction because other environmental parameters, such as growth medium temperature, may influence this relationship. In other words, there may be a complex non-linear combinative relationship between the odour signature and all environmental parameters. The conclusion from this is that these parameters may need to be precisely controlled which would in turn lead to the need for more complex instrumentation, more demanding operating procedures, and ultimately less reliability.

- When a mixture of different bacteria types (and/or genera) is smelt, the odour signature of a pathogen may be 'masked' by the odour of the other bacteria

types. Since most specimens have several bacteria species present, identifying the pathogen could prove to be problematic. However it is often the case that the pathogenic organisms are much greater in number than the other bacteria (hence the onset of disease in the first place) so that it is not uncommon for the pathogenic organisms to mask out the odour of other background bacteria.

- When a specimen of infected matter is collected using, for example, a swab, from the moment the sample is collected the relative population sizes of any bacteria species present may start to change. This change is due to the different environmental conditions that exist on a swab (as opposed to those, for example, on the lining of the throat). Therefore the characteristic smell of a particular mixture of pathogen when 'smelt' either on the patient or on a fresh sample of infected matter would not necessarily be recognised as the same smell by the electronic nose some time later. It may be possible for a bacteria species that was the pathogen in the original infection to die out and a previously small population of another bacteria species start to dominate.

The proposed system comprises of an electronic nose together with suitable pattern classification algorithms which 'smells' the bacteria odours and classifies the resultant odour signatures as belonging to particular bacteria species (or larger groupings such as genus). This approach would formalise the activity of bacteria classification by means of odour and produce both a useful and worthwhile system.

In order to design and test such a system it is necessary to gain a greater understanding of the problem. Questions such "as what bacteria types are we interested in?" and "what products do they manufacture that contribute to their odour?", need to be resolved. The study of bacteria growth, bacteria metabolism (the products of which cause the odours in the first place), and the factors that cause pathogenic microbes to grow are consequently of great interest. Therefore this chapter will proceed to discuss the relevant issues of bacterial infections of the upper respiratory system in order to ascertain what bacteria types will be encountered. Bacteria growth will be discussed subsequently highlighting its effect on odours.

1.1 Bacterial Infection

When gaining an understanding of bacterial infections of the human body and more specifically the upper respiratory system, it is first necessary to take a look at how micro-organisms and humans interact.

1.1.1 Interaction Between Humans and Micro-organisms

The human body is constantly in contact with micro-organisms, some of which are bacteria. Indigenous microbial flora (many different types of micro-organisms co-existing in an equilibrium) can be found on most parts of the body that come into contact with the outside world, including the upper respiratory system. Normally flora do not pose any great threat to the health of the host and can sometimes be of benefit to the host. Knowledge of how flora are composed and distributed is necessary for the analysis of specimens of infected matter and for the interpretation of any results from laboratory analysis.

Micro-organisms that exist on the surface of the skin and internally for a significant length of time (at least several years) without normally causing disease are called *normal flora*, *resident flora* or *normal microbiota*. Those flora that exist for a short period of time (days, weeks or months) are called *transient flora*. The composition of a flora is complex and changes dynamically in response to environmental changes, some of which change due to human activities, such as diet, hygiene and exercise. If a person works in an environment where they breath air with high levels of contaminants then this would effect the dynamics of *normal flora* that exist in the respiratory system thus causing changes in the pattern of infections. The specimens that are required to be 'smelt' using an electronic nose therefore tend to have not one but many different species of bacteria present. This point is important because previous applications of electronic nose technology have focused on a single source of odour (the odour may contain many different molecule types, but the source is usually one compound) such as alcohols [4], or mixtures of pure gases [5]. Even applications of electronic nose technology where the odour source has been bacteria, the bacteria concerned has been

mainly (if not exclusively) of one species [6, 7, 8] or sometimes many species in a stable environment [7, 9, 10]. This application therefore requires the resolution of multiple bacteria species (or genera) in the presence of other multiple (and dynamically changing) bacteria species.

The relationship between a *normal flora* and their host can be described as *symbiotic*, where the participant organisms are referred to as *symbionts*. One particular type of symbiotic relationship is termed *commensalism*, meaning that the micro-organisms benefit by the relationship with the host and that the host neither benefits nor is harmed. Examples of *normal flora* microbes which are commensals are those that exist on the surface of the eye. Another type of symbiotic relationship is *mutualism* where both the micro-organisms and the host benefit each other. Examples of *normal flora* microbes that are mutuals are the *Escherichia coli* (often referred to as *E. coli*) bacteria that exist in the large intestine; the host benefits because the bacteria produce vitamin K and some B vitamins as by-products of their metabolism, and in return, the hosts provides an environment where the bacteria can grow and receive nutrients. The third type (and worst as far as the host is concerned) is the symbiotic relationship of *parasitism*. Many pathogens are parasites, for example, *Staphylococcus aureus* (often referred to as *S. aureus*) which grows in the nostrils is normally a commensal, however when this bacteria is breathed into the lungs or introduced to a wound then it can become a pathogen and therefore a parasite.

The odour produced by a bacteria species that is part of a *normal flora* may differ according to the symbiotic status (e.g. parasite) of that organism at the time collection of the specimen. The fact that a particular bacteria species being present in a specimen of infected matter does not necessarily mean that it is the pathogen; the electronic nose will be required to not only identify bacteria species as being present (especially if they are part of the *normal flora* at the site where the specimen was collected) but also the symbiotic status and therefore whether any bacteria species present is a pathogen[2].

[2]Moreover, it is possible for a specimen of infected matter to contain more than one species of bacteria that are pathogens

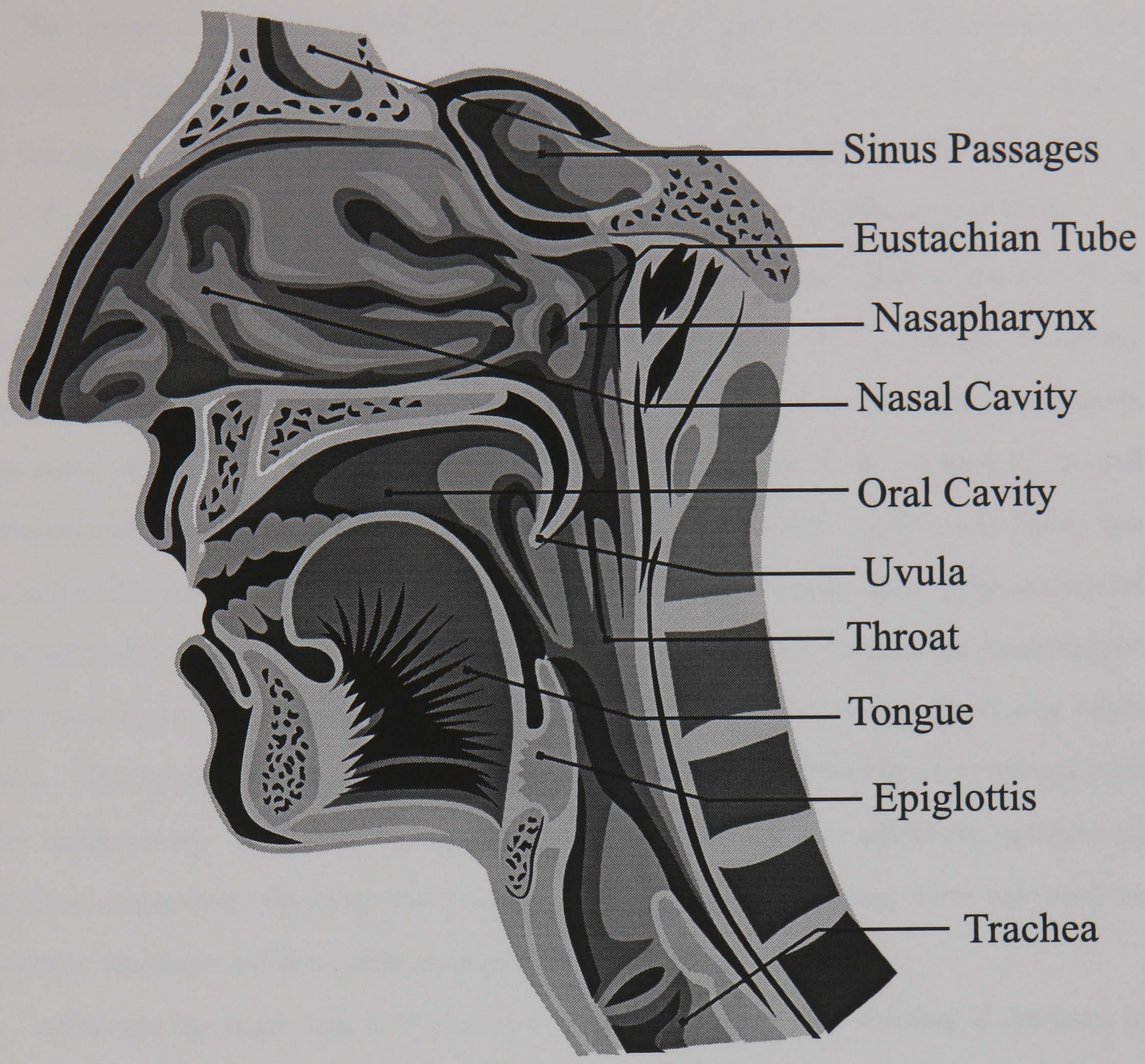


Figure 1.1: A simplified cross-section of the lower head and neck showing the major parts of the upper respiratory system. The ears are not shown for clarity.

1.1.2 Bacterial Infections of the Upper Respiratory System

The most common infectious diseases that afflict mankind, by a large margin, are those of the upper respiratory system. In order to classify infectious bacteria we must first detail what pathogens are common causes of infection, what conditions they grow in and therefore what products are likely to be found in their odours. Because disease of the upper respiratory system is common there are large amounts of reference material on the subject [11, 12]. Figure 1.1 shows the major parts of the upper respiratory system (excluding the ears), it can be observed that the tissues have a complex structure that provide many sites for infection.

The purpose of the upper respiratory system is to regulate the temperature and humidity of air that passes into the lungs during breathing and to filter the air so that no foreign material or microbes pass through. Air reaching the lungs varies only by 2 or 3°C, and particles as small as 1 to 5 μm are filtered with an efficiency of 50%.

Over 11,000 litres of air per day is breathed by a normal and healthy person. This large amount of air flows through the upper respiratory system, contained in this air are vast numbers of air-borne microbes. The air firstly enters the nostrils, then enters the nasal cavity and finally enters the throat; from here (with the epiglottis[3] moved downwards) it enters the lower respiratory system. The nasal cavity exists above the mouth and is partly divided into left and right parts. In this cavity there exist structures of spongy tissue which can become erect due to stimuli such as infections, temperature and mental state (this gives rise to the general feeling of nasal congestion during colds etc.). The sinuses passages, eustachian tubes and tear ducts all have openings into the nasal cavity. Thus the various components of the upper respiratory system are all inter-connected; therefore the possibility for infection to spread from one area to another via these various connections exist.

Although the main aim of this project is aiding antibiotic treatment of diseases, it has to be kept in mind that there are various natural mechanisms for preventing and fighting infection. These mechanisms work by means of lymphoid tissue, the tonsils and the adenoids (which are situated at the junction of the nasal passage with the throat). Oddly enough when these structures become swollen due to infection, they can prevent the drainage of fluid from the middle ear via the eustachian tubes and so increase the risk of infection. Another mechanism for reducing infection is a mucus secreted from cells in the lining of the nasal cavity and middle ear. Tiny hairs (cilia) move the mucus (and with it micro-organisms) from the nasal cavity to the throat and nose where it is expelled from the body. This action results in the ears and sinuses being free from organisms under normal circumstances.

Although the cause of the infection starting can be quite complex, the origin of the pathogen is one of the following:

[3] A muscular fold of tissue which, during swallowing, closes off the wind pipe.

Table 1.1: Typical composition of the indigenous flora of the upper respiratory system.

Species (or genus)	Possible pathogen
Micrococci and staphylococci	*Staphylococcus aureus*
Corynebacterium	*Corynebacterium diphtheriae*
Neisseria	*Neisseria meningitidis*
Haemophilus	*Haemophilus influenzae*
Bacteroides	Strict anaerobes
Streptococcus	*Streptococcus pneumoniae*

1. A micro-organism that was originally part of the *normal flora* for that area of the body.

2. A micro-organism that was originally part of the *normal flora* for another part of the body.

3. A micro-organism that is not part of the *normal flora* for any part of the body.

The bacteria listed in Table 1.1 are pathogens from the first category in the list above. One in three people normally carry the ***Staphylococcus aureus*** in their *normal flora*, which is pathogenic. Therefore the bacteria listed in this table are likely to be encountered by the electronic nose system.

An example of a pathogen from the second category in the above list is ***Escherichia coli***. This bacteria is part of the *normal flora* of the large intestine but when it is introduced into an area such as the ear (this could happen by, for example, a contaminated object being inserted into the ear), disease can result.

An example of a pathogen from the third category is ***Legionella pneumophilia*** which is responsible for Legionnaire's disease. In order to reduce the complexity of the problem to a manageable level, only common pathogens from the first two categories will be considered. It may be possible at a later stage of development to specialise the electronic nose system to isolate a rare, but important, pathogen.

Table 1.2 lists most of the common pathogens. It can be seen from this table that

Table 1.2: Common bacterial infections of the upper respiratory system and their associated bacterial pathogens.

Condition	Cause
Throat infections (pharyngitis-tonsillitis)	*Streptococcus pyogenes*(Strep throat), *Corynebacterium diphtheriae*, *Neisseria gonorrhoeae* and *Mycoplasma pneumoniae*
Ear ache (acute otitis media and serious and mucoid otitis media)	*Streptococcus pneumoniae*, *Streptococcus pyogenes*, *Haemophilus influenzae*, *Branhamella catarrhalis*, *Staphylococcus aureus*, *Staphylococcus epidermidis*, *Escherichia coli*, *Klebsiella pneumoniae* and *Pseudomonas aeruginosa*
Sinusitis	*Haemophilus influenzae*, *Streptococcus pneumoniae*, *Streptococcus*(nonpneumonococcal), *Neisseria*, *Escherichia coli*, *Pseudomonas aeruginosa*, *Bacteroides*, *Fusobacterium*, Rhinoviruses, and *Penicilium*
Acute epliglottitis (inflammation of the epiglottis)	*Haemophilus influenzae*
Diptheria	*Corynebacterium diphtheriae*
Laryngotracheobronchitis	*Bordetella pertussis*, *Streptococcus pneumoniae*

the number of pathogens is large and varied; however, most infections are the result of only a few of these pathogens such as *Staphylococcus aureus*, *Streptococcus pneumoniae* or *Escherichia coli*. Therefore for the system to be effective it is not necessary to be able to classify large numbers of pathogens but rather to be able to classify a small number of pathogens well.

Viral infections have been deliberately left out of this project because by their very nature [4] they are more difficult to identify by a odour signature. However, it is possible for a bacterial pathogen to trigger a dormant virus and therefore cause a secondary infection; it is possible for viral infections to result in super-infections with ***Streptococcus pneumoniae***, ***Haemophilus influenzae***, ***Streptococcus pyogenes***, ***Neisseria meningitidis*** or other pathogens. Studies of the link between viral and bacterial infections is not complete, therefore no definitive arguments can be asserted here. Even if the bacterial pathogen could be identified and classified, the viral agent could not. This system could aid the treatment of the bacterial infection, but the viral infection would be left to the body's own defense systems (which may render the antibiotics ineffectual). The most common viral infection is the common cold for which presently there is no known cure. Further, it may be possible for a virus to change the odour of the bacterial pathogen if the virus also infected that pathogen, thus reprogramming the pathogenic cell to change its behaviour (including how it metabolises); this is outside the scope of this research at present.

It can now be seen that the number of different types of bacteria that would be required to be detected and classified for this application to be a useful tool, is in the order of 5 to 10. This research project was carried out with these aims in mind.

1.2 Biological Aspects Of Bacteria Growth

In order to classify bacteria according to their odorous products we must first investigate the biology of bacteria growth. This will give us some knowledge of the sort of gaseous products that bacteria produce and how their odours might vary due to environmental conditions. More information about the constituents of the bacteria odours and sources of their variance will enable the system design to be better specified.

The biochemistry of micro-organism metabolism is complex. There are two main groups of substances that contribute to make an odour; primary metabolites which are transitionary substances that are 'stepping stones' from one process to another and

[4] Being genetic parasites they invade cells and so do not have a characteristic odour of their own.

can be crossed either way, and secondary metabolites which are end (waste) products that cannot be re-used. Examples of primary metabolites are amino acids, alcohols and aldehydes, and examples of secondary metabolites are toxins and acetic acid. The ratios of the various primary and secondary metabolites changes depending on the state of growth of the micro-organisms.

When a small number of bacteria are inoculated into a liquid growth medium (or from one medium to another), and the population is counted at intervals, a plot of the typical bacterial growth curve that shows the growth of cells overs a period of time can be drawn. The four main phases of growth can be identified from this plot, an example is shown in Figure 1.2. This type of growth curve is typical of what happens when a specimen of infected matter is cultured on growth media. A disease may also follow a similar growth curve, where the onset of infection is defined by the first two phases, the peak of infection is defined by the third phase and the cessation of the disease (either by natural or artificial methods) is defined by the final phase.

It is possible, when a particular specimen is collected, that the majority of the population of the pathogenic cells will be in one of the four growth phases. Therefore, the effect of growth phase upon bacteria odour is important.

The metabolic activities of bacteria cells in each of the four phases are as follows:

Lag phase: Initially there is very little change in the number of cells; this is because bacteria do not reproduce immediately in a new growth medium. This period of little change is called the lag phase. The duration of the lag phase can be anything from one hour to several days partly depending on environmental conditions, such as medium temperature and initial population distribution. The cells are not dormant during this period, there is intense metabolic activity, in particular enzyme synthesis. Towards the end of this phase some cells will double or treble their initial size in order to prepare for reproduction. This period could be described as 'tooling up' for cell division, i.e. growth without cell division where cells are synthesising and storing molecules that are required for the cell division process. The odour from the bacteria in this growth phase is relatively small in magnitude, and undergoes significant change from beginning to end.

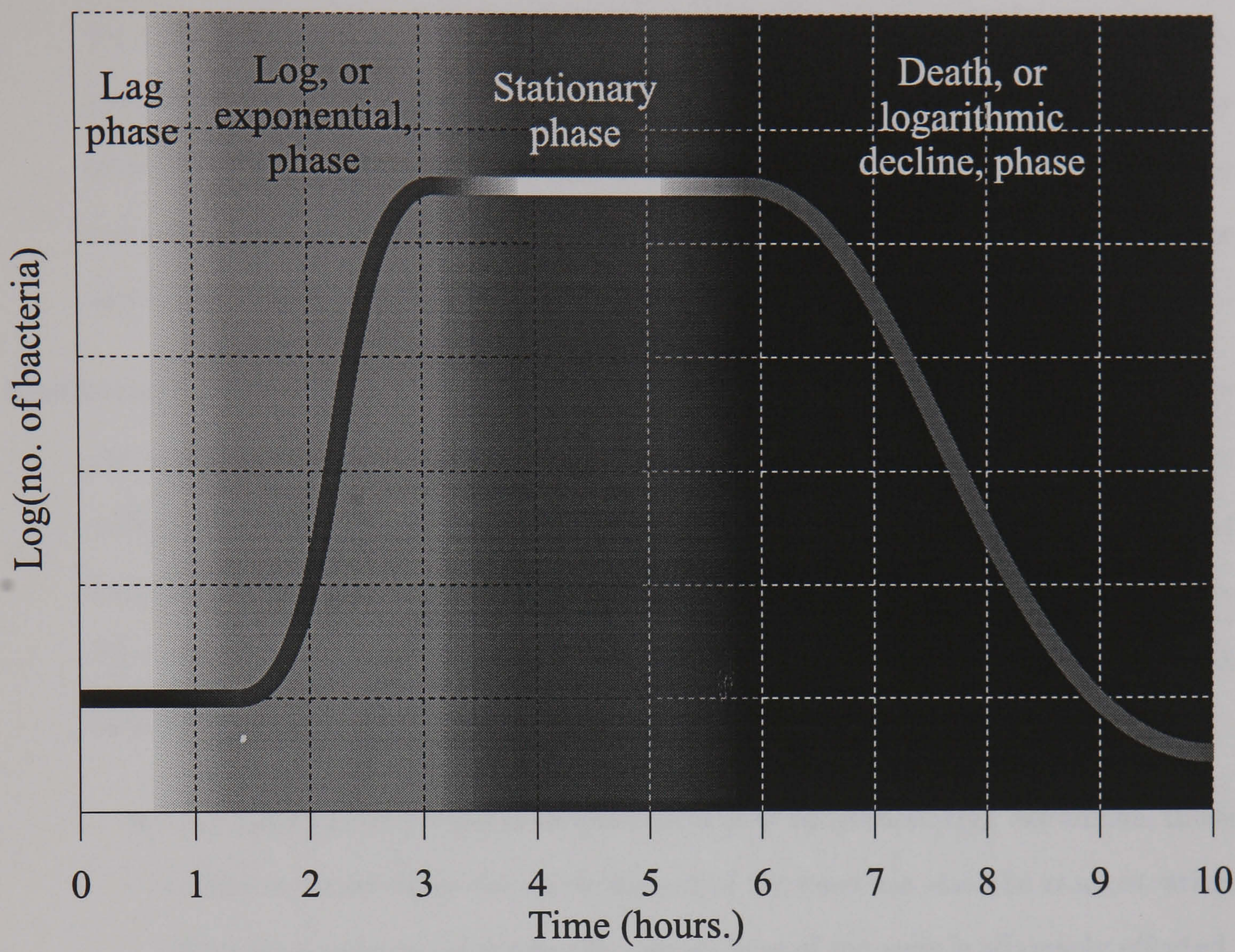


Figure 1.2: Typical growth curve for a population of bacteria showing the four growth phases.

Log phase: Some time later, the bacteria cells start to divide and enter a period of growth. During this phase cells divide at their maximum rate and a logarithmic increase in cell numbers results. This phase is called the log phase (or exponential growth phase). The increase in cell numbers does not occur in discrete steps but rather a continuous increase where at any one point in time cells may be observed at all stages of division. The minimum generation time is achieved here, i.e. the minimum time between divisions for a given cell. Each bacteria type has a characteristic generation time for a given set of environmental conditions. The generation time of many bacteria is as low as 20 minutes, therefore over a period of 10 hours, one cell will multiply to reach a population of 10^9 cells (usually, bacteria populations in most environments, stop growing when their number reaches approximately this amount). Due to their active metabolic state,

the cells are more vulnerable to adverse conditions and therefore are more easily controlled by antimicrobial drugs. The magnitude of the bacteria odour may increase[5] during this phase, the odour components may start to stabilise to the ones normally associated with primary metabolites such as CO_2, ketones, water vapour, alcohols and aldehydes will start to dominate.

Stationary phase: Eventually the rate of growth reduces and a point is reached where the number of cell deaths balances the number of cells that are dividing (viable cells). The cells adapt to their changing environment in such a way that this balance is maintained for a period of time, this is called the stationary phase. The reason why the number of viable cells reduce is not totally clear in every case, it is usually due to a combination of two factors:

- Cell metabolism produces as waste products substances that are toxins, these toxins accumulate in the environment of the bacteria and the concentration of toxins reach a level where the metabolism of the cells is adversely effected.
- The available nutrition is reduced because the demand for nutrition rises as the population increases. The competition for nutrition inevitably leads to some cells receiving less nutrition than is required for division and therefore they fail to become viable cells. The material most commonly exhausted by aerobic organisms is oxygen, and for anaerobic organisms, energy supply substances, such as sugars.

If the growth medium is renewed at a sufficient rate the population can be maintained in the log phase. The majority of the odour components from bacteria in this phase are likely to be primary metabolites, with little change, even when the death phase is entered.

Death phase: Sooner or later the rate of cell deaths rises to a point where cell division can no longer maintain the balance that was present during the stationary phase.

[5]The total odour may reduce if the odour from the growth medium reduces by a greater amount than the bacteria odour increases. This is due to nutrients in the growth medium, which give off an odour, being metabolised by the bacteria.

This phase is called the death phase (or logarithmic decline phase), during which the number of viable cells exponentially decreases. The population is reduced, usually within a few hours, to extinction or to a small number of more resistant cells. Exactly why some cells in the population should become more resistant is not entirely understood, this trait is not passed on to daughter cells.

The odorous gases produced by growing bacteria are highly variable. This is partly because the cellular metabolism that is responsible for the production of gaseous metabolites is itself highly variable due to the changing environmental factors in which the bacteria is growing. Cell metabolism during the four phases of growth (as previously described) clearly changes; for example, the source of energy in the early phases may mostly consist of simple sugars which are easy to break down, but in the later phases the source of food may be mostly the products of other microbes (or other microbes themselves) which may be complex carbohydrates that are more difficult to break down. This is responsible for the change in relative quantities of primary and secondary metabolites.

1.3 Current Clinical Practice for Analysis of Bacteria

This section describes general current clinical bacteriological practices applied to the identification of infectious bacteria of the upper respiratory system originating from specimens taken from a patient. Here we identify problems and strengths of current practices, and having done so identify potential problems for an electronic nose. Furthermore, current problems and their effect on patient care will be examined.

The subject is described by firstly detailing the clinical strategies, then by a functional breakdown of a single clinical investigation, and finally by a description of the different types clinical investigation based on the type of specimen involved. The precise procedure followed for each different type of specimen is too complex to document here and would not be helpful background information to the project. Therefore special emphasis is placed on describing the most time consuming practices (e.g. incubation).

1.3.1 General Strategies Adopted in Clinical Bacteriology

Clinical bacteriology is the study of specimens taken from patients suspected of infectious disease in order to find, firstly, if there is any change in the kind or distribution of the *normal flora* and, secondly, if the abnormal bacteria found are the cause of the disease. In most cases it is fairly easy to answer the first of these questions. The second is often difficult and sometimes impossible to solve; it may be approached in the following way. The question "We have found microbe A, is it causing disease B in this patient?" is asked. We may use the statistical argument that in many previous cases A has been satisfactorily found to be the cause of B; therefore the chance of it being so in this case is very great and the assumption may safely be made. There are many pitfalls in the use of this argument for individual cases because no two patients are exactly alike and there is also wide variation in virulence between strains of the same species of microbe. For example, it has been established beyond reasonable doubt that ***Streptococcus pyogenes*** is the cause for a sore throat, and if we use this argument we shall assume that in all cases of a sore throat when this microbe is found it is the cause of the infection. But this is not so; healthy people are sometimes carriers of ***Streptococcus pyogenes*** and such a person may develop adenovirus infection, in which case if we say that microbe A (***Streptococcus pyogenes***) is the cause of disease B (sore throat) we may be at fault. It follows that clinical practices are not straightforward and that even a case that appears initially simple may turn out, eventually, to be highly complex.

The conflict between the necessarily swift investigation of ENT specimens and the fact that, generally, the more detailed (and reliable) investigations require large amounts of time to perform, lead to two main courses of action. Firstly, scientific integrity may be sacrificed by the investigation only including one or two easily tested strains, and that remaining strains may be guessed by morphological appearances and knowledge of the expected normal flora. Secondly, the investigation may be limited by excluding known pathogens and not attempting to identify the remaining organisms. Of the two the latter is preferable because the former can result in a misleading report which could lead to a worsening of the patient's condition. In practice an investigation may contain elements of these two scenarios and in general there are four

types of investigation which can usefully be performed in a hospital laboratory on ENT specimens:

1. Exclusion of known pathogens.

2. Exclusion of known pathogens plus partial investigation of normal flora.

3. Full bacteriological investigation of clinically rare conditions.

4. Full investigation of all bacteria found in specimens.

As to be expected, the strategies adopted tend to be a compromise between what is an ideal situation (e.g. identification of all micro-organisms) and what is practically possible (e.g exclusion of common pathogens).

1.3.2 Description of the Investigative Processes

There are many clinical strategies and no two patients are identical, this means that there is a unique element to each clinical investigation. However, clinical investigations do tend to follow, overall, the same set of procedural steps. Therefore the investigative process can be broken down into the following broad areas[6]:

- Specimen collection
- Culture of specimens including routine and selective methods
- Visual inspection
- Microscopic examination
- Collation of reports and interpretation of results

This description does not provide a comprehensive account of clinical bacteriology but an introduction for the intended application of electronic nose technology; a more complete description can be found in the many books written on the subject [13, 14]. Figure 1.3 shows a typical laboratory investigation, from this it can be noted that laboratory techniques for culturing specimens is highly skilled and labour intensive.

[6]These areas are not necessarily performed in the order given in this text. For example, macroscopic and microscopic examination may be performed many times during the entire procedure.

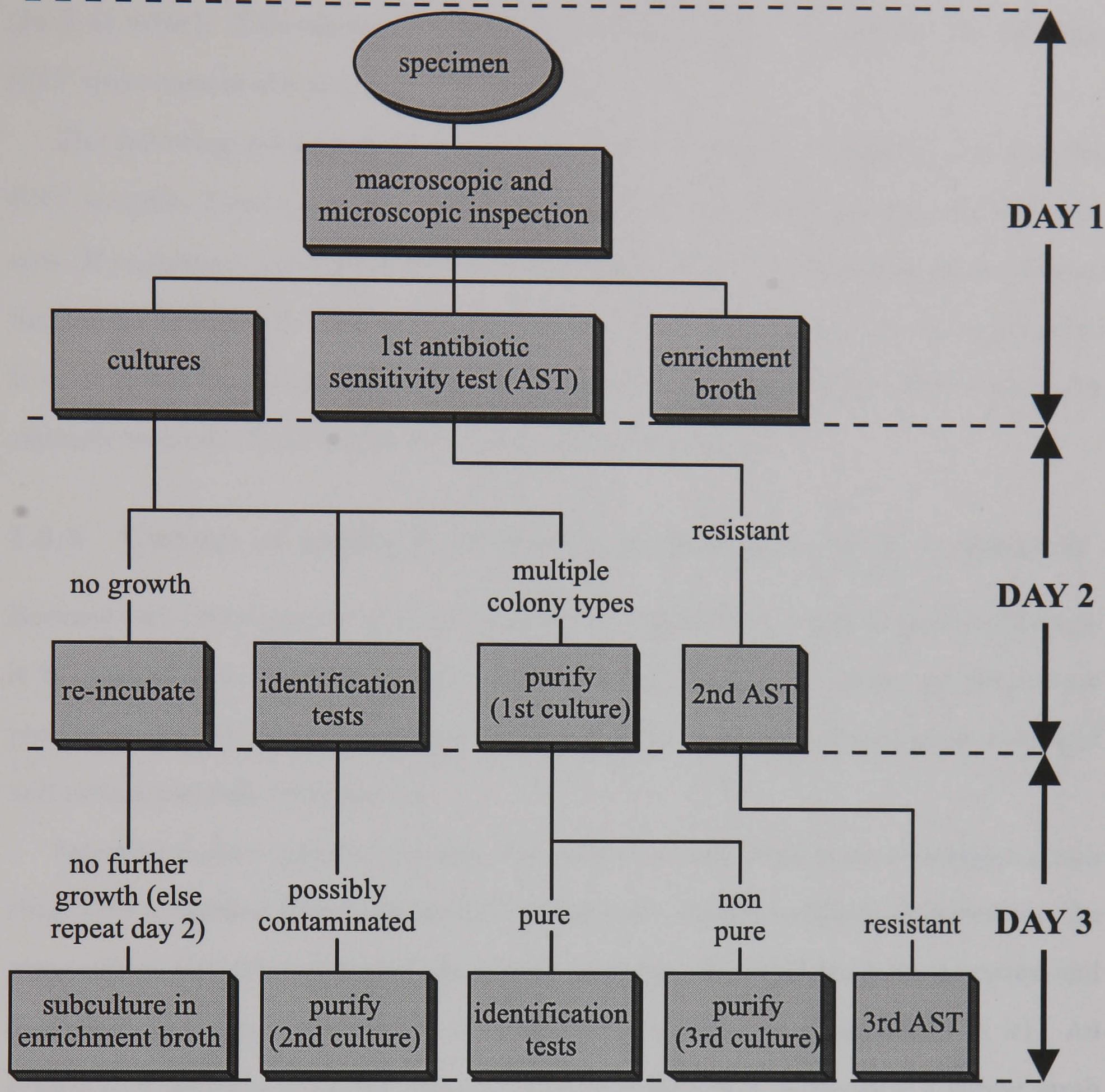


Figure 1.3: A diagramatical representation of typical laboratory procedure, showing the main processes of activity. Note that this diagram is not specific to any particular type of clinical investigation.

Specimen Collection

The results of investigations of infected material and the speed with which they are obtained depend not only on laboratory methods but also on the manner in which the specimens are taken and the promptness with which they are delivered to the laboratory. Generally, specimens from areas of the body that have a *normal flora* (e.g. ENT) can be handled in a less stringent manner than those from areas that are normally sterile

(such as urine). This allows more flexibility in the design of equipment for handling ENT specimens in the electronic nose system.

The following points must be considered when designing equipment that handles ENT samples. Firstly, specimens must never be in contact with antiseptics or disinfectants. If equipment needs to come into contact with such substances it must be cleaned thoroughly afterwards. Secondly, specimens should be analysed as soon as possible. It is desirable for equipment to facilitate prompt usage, long delays before usage, for example because of calibration or testing, are to be avoided.

1.3.3 Culture of specimens including routine and selective methods

Because each investigation develops individually according to the findings at each stage, it is impossible to lay down a strict routine for all specimens. However, the culture process is the one which takes most of the time; the quickest cultures grow overnight and others can take several days.

Selective culture greatly increases the chance of recovering known pathogens from sites with a *normal flora* such as ENT. There are several methods of selection: the atmosphere, the composition of the growth medium, the incubation temperature and antibiotics spread on the surface of solid growth media (or incorporated in it). An example of selective culturing using atmospheric factors is incubating a throat swab specimen anaerobically. The *normal flora* present grow mainly aerobically and are therefore inhibited but haemolytic streptococci grows at an increased rate and can easily be identified. The electronic nose, in order to operate under anaerobic conditions would not be able to use laboratory air as a carrier gas and would probably have to use an inert gas.

There are a wide variety of selective growth media available. The incubation period for some elective media is at least 48 hours because the lag phase of some or all of the bacteria may be lengthened by the media and may last for up to 24 hours. Therefore selective culturing can have the effect of lengthening the investigation. Selective culturing using atmospheric temperature is not common because ENT bacteria (including pathogens) grow at (or very near) body temperature and therefore they all tend to be

able grow at the same temperature. The main use of temperature election is for other 'low-temperature' bacteria such as the listeria bacteria group that can grow in food kept in a refrigerator; this is not directly of relevance to ENT disease.

Antibiotics are common in use for selective culturing. A common practice is to divide a Petri Dish, containing a solid growth medium, such as blood agar, into two or four equal segments where each segment contains an antibiotic[7] and any subsequent colonies can be observed to grow only in one ore more segments.

Visual Inspection

Visual inspection is used at various stages in the investigation. Firstly, specimens are inspected for overall condition, if a specimen is contaminated with an antibiotic or is badly taken (e.g. contaminated, or taken from the wrong site) then a new specimen may be necessary. The plate cultures are examined by macroscopic inspection (visual inspection) in order to ascertain the appearance of the colonies. Changes in normal flora or the existence of pathogens can often be indicated by the appearance of visually distinctive colonies. Overall the results of the investigation are based on the observation (or non-observation) of colonies the types of which are identified by their morphology.

Microscopic Examination

Microscopic examination of stained films or of a wet preparation from the specimen should precede culture. The purpose of the examination is twofold; first it is a guide to further procedures (e.g. if fungi are observed then this can be cultured as well in special media) and second it is the most valuable indication of the proportion of different species in the specimen. The most popular method of staining is the Gram stain and one of the most common groupings of micro-organisms are Gram +ve or Gram -ve. Microscopic examination is also a useful tool after incubation of colonies; the precise method of staining is dependent on the type of specimen and the type of investigation being performed.

[7] One segment is usually left untreated with antibiotics in order to act as a control.

Collation of Reports and Interpretation of Results

Each laboratory tends to have its own standard for the structure of reports, however there are strict standards that have to be observed. Usually each specimen is labeled with a unique number and a corresponding record kept detailing the tests performed and the results observed. The results are not interpreted to any great degree by the laboratory staff, often results will be summarised in the report but the ultimate interpretation will be left to the physician concerned.

1.4 Common ENT Specimen Types/Sites and Investigations

A useful way to categorize ENT clinical investigations is to classify them according to the site from which the specimen was taken.

1.4.1 Throat Swabs

The lining of the throat is sampled by wiping with a cotton swab. This swab is then placed into a protective container, is then labeled and sent to the laboratory for analysis.

The most common cause of upper respiratory infection in Britain is ***Streptococcus pyogenes***, therefore a common procedure is to exclude this pathogen from the specimen. The swab is seeded on a blood agar plate and incubated in an anaerobic environment, this selects haemolytic streptococci from the rest of the throat flora. If the incubation was performed in air, ***Streptococcus pyogenes*** might be missed amongst colonies of other throat commensals. Other common pathogens are affected by an anaerobic environment; ***Streptococcus pneumoniae*** grow well (at least as good as in air), and ***Haemophilus influenzae*** grows less well (often 10% CO_2 is added in order to encourage growth). The minimum time for this procedure is one day; if the visual inspection of colonies is not conclusive then further sub-culturing may be necessary which can take one or more further days. The plates are examined firstly using reflected light and some magnifying apparatus (probably a hand lens) and then using transmitted light.

When a patient is suspected of suffering from tonsillitis then the presence of ***Coryne-***

bacterium diphtheriae is usually investigated. The swab is seeded on blood agar and blood tellurite agar. The tellurite selects *Corynebacteria* from other throat flora. These plates need to be incubated for up to 48 hours. The blood agar plate is incubated in air and the tellurite plate anaerobically. The blood agar plate is visually inspected for the amount of growth. Stained slides can be made from both plates for identification of organisms. Tellurite inhibits all bacterial growth but *Corynebacteria* is least affected[8]. The blood agar serves two purposes; firstly it shows up any haemolytic streptococci which can exhibit similar symptoms and secondly it serves as a check to ensure that the sample was taken correctly. Various smears and stains are performed on the colonies formed on the tellurite medium in order to aid observation.

Other pathogens found in throat swabs are Haemolytic streptococci other than *Streptococcus pyogenes*, *Streptococcus pneumoniae*, *Haemophilus* and *Staphylococcus aureus*. These organisms, if found, are usually discovered whilst investigations for the more common pathogens are being performed. When this is the cause a second investigation is started which can be performed simultaneously with the original one. Also *Staphylococcus aureus* is not commonly found on throat swabs and rarely causes a sore throat (more likely to cause swelling).

1.4.2 Nasal Swabs

The lining of the nasal cavity is sampled in much the same manner as that for the throat. The major complaint of infection of the nasal passage is rhinitis which is inflammation of the mucus membrane. Three cultures are made from the swab; the first is seeded on blood agar in air plus 5% CO_2, the second is seeded on blood agar which is incubated anaerobically and the third is seeded on blood tellurite medium. The purpose of the first culture is to investigate the presence of *Neisseria meningitidis*, *Staphylococcus aureus* and haemophilus. The purpose of the second culture is to investigate the presence of *Streptococcus pneumoniae* and haemolytic streptococci. And the purpose the third culture is to investigate the presence of *Corynebacterium diphtheriae*. It is possible for

[8]It also has the helpful property of showing up the three major types of *Corynebacteria diphtheriae* although this is not always necessary.

Staphylococcus aureus to be present in the nose of a healthy person therefore detection does not necessarily mean that it is the pathogen.

1.4.3 Nasopharyngeal Swabs and Aspirated Muco-Pus

There are two main reasons for taking samples of matter by means of nasopharyngeal swabs; firstly for detection of carriers of meningococcal meningitus which involves isolating ***Neisseria meningitidis*** and secondly for suspected whooping cough which involves isolating ***Bordetella pertussis***. For adults, a special swab is employed called a 'West's Swab'[9] is used and is seeded on blood agar and incubated in air plus 10% CO_2. For children, a standard swab is used which is seeded on selective growth media in order to suppress the growth of the normal flora. After incubation, which usually takes one day, the organisms present are detected in the normal manner (i.e. visual inspections and stains). It is possible for gonococci and meningococci to exist in healthy people with carrier the rate being over half the population, this test therefore does not necessarily mean that the person is suffering from disease. However it is possible for a healthy carrier to become diseased and the health of the patient can reduce at a dramatic rate which can be fatal; speed of investigation can therefore be vital.

When a person, most often a child, who has pneumonia is unable to produce sputum, aspirated muco-pus from the nasopharynx can provide a specimen suitable for investigation. The specimen is collected by inserting a plastic catheter into the nose and collecting the sample in a plastic bottle, once the specimen is collected it is treated in much the same manner as sputum (see §1.4.7).

For the investigation of whooping cough a special swab called a pernasal swab is used. This swab has a flexible stem so that it can be inserted into the nose and can move down the nasal passages into the nasal cavity. It is possible to have the patient cough into a plate, however this method as satisfactory. The swab is seeded on special growth media that allows ***Bordetella pertussis*** to be isolated from other commensals. The culture is inspected after 48 hours and if there are no colonies present of the

[9]This is a specially curved swab that enters through the mouth and is protected so that it does not become contaminated with the normal flora.

pathogen then the media is seeded again. If plates are used instead of swabs then these plates have to incubated in moist air for four days after which they are visually inspected.

1.4.4 Laryngeal Swabs

The swab is inserted through the mouth and into the larynx, when contact is made the patient is stimulated to cough and mater from the larynx and trachea is deposited on the swab. Apart from the common infections of the larynx, this type of swab is also useful for some types of diphtheria and some types of acute lung infection. The specimens that are collected in this fashion are treated in the same manner as samples of sputum (see §1.4.7).

1.4.5 Mouth Swabs

Most infections of the mouth exhibit themselves as ulcers, which are sometimes cause by bacteria. In order to investigate these infections, the lining of the mouth is sampled, this is performed in much the same manner as that employed for the throat and nose. The specimen is stained and examined for Vincent's organisms or yeasts, if this proves negative then the swab is seeded onto blood agar plates where one is incubated aerobically and the other anaerobically. Organisms such as haemolytic streptococci or *Staphylococcus aureus* can be identified on these plates after 24 hours of incubation.

1.4.6 Antrum Wash-Out Fluid

The saline washings from patients that are suffering from chronic antrum infections can be the subject of investigation. There are two main problems; firstly that the washings are contaminated with the *normal flora* of the upper respiratory system and that the saline fluid can kill some of the more sensitive organisms that may be present[10]. Often this type of specimen is investigated in order to gauge the success of a particular treatment (often antibiotic). If an electronic nose was used to sample the odour of the specimen immediately after it was collected then the saline solution would not have a

[10]This usually occurs in the first few hours after the washings have been collected.

significant affect on the results of investigation. The ability of the electronic nose to cope with normal flora is also a factor here.

1.4.7 Sputum

There are many pathogens that are investigated in specimens of sputum from patients with upper respiratory infections, examples of such pathogens (including some more rare ones) are *Streptococcus pneumoniae*, *Streptococcus pyogenes*, *Staphylococcus aureus*, *Haemophilus influenzae*, *Mycoplasma tuberculosis*, fungae, *Legionella pneumphila* and *Friedlander's bacillus*. Stains of samples of the sputum can be made that allow immediate inspection, this can show up pneumonia. The sputum can also be chemically treated and seeded on blood agar plates which are incubated for 24 hours[11] in air plus 5-10% CO_2. Also in order to isolate particular organisms, such as *Haemophilus influenzae*, antibaterial agents are used in the cultures to inhibit the growth of unwanted micro-organisms. The colonies that grow in the cultures are visually examined (often with a hand lens) and sometimes the colonies are stained in order to aid identification, for example to stop the similar colonies of *Streptococcus pneumoniae* and *Streptococcus viridans* from being mistaken for each other. Because there are so many different micro-organisms that can be isolated, the actual procedure is tailored to each patient, so therefore there is a very large number of different investigations that can be performed. The problem for an electronic nose might be the odour of the sputum itself, if the odour of sputum from patient to patient (or from day to day in the same patient) is highly variable then there may be difficulties in obtaining performance that is good enough to be useful. A possibility may be to treat the sputum chemically in some manner that would isolate the bacteria cells from, say, the pus cells within the sputum, once this is achieved then the bacteria could be smelt.

[11] Cultures can be made of untreated sputum immediately after collection in order to perform quality checks on the specimen.

1.5 The Role of An Electronic Nose

The application of an electronic nose would reduce the role of culturing as the major tool in clinical practice, and therefore a modified faster procedure would be followed. The description detailed in this section is for illustration purpose, the actual procedure followed would be developed by exhaustive clinical trials and development. Where originally 1, 2 or more specimens were collected from various sites on the patient; double this number would be collected. The first set would be sent to the laboratory for analysis, as before and the second set would be 'smelt' by an electronic nose.

An alternative to this method, in some cases, would be to use an electronic nose to sample a patient's breath. Work has been previously performed where an electronic nose was used to sample cows breath in order to identify those cows that were suffering from ketosis [15, 16]. The variability of odour of the patient's breath may be a problem when analysing subtle odour differences. Directly smelling an ear would prove more difficult because air is not expelled from the ear. It may be possible to pump air carefully into the ear and use the air that would be expelled from the ear for analysis.

The analysis procedure may be two stage; firstly ascertaining whether the specimen contains infected matter (i.e. abnormal bacteria populations compared to the *normal flora*) or not, and secondly if the specimen contain infected matter what is the responsible pathogen.

If the electronic nose is differentiating between a specimen which includes a pathogen and a specimen with just the normal flora, is not that the former gives off an odour and the latter does not but rather that the odour associated with a *normal flora* is tainted (or changed) in some way. The exact nature of this change of odour from the *normal flora* odour is dependent on many variables, such as the type and population size of the pathogen, length of time since the onset of infection, the habits of the patient (does he/she smoke?) and the action that triggered the infection (e.g. viral infection or wound). Therefore attributing a particular change in odour as being due to a particular variable is difficult, however if it can be attributed to the presence of a pathogen some useful results will result.

This system could be used to test the efficiency of antibiotics. After a patient has received antibiotic treatment, the patient could return to the clinic at set intervals, for example once every two days, in order to have further samples analysed. If the treatment is effective, the analysis should reveal a decreasing presence of pathogens. Also, clinical trials of new antibiotics could involve this system. Comparisons between the effectiveness of different antibiotic treatments could also be performed.

It would be possible to analyse bacteria growth from samples taken from many other sites around the body (or even bacteria samples that are grown within a laboratory). More details on the possible role of electronic nose technology is given are the last chapter.

To summarise, an electronic nose was applied to the detection and classification of bacteria because it allows the possibility of a breakthrough in the area of treatment of infectious disease. Rapid analysis of infection may be possible using electronic nose technology. Current clinical methods have not undergone significant change for many decades, the solution proposed here is a major development.

Chapter 2

Review of Electronic Nose Technology

This research project had a basis in electronic nose technology. The term 'electronic nose technology' is used to mean the wide range of scientific disciplines that are involved with the application of the analysis of odours to various (mainly industrial) problems. Such scientific disciplines include sensor design, electronics design, microprocessor systems, artificial intelligence, statistics, chemistry and (in this particular research project) micro-biology. Here the basic theory of olfaction is introduced and the relationship between natural and artificial olfaction is highlighted. The major developments in electronic nose technology are also highlighted with theoretical and technical detail given in later chapters. The overall purpose of this chapter is to furnish the reader with an appreciation of the state of electronic nose technology, what methods have been attempted and what successes have been achieved. Decisions on the paths of research that were pursued were guided by results achieved in previous work in this field.

2.1 Introduction to Electronic Noses

An 'electronic nose' has been defined in many ways:

> "An electronic nose is an instrument, which comprises an array of electronic chemical sensors with partial specificity and an appropriate pattern recognition system, capable of recognising simple or complex odours [3]."
>
> "An electronic nose consists of an array of gas sensors with different selectivity patterns, a signal collecting unit and pattern recognition software applied to a computer [17]."
>
> "The concept of the electronic nose is the arrangement of an array of non-selective sensors (instead of biological sensors) feeding data to a pre-trained neural network [18]."

Moreover, an 'electronic nose' is a system, inspired by biological systems, that samples odourous gases and provides information about those gases. The system encompasses the sensor array, any interfacing electronics, any additional instrumentation and a pattern recognition sub-system. The sensor array consists of non-specific gas sensors the sensitivities of which overlap. Non-gas sensors (for example temperature and humidity) may also be included in the sensor array and they may be specific.

A commercial electronic nose was the starting point in this research. The commercial electronic nose was itself the product of considerable research and development[1]. The appearance of commercial electronic noses has been a recent occurrence (mainly happening in the previous 3 years). Presently there are only a few manufacturers involved, these being mainly Alpha M.O.S. (France), Neotronics Scientific Ltd (UK) and Aromascan PLC (UK). There have been other commercial products available that can best be described as gas monitors which have been available for longer, however these products do not really come under the above definitions of 'electronic nose' because they either comprise of one sensor or do not discriminate between odours but monitor the level of a particular gas (or group of gases).

An Electronic Nose has been the subject of much research at the University of Warwick over the past 15 years. Research began at the University of Warwick in 1982 [19], although the use of the term 'electronic nose' did not enter common usage

[1] Some of the research and development was performed at the University of Warwick.

until the late 1980's. The first work performed in the area of artificial olfaction was performed in the early 1960's by Moncrieff [20], this research was not electronic but mechanical based, it was three years later that the concept of an 'electronic nose' was investigated by Wilkens [21]. The idea of using metal and semiconductor gas sensors within an electronic nose was developed one year later by Buck [22] and also the use of modulated contact potentials by Dravnieks [23]. A brief history of electronic nose technology has previously been published [3, 24, 25]. A more accurate description of the instrumentation employed is given in section 2.4 of this chapter.

2.2 What Are Odours?

Fundamental to a nose of any kind are the actual odours themselves. An understanding of what odours are, from a scientific point of view, is essential if electronic nose technology is to be successfully applied. An odour consists of one or more types odorants (molecules responsible for odour), depending on the type. A simple odour, for example ethanol, has only one odorant note. A complex odour, for example coffee, has hundreds of odorant molecules [1]. Table 2.1 shows the composition of a typical coffee odour, it can be observed that this odour was in fact made up from 631 different molecules. For complex odours it is also important to consider the relative concentration of each odorant, a small change in relative concentration can result in a change in the odour quality. In general odorants are hydrophobic, polar and small (18-300 Daltons), these qualities allow odorants to react with gas sensing devices (natural or artificial) and so to give a response.

The relationship between odorant structure and the perceived smell has been previously researched [26, 27] but still is not fully understood. However different groups of odorants have been identified, such as those indicated in Table 2.1, and particular notes (for example rose, lemon and off-flavours) have been linked to particular odorants. The large number of constituents of a complex odour, which can number many thousands, mean that it would be difficult to break down an odour into its constituents and perform an analysis of each constituent. Human perception of complex odours are

Table 2.1: Typical composition of a coffee odour (source [1]).

Class of Odorant	Number of Types in Class
Furans	108
Pyrazines	79
Pyrroles	74
Ketones	70
Phenols	44
Hydrocarbons	31
Esters	30
Aldehydes	28
Oxazoles	28
Thiazoles	27
Thiophenes	26
Amines	21
Acids	20
Alcohols	19
Pyridines	13
Thiols	13
Total number:	631

often associated with few simple notes, such as musky or nutty, rather than a detailed chemical analysis. The molecular construction of some typical odorants are shown in Figure 2.1; it can be observed that most odorants are simple cyclic molecules or short chains.

Each odorant has an associated threshold above which they have a perceived odour. This thresholds are usually expressed as parts per billion in water (ppb in water). Table 2.2 shows some common odorants and their olfactory thresholds, it can be observed that the spread of thresholds covers up to 6 or 7 log steps, and that some odorants have very low thresholds (fractions of 1 ppb). This contrasts quite differently

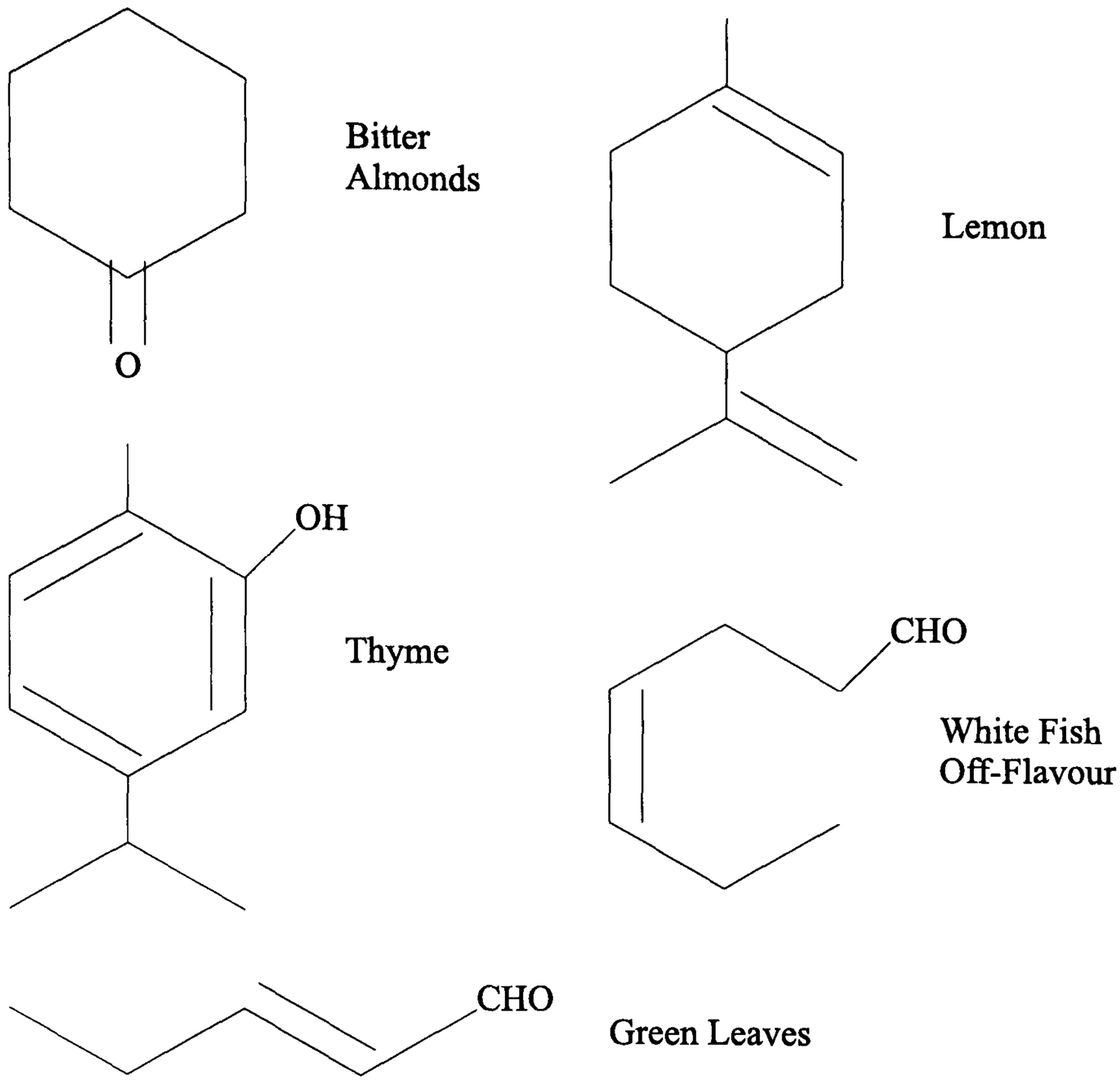


Figure 2.1: Diagram showing the molecular construction of some typical odorants along with a description of their odours (source [1]).

to artificial gas sensors (details of which are given in section 2.4.1) where sensitivity to volatile compounds, such as odorants, is at best in the region of 100 ppb to 100 ppm[2]. This raises the question whether an electronic nose would be sensitive enough to detect the subtle changes in odour from different bacteria types. Unfortunately this is a difficult question to answer other than to test this hypothesis empirically. The actual olfactory receptor cells only have sensitivities in the region of ppm yet the olfactory system as a whole, has a sensitivity as high as a fraction of ppb, this enhancement indicates the effect of signal processing that takes place within the neural pathways of the olfactory system (discussed in section 2.3). This enhancement principal may also apply to electronic noses.

The odour resulting from the growth of bacteria have particular qualities, often these qualities are unpleasant to humans (such as the smell of off-food). It is common

[2]The lowest published figure is 10ppb. Many odours have values at ppm in air.

Table 2.2: Table showing some common odorants and their thresholds of detection in the human olfactory system (source [1]).

Odorant	Odorant Description	Threshold (ppb in water)
Diacetyl	Beer off-flavour	500
Trans-2-hexenal	Green leaves	316
Geraniol	Rose	290
5-isopropyl-2-methylphenol	Thyme	86
Limonene	Lemon	10
Cis-4-heptenal	White fish off-flavour	0.04
Octa-1,5-diene-3-one	Butter off-flavour	0.01
2-isobutyl-3-methoxypyrazine	Green peppers	0.02
α-terpinethiol	Grapefruit	0.00002

practice in a micro-biology laboratory to smell bacteria in order to aid the process of identification. Particular bacteria types, such as *Pseudomonas aeruginosa*, have a characteristic odour that is rarely mistaken to the trained nose. However, laboratory staff can only learn these smells by experience, simply being informed that, for example, a particular bacteria smells 'nutty' or 'musty' is not specific enough. This fact indicates another property of odours, which is association of odours to abstract concepts. The laboratory staff learn to associate a particular odour with a particular bacteria type. Because no formal system of bacteria identification has been developed from this skill, it is possible that the human nose is not able to discriminate reliably between bacteria types. Many smells that are important to humans are the result of bacteria activity, for example bad food and body odour, and failing to correctly associate a particular odour could result in, for instance, bad food being eaten (which could possible be fatal). The role of the electronic nose in treatment of disease is not to merely discriminate whether an odour originates from bacteria or not, but to discriminate between bacteria types (or genus).

A common procedure for analysis of odours are gas chromatograph scans (GC

scans). This results in a spectral breakdown of the chemical constituents. There are sensitivity problems with this procedure [28] that mean that some of the subtle differences between odours that a human can differentiate cannot be detected. Unlike the ear, the olfactory system does not work by spectral analysis. Usually, a singular property of an odorant, for example its molecular weight, accounts for only a limited number of the ascribed properties. Thus, one dimensional measurements (or primitives) such as wavelength for light (used for colour perception) or frequency for sound (used for hearing) are not suitable for classifying odorants; this accounts for the multi-dimensional analysis observed in natural olfactory systems [29]. There seem to be three groups of properties of odorants that affects olfaction [30]:

Determinants: These are individual properties of odorants such as functional groups, chain length, or positions of double bonds.

Ligand: These are the properties of the single type (or group of) of odorant, in analogy with common terminology used to characterise the biochemistry of the molecular signals and receptors.

Odour Object: These are the properties defined by mixtures odorants and result from the reactions that take place between odorants.

From this is can be concluded that the analysis of odours at a molecular level, whilst being productive for particular types of odour research projects, would be inappropriate where odour classification is sought by means of a parallel to the human olfactory system (such as this research project). It is unnecessary to discuss the chemistry of odours further, however research has been performed in order to identify the components of bacteria odours [31], this showed that bacteria odours are complex and dependent on a large number a environmental variables.

2.3 The Human Olfactory System

Since the methods employed in this research project are inspired by and modeled on (from a mathematical point of view) the human olfactory system, an overview of the

human olfactory system is given here in order to indicate the origins of the data pre-processing and classification system design that were employed.

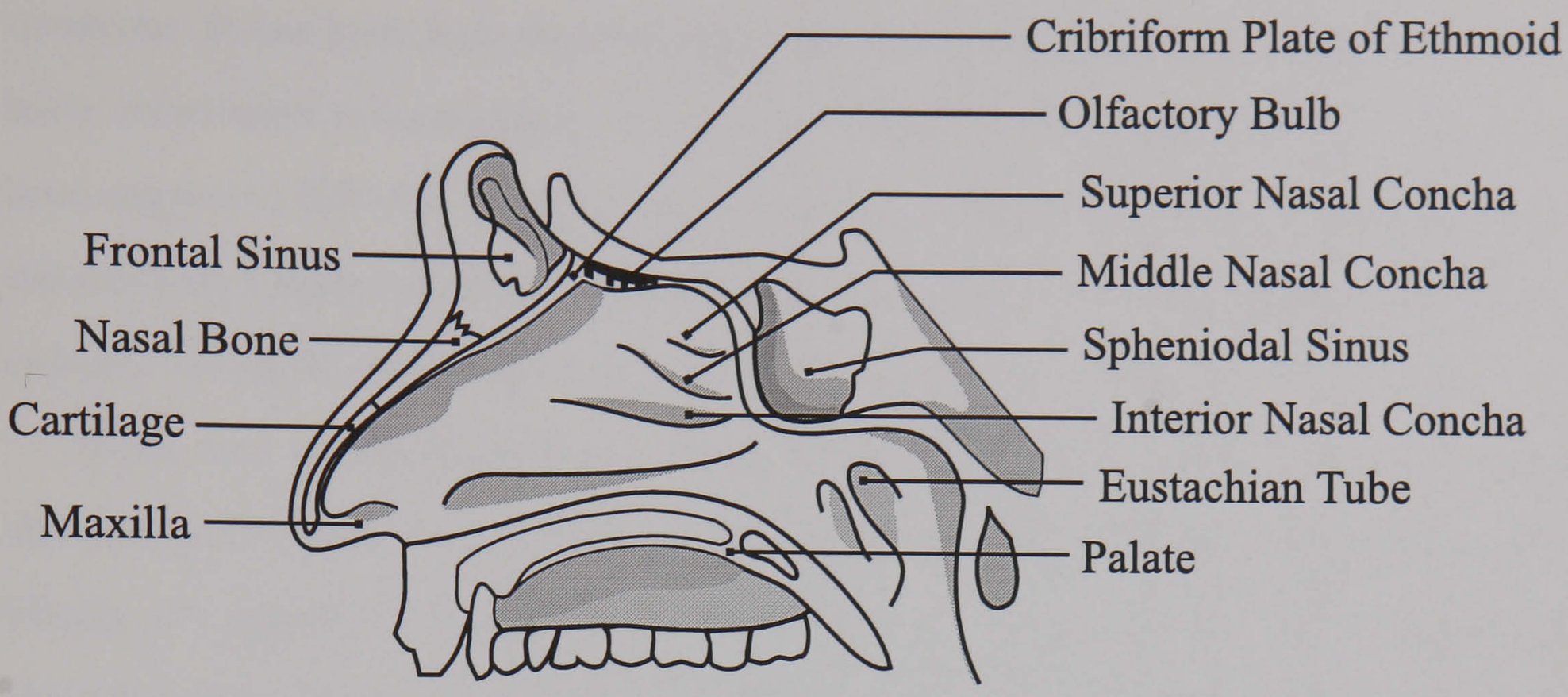
The human olfactory system is stimulated by information contained within odorants that are released from an object, causing patterns of brain activity that are essentially the sense of smell. The relationship between brain activity and odorant quality is one of the least understood problems in biology. Research has, however, been reported [32, 27] that allows sufficient understanding in order to implement the fundamentals of olfaction in an artificial system. Figure 2.2 shows where the olfactory system is located in a human and the basic structure of the olfactory nervous system. The olfactory system can be divided into three main stages as described in the following text.

2.3.1 Human Olfaction: First Stage - The Olfactory Receptor Cells

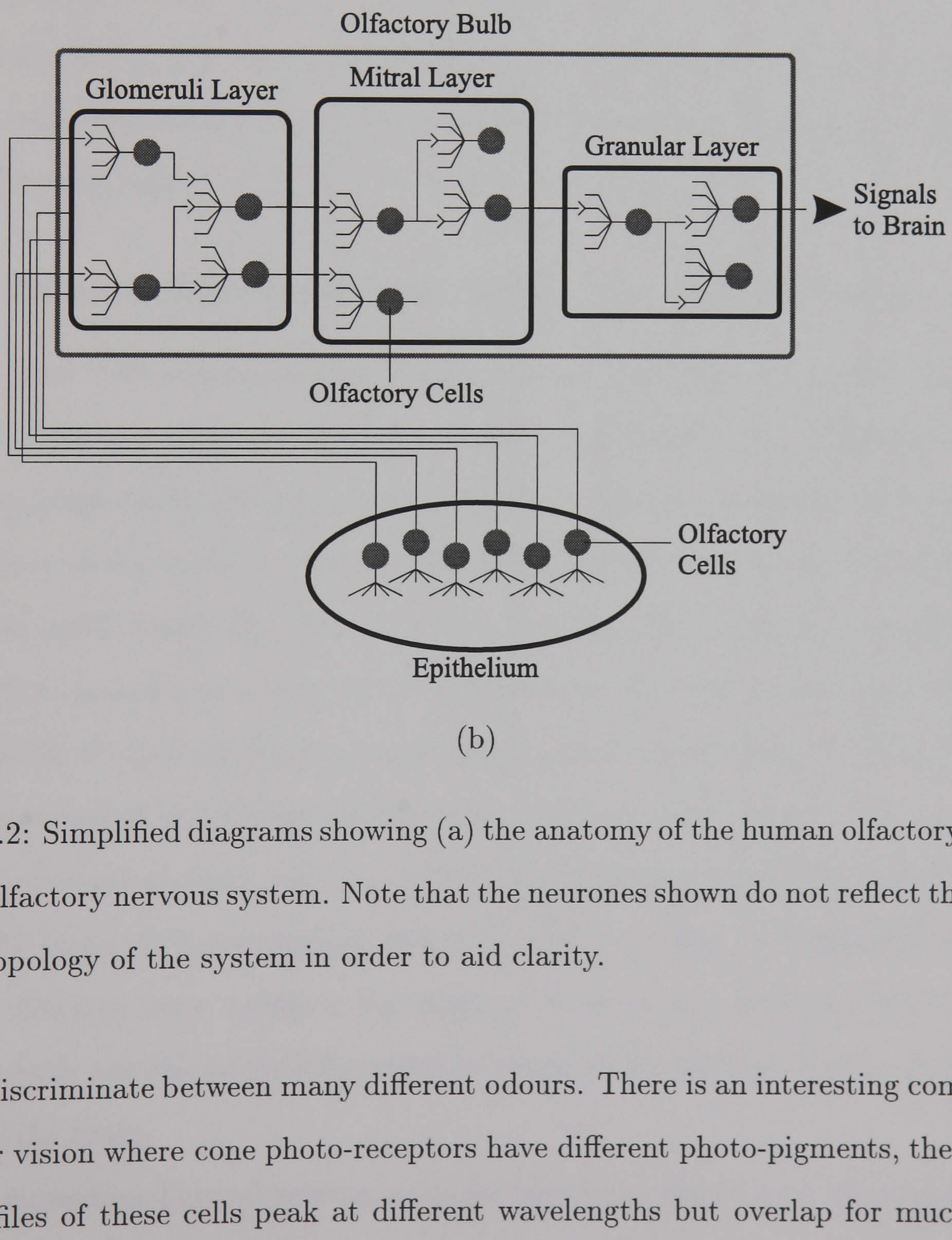
Odorant molecules are inhaled into the nasal cavity and onto the epithelium, which is the area where the olfactory receptor cells exist. The olfactory receptor cells have hairs, or cilia, which are embedded within a thin aqueous mucus layer which covers the epithelium. The odorants enter the mucus layer where they react with the cilia. The nature of this reaction seems to be based on G-protein neurotransmitter receptors [33] which exist on the surface of the cilia. Because of the number of different receptor protein types is relatively small (i.e. 200 to 500 [34]) and the number of receptor cells is relatively large (approximately 100 million for both nostrils), there is overlapping sensitivity between individual receptor cells[3]. The exact mechanisms employed in the olfactory receptor cells is not of great importance here since we are not able, at present, to clone artificial versions and also that what is of interest are the fundamental concepts that result from neurological investigation. The important qualities of olfactory receptor cells are that they are high in number, are non-specific and have a limited life-span[4]. The overlapping sensitivities of the olfactory receptor cells would indicate that our sense of smell would be non-specific, this however is not the case as we are

[3]This may not precisely true because there may exist a small number of olfactory receptor cells that have a very specific sensitivity to special odours called pheromones.

[4]The entire population of olfactory receptor cells is constantly being renewed, each cell has a life span of 3 to 4 weeks.



(a)



(b)

Figure 2.2: Simplified diagrams showing (a) the anatomy of the human olfactory system (b) the olfactory nervous system. Note that the neurones shown do not reflect the actual neural topology of the system in order to aid clarity.

able to discriminate between many different odours. There is an interesting comparison to colour vision where cone photo-receptors have different photo-pigments, the absorption profiles of these cells peak at different wavelengths but overlap for much of the

spectrum. It has been indicated [35] that this overlap is necessary in order to discriminate wavelength independently of intensity. Similarly in the olfactory system it has been suggested [36] that overlapping sensitivities enable odour quality (ligands) to be discriminated separately from odour concentration. Therefore overlapping sensitivities enhance the ability to discriminate between many different odours.

Given that an olfactory receptor cell has a response to an odour there are some characteristics of the response that affect subsequent stages in the olfactory system. Firstly, the response is delayed by several hundred milliseconds after the time at which the odour first starts to reach the cilia. Secondly, after an initial peak, the response (assuming that the concentration of the odour is constant) declines with time. Thirdly, the response only occurs over 1 or 2 log steps in odour concentration.

A small amount of signal processing takes place in this stage of olfaction. The odour signal is amplified and secondary messenger signals are generated which are transmitted to the olfactory bulb.

2.3.2 Human Olfaction: Second Stage - The Olfactory Bulb

The olfactory bulb consists of three major layers of neurones: glomeruli, mitral and granular. Olfactory signals in this stage undergo considerable processing that enhances sensitivity, improves immunity to noise, increases selectivity and allows for the constant regeneration of the olfactory receptor cells. Figure 2.2(b) shows the arrangement of the individual layers within the olfactory bulb. It is slightly misleading to segregate the bulb in this manner because there is a high amount of interconnectivity between the layers and there are no strong functional distinctions between adjacent layers. However for the purposes of simplifying the olfactory system in order to gain some insight into its properties and makeup, the bulb will be described as composing of definite layers. Figure 2.3 shows the basic configuration of connections within the olfactory bulb. The primary olfactory nerve connects the olfactory receptor cells in the epithelium to the olfactory bulb, and the lateral olfactory tract connects the olfactory bulb to the piriform cortex in the brain.

Two dimensional neural activity patterns have been observed on the surface of the

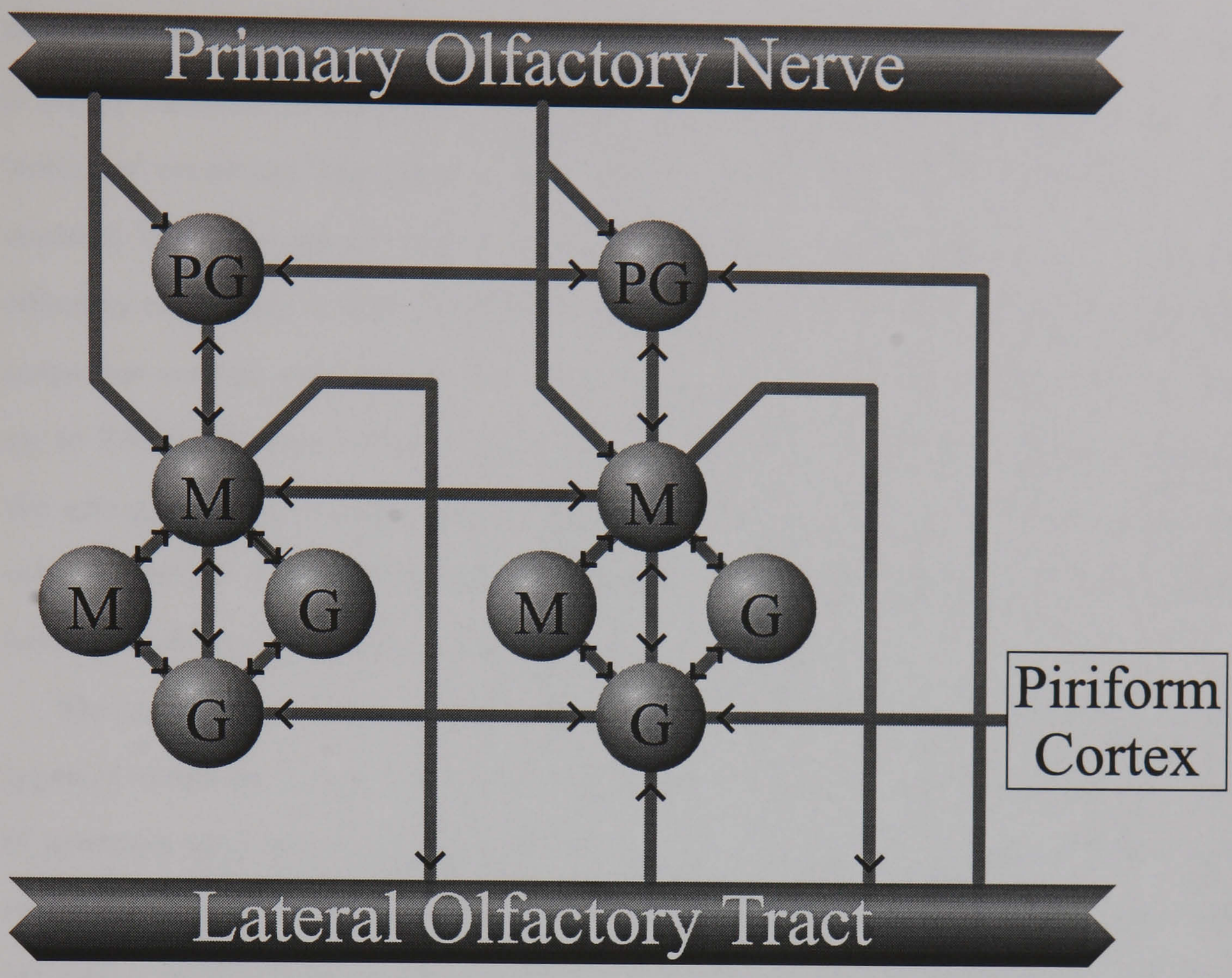


Figure 2.3: A simplified diagram showing the connectivity within the olfactory bulb, based on model proposed by Freeman [2]. Note the number of connections has been reduced to enhance clarity. Key: PG - Periglomerular (glomerular) neurone, M - Mitral neurone, G - Granule neurone.

olfactory bulb [37]. Particular areas have not been linked to particular odours, so there is no area that corresponds to say alcohols. The observed patterns seem to be related to spatial information within signals received from the olfactory receptor cells. This therefore indicates that there is no corresponding areas of the olfactory receptor cell sheet that can be identified with particular odorants [38]. Therefore if a particular group of olfactory receptors cells are damaged, this does not result in the ability of the animal concerned to smell a particular odorant but rather the sensitivity is reduced. This aspect of the olfactory system highlights its fault tolerant design (a trend also reflected in the processes that occur in the brain).

The glomeruli (glomerulus) layer contains in the order of 5000 neurones. The pat-

tern of neurone organisation is also observed in other parts of the brain (the brain is highly modular with common organisational patterns, such as 'blobs', 'barrels' and 'columns' occurring throughout); this neural structure has been the subject of much research [39]. The glomerulus essentially extracts features (or descriptors), grouping olfactory maps that contain similar properties (such a ligands) and separating olfactory maps that contain dissimilar properties. There is massive convergence in this layer with up to 25000 olfactory receptor cell feeding into one glomeruli cell, this fact mediates the grouping activity and enhances sensitivity. The discriminatory behavior of these cells are further enhanced by lateral inhibitive connections; this layer is not a simple feed-forward structure but a complex feedback structure.

The mitral layer[5] contains approximately 100000 neurones, in which there are two types of neurones, larger mitral cells and smaller tufted cells. Although the two types of neurones are high interconnected, they do perform separate functions. The mitral cells are highly involved in the divergent and convergent operations within the bulb (grouping of olfactory maps), whereas the tufted cells control interactions within the bulb (control mechanism). Another interesting property of the mitral cells is that they receive signals from areas of the brain (i.e. feedback) that relate to the current emotional status [40]. For example, if an animal is hungry, this emotional state is communicated to the mitral cells which results in a different response to food odours. An electronic nose might be able to implement a compensatory system where for example ambient air temperature is sensed and fed back into the pattern recognition sub-system.

The granular layer may be involved in improving specificity in the olfactory system [27]. The connections between this layer and the mitral layer are mainly inhibitory, thus signal compression occurs. The output from this layer is then input to the brain via the lateral olfactory tract.

Feedback within a neural network, in this case the olfactory bulb results in chaotic behavior which is extremely (if not impossibly) difficult to model precisely, most information described here is the result of observation of neurone activity in the olfactory

[5]The term 'mitral layer' as used in this text actually refers to two layers called the External Plexiform Layer and the Mitral Cell Body Layer.

systems of various animals to various stimuli. The feedback connections are shown in Figure 2.3, it can be observed that there are many feedback neural paths. There have been attempts to form an accurate artificial model [2], these have shown some success. However the limitations of implementing close biological models on present day computer systems continues to be a limiting factor.

2.3.3 Human Olfaction: Third Stage - The Brain

The actual area of the brain that is concerned with olfaction is the olfactory (piriform) cortex. It is connected to the olfactory bulb via the lateral olfactory tract. This area is located in the sensory area of the cortex, compared to the visual cortex (which is the biggest single part) it is small. This reflects the simpler task of olfactory perception compared to the highly complex task of visual perception. The operation of this part of the brain is mainly that of content addressable memory where incoming odour signals are compared with previously learnt odour signals and, where appropriate, associations made. Because of the relative simplicity of the olfactory cortex, many attempts have been made to model this part of the system with artificial neural network (ANN) paradigms such as Hopfield Networks [41] and Boltzman Machines [42]. This work have shown some success, but it is difficult to model accurately such a complex system employing conventional computing resources. Many other workers have been able to model certain behavioral aspects of odour perception. These successes reinforce observations and theories that have resulted from biological empirical study. This type of research was highly relevant to this project where artificial neural networks were employed. Detailed description of the olfactory cortex topology is unnecessary, but there exist many publications which cover the subject in great detail [27]. The field of artificial intelligence is a fast moving one and developments in the artificial neural network models of olfaction are set to progress. However their suitability to a 'real-world' application such as the one described in this thesis is as yet unproven.

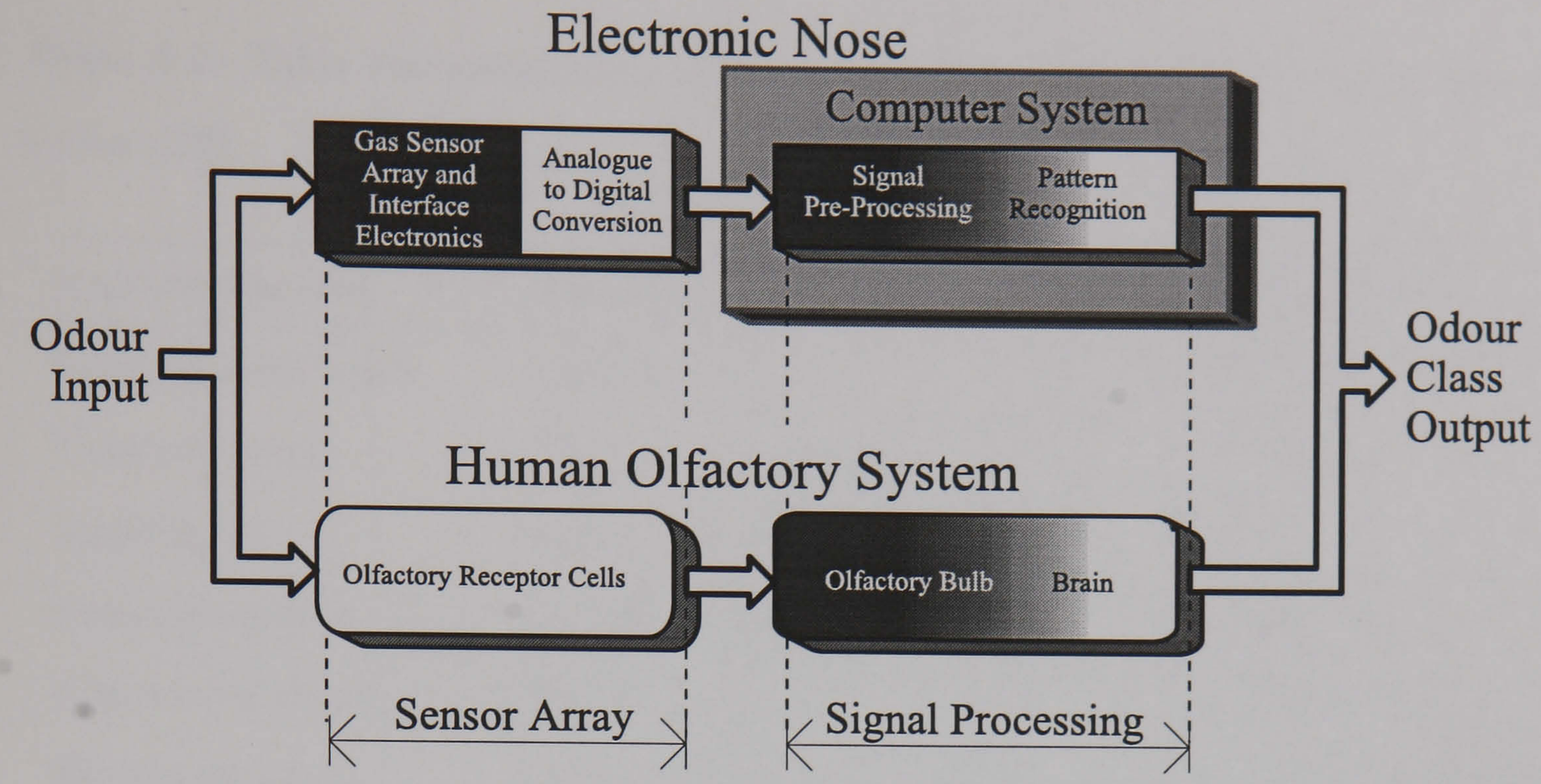


Figure 2.4: A comparison of a typical electronic and the human olfactory system showing the major processing elements. The graded shading indicates fuzzy functional boundary.

2.4 Artificial Olfaction

The role of an electronic nose is to perform artificial olfaction. This section describes how electronic noses operate and how this is related to the human olfactory system (as described in section 2.3). Artificial olfaction usually includes of an array of gas sensors coupled via a suitable electronic interface to an analogue to digital conversion (ADC) circuit. The resultant digital information is fed into a computer system where software is running that applies signal processing algorithms. Figure 2.4 shows a comparison between a typical electronic nose, such as that employed in this research, and the human[6] olfactory system.

Although the electronic nose was inspired by nature, there are many functional differences. Firstly, the olfactory receptor cells in the human perform a limited amount of signal processing, whereas the artificial counterparts are 'dumb'. The human olfactory system is difficult to segment in terms of functionality because of the manner in which the signal processing is distributed. The functionality within the artificial olfactory

[6]This diagram also adequately describes the olfactory system of most vertebrates.

Table 2.3: Table summarising common sensor technologies and their target gases (source [3]).

Reactive Material	Sensor Class	Target Gases
Sintered metal oxide	Chemoresistor	Combustible
Catalytic metal	Thermal, e.g. pellistor	Combustible
Lipid layers	Acoustic, e.g. piezoelectric/SAW	Organic
Phthalocyanines	Chemoresistor	NO_x, H_2, NH_3
Conducting polymer	Chemoresistor	NH_3 and alcohols
Electro-chemical	Potentiometric/amperometric	NH_3, CO, CH_3CH_2OH
Catalytic gate	Potentiometric, e.g. Pd-MOSFET	Combustible
Organic semiconductors	Optical, e.g. IR absorption	CH_4, CO_2, NO_x

system (an electronic nose) is much more segmented, this is due to the nature of the technology, for example the complex artificial neural networks employed are simulated in software and cannot exist at hardware level (unlike the gas sensor array). Artificial olfaction can therefore be split into three functional stages; (a) gas sensor array (and interface electronics), (b) signal pre-processing, and (c) odour classification. This review of artificial olfaction is therefore continued in sections that reflect these functional stages.

2.4.1 Gas Sensor Technology

There are many types of sensor technology that have been developed for gas sensing, see Table 2.3.

As previously stated in section 2.3, the special feature of the gas sensor array within a nose is that they should be non-specific (i.e. have overlapping sensitivities). This is relevant to both human and artificial olfactory systems. It is more difficult to develop a specific gas sensor than to develop a non-specific gas sensor, and this has resulted in a wide range of applicable gas sensor types. The most commonly employed gas sensor types are sintered metal oxide [43, 44, 45], lipid layers [46] and organic polymers [45, 6].

Table 2.4: Types of gas sensor array previously employed in electronic noses (source [3]).

Gas sensor array type	No. of sensors	Country of origin
Sintered metal oxide Chemoresistors	6,6,8,12	Japan, USA, Japan, UK
Lipid layer (piezoelectric crystals)	8	Japan
Lipid layer (SAW devices)	6	Japan
Lipid layer (Phthalocyanine Chemoresistors)	5	UK
Organic polymers (Chemoresistors)	12,20	UK, UK
Organic polymers (SAW devices)	12	USA
Lipid layer (electrochemical)	2-18	USA
Pd-gate MOSFET	10	Sweden
Optical FET camera	324 pixels	Sweden

Table 2.4 shows the types gas sensor array employed in previous electronic noses.

At the University of Warwick, electronic nose research has focused on two sensor types, metal oxide and polymer. Sintered tin oxide sensors have been previously shown to perform well in electronic noses [45], their major weaknesses are drift due to slow chemical reactions in the reactive element and sensitivity to ambient temperature and humidity. The degradable effect of these weaknesses on electronic nose performance has been reduced by instrumentation design and compensation within the signal pre-processing and classification stages. Polymer based sensors tend to suffer from sens-

itivity to ambient humidity (generally worse than metal oxide sensor types), and this weakness is one of the current research activities at the University of Warwick. Careful setup of the experimental environment can reduced the effect of humidity, for example precise control of ambient humidity. This may prove difficult in some of the conditions that electronic noses may be required to operate in in future applications, for example a farmyard [15]. Miniaturisation has been attempted by producing thin-film devices [10]. One advantage of polymer based sensors is that there is a limitless number of polymer 'recipes' which give these sensors the potential to be available in a wide number of sub-types which could allow for more application specific sensor arrays (for example one specialising in beverage odours). The current research effort is mainly directed towards miniaturisation, reducing power consumption, increasing stability and reducing production costs. It is difficult to state that one particular sensor type performs best but it has been suggested that the polymer types are the most promising [47]. A book edited by Gardner and Bartlett includes a good investigation into the major gas sensor technologies [45]. The sensor type employed in this research was sintered tin oxide, the operation of which are detailed in chapter 3, because it is particularly sensitive to the levels of primary metabolites (for example alcohols, ketones and amines).

2.5 Signal Pre-Processing

The signal pre-processing stage modifies the signals from the sensor arrays into features which are suitable for input into the pattern recognition stage. Signal pre-processing is described in detail in chapter 4, here the popular and promising algorithms are assessed.

Numerous pre-processing algorithms have been employed in previous work. They mainly fall into two groups: sensor feature models and basic statistical manipulation. The Difference Feature model outputs features that are related to the difference in the sensor signal when exposed to air and when exposed to the odour being measured. This feature model has been often applied [48, 49, 50, 51, 52]. However, it has generally proved inferior to the fractional feature model. The Fractional Difference Feature model outputs features that are related to the ratio of the difference of sensor response

(in air to that in the odour being measured, i.e. the Difference Feature model), to the response in air. This model has also been widely applied [50, 6, 52] with good results. Another common model is the Relative Feature Model, which outputs features that are related to the ratio of the sensor response in air to the response in the odour being measured [53, 50]. This model also tended to perform less well than the Fractional Difference Feature model. It has been suggested that the Fractional Difference Feature model performs relatively well [50] for metal oxide sensors because it reduces the concentration dependence of the features that are output. Also the effect of base-line drift of each of the gas sensors being modeled, upon the features being output, is reduced. One common feature of the Difference Feature model, Relative Feature model and Fractional Difference Feature model is that they are steady-state. They do not make use of transient information contained within each sensor response. It is only very recently that transient (or dynamic) information has been employed in addition to steady-state information [54]. This work is promising and has shown that pre-processing utilising both transient and steady-state information can improve classification performance compared to pre-processing utilising steady-state information.

Basic statistical methods employed include linearisation, normalisation and auto-scaling. These methods are additional to the sensor model pre-processing algorithms. Linearisation methods have been, for example, included using the log of the output from the sensor model [43]. It has also been shown that, if operated over a limited output range, some sensor types (e.g. sintered tin oxide) behave in a quasi-linear manner [45]. In theory, linear data is simpler to process than non-linear data, this improves pattern recognition performance. Normalisation has been performed in one of two ways: array (or vector) normalisation and sensor normalisation. Array normalisation scales each vector, within the same data set, to the same length; usually unit length. This technique has been shown to improve performance for small sensor arrays and pattern recognition techniques that are sensitive to the scaling of input variables [50], in theory the scaling of input variables is unnecessary for many ANN paradigms [55]. Sensor normalisation scales the output of each sensor over the entire data-set to lies in the range [0, 1], this technique has been shown to improve overall pattern recognition per-

formance, however performance can be reduced if a 'noisy' sensor output is amplified during normalisation. Therefore, applying normalisation enhances good quality data and degrades bad quality data. Auto-scaling scales each sensor so that the average of its output is 0 and the standard deviation is 1. The effect of auto-scaling on pattern recognition has been previously discussed [55], and has been shown to be beneficial under certain circumstances. In general, more effort has been spent on researching classification than on pre-processing, this is not a good reflection of the importance of pre-processing on overall electronic nose performance.

As mentioned earlier, a recent development in pre-processing is the use of dynamic (or transient) models, i.e. models which output features which contain time domain information. This was first highlighted as promising in previous work [52]. A comparison has been performed between the performance of static and dynamic models [54], it showed that dynamic models can perform better than static models. The dynamic model employed was applied to tin oxide sensors, although models have been applied to other sensor types, such as polymer sensors [56]. Spectroscopic methods have been employed to the transitory information within sensor signals with success [57], using a method called multi-exponential transient spectroscopy (METS) and applies it to alcohol vapour data. There is relatively a small amount of research where dynamic sensor signal information has been utilised, but the results so far are very encouraging.

2.6 Odour Classification

The field of classification, like many other fields of research, is still the subject of much argument. This field has been split into two areas within this text; classical statistical methods and ANNs. This categorisation is based more on implementation than any other quality. ANNs can be configured to behave as classifiers, and as such they implement classification by means of a network of simple interconnected nodes (or neurones). Classical statistical techniques implement classification by means application of a probability model. In reality, classifiers are 'virtual' in that they are simulated in a computer environment (in theory pen and paper could be used but would be imprac-

Table 2.5: Table showing the major advantages and disadvantages of artificial neural networks and classical statistical classification techniques.

Classification Type	Advantages	Disadvantages
Artificial Neural Networks	Good noise immunity, more able to cope with non-linear data, adaptable to system drift, confidence measure possible, ideal for VLSI implementation	Optimum solution difficult to achieve, can be unpredictable, can require massive computing power, perceived as 'black box' so difficult to analyse
Classical Statistical Techniques	Well established and trusted, tolerant of small data sets, better established software	Poor immunity to noise, advanced methods require substantial expertise

tical, to say the least); therefore implementation is the quality by which classifiers are visualised by their designers and simulated on computers. The field of classification is massive, a complete review of which is not possible here. The present review is based mainly on material published in the area of electronic nose technology. There are key texts, not based in electronic nose technology, such as [58, 59, 60, 55] which were used as a basis in which to understand the field of pattern recognition as a whole.

There have been many studies of the relative performance of artificial neural networks, applied to electronic nose technology, compared to other classification techniques [3, 45, 25, 48, 61]. These studies have shown that, in some cases, ANNs have displayed some advantages when compared to classical statistical techniques. Table 2.5 summarises the major advantages and disadvantages of ANNs compared to classical statistical techniques that have been identified in previous electronic nose research.

The comparisons shown in Table 2.5 are only what has been reported, and should be used as an indication only. Any attempt to rule absolutely on the relative merits

of any classifier type is futile for many reasons. Firstly, qualities of the data used in conjunction with classification methods have a major influence on the performance. Secondly, the 'skill' of the classifier designer can influence results (this is especially true of ANNs). Finally, the capabilities of the computer software employed impact upon the ultimate classifier performance. Therefore it was necessary to investigate a number of classification techniques, an exhaustive study of all possible classifier types would be too time-consuming, so the classifiers most likely to perform well, based on previous research, were studied.

The boundary between ANNs and statistical techniques is not easy to define. There is strong opinion [58, 60] that many ANNs simply implement previously known statistical algorithms, but using a different modelling method. For example, it has been proposed [62] that a feed-forward multi-layer perceptron (see section 2.6.2) can implement algorithms such as non-linear principal component analysis and multivariate non-linear regression. It has been claimed that the back-propagation ANN training technique (see section 2.6.2 was independently discovered in the statistical community, which called it 'gradient descent'. This type of parallelism of discovery is the result of groups of researchers working independently. Recently, however, there has been a tendency for researchers from different groups to collaborate which has reduced these problems, as books such as [60] show.

2.6.1 Classical Statistical Techniques

The term 'classical' as used in this text refers to those techniques of classification which where first developed and applied by the statistical community[7]. Although these techniques have been identified as inferior to ANNs in many applications it was necessary to investigate them in order to provide a 'benchmark' for ANNs and to ascertain whether ANNs were better classifiers or not in this application. There have been recent reviews [50, 63, 64] of the performance of many pattern recognition methods when applied to electronic nose technology. Categorisation of classifiers of this type is difficult, however distinctions are usually made depending on certain aspects:

[7]Statisticians who mainly work in the field of applied mathematics.

- Whether the classifier is parametric or non-parametric. Parametric classifiers assume quantities concerning the statistical nature of the data, such as the probability density function (PDF). Advantages of these classifiers are that they are able to obtain good performance with small and/or bad quality data. An example of a parametric classifier is a Bayesian based classifier. A non-parametric makes no assumptions about the statistical nature of the data and therefore can adapt. So therefore complex data can be utilised that would be difficult to analyse in order to, for example, approximate its PDF. An example of a non-parametric classifier is density estimation.

- Whether the classifier undergoes a supervised or unsupervised learning phase. A supervised learning technique is one where a data set (called the training set) comprising a set of input vectors with known output vectors are presented to the classifier during the learning phase. Internal classifier parameters are adjusted according to the error between the actual classifier output and the desired classifier output. When the classifier has been taught, a second data set can be input with unknown output vectors, the classifier should (if its design is correct) generalise in order to generate output. An example of a supervised learning classifier is discriminant function analysis. An unsupervised learning technique is one where the output vectors in the learning data set unknown and the internal parameters of the classifiers are adapted using only information contained in the input. Examples of an unsupervised learning classifier is k-means cluster analysis.

The following subsections describe the major classical statistical techniques so far applied to electronic nose technology. In each case the success of the technique is highlighted.

Principal Components Analysis

Principal components analysis (PCA) is a statistical technique used to reduce the dimensionality of a multi-dimension data-set. PCA is often used to implement a linear supervised classifier, in conjunction with discriminant functions. PCA is suited to

handling a sensor array were the gas sensors have overlapping sensitivities but not particularly suited to non-linear signals that gas sensors can output. PCA was applied to the discrimination of 5 different alcohols [50] with success (i.e. discrimination correct in all cases). Examination of the output from PCA in this application also showed that only 5 out of the total of 12 gas sensors were required. This highlights another useful function of PCA, namely that it can be used to simplify classifiers by eliminating unnecessary inputs. However, when PCA has been applied to more difficult applications, such as the discrimination between different beverages [50] or different coffees [65, 51] the success has been somewhat less. In the former, PCA was only able to differentiate between more general groups such as beers and spirits (and with less accuracy between bitters and lagers). PCA has been applied to basic classes of odours [43] with good results.

PCA does not necessarily have to be a linear technique [55]. The problem with non-linear PCA is that an optimum solution cannot be guaranteed and that the complexity of non-linear PCA is considerably greater than linear PCA.

Feature Weighting

Feature weighting (FW) is a statistical technique used to enhance separation. This technique is described in [64, 44]. FW is parametric in nature, where the data set is assumed to have a normal PDF with known standard deviations. FW can be employed as a parametric classifier in conjunction with discriminant functions. FW has been used to discriminate between 4 different tobacco odours [44] using a sensor array of 2 commercial semiconductor (tin oxide) gas sensors, where FW enabled all cases to be correctly discriminated. FW appears to be a promising statistical tool, although it is not always possible to accurately estimate the PDF of a small data set (especially if the data vectors have a high dimensionality, or are noisy). Also, this technique has not been widely appraised. It is common to employ this technique to 2 or 3 dimensional plots of features (such as principal components) in order to enhance groupings of vectors.

Multivariate Linear Regression

Regression is probably the most popular general purpose statistical classification technique. It is, however, used for predicting concentrations and not class membership, therefore its usefulness with electronic nose data is limited. Multivariate linear regression (MLR) is a form of linear regression where instead of 1 independent variable (input) being linearly regressed with 1 or more dependent variables (outputs), multiple (i.e. more than 1) dependent variables (inputs) are regressed. MLR has been applied to electronic nose data previously [66, 5] with some success. In order for MLR to work has been shown that it is necessary for the input vectors to conform to the principal of superposition [66, 44]. It has also been shown that an ANN can approximate a MLR classifier [55], and hence this technique has been unwittingly applied in many cases. It is possible, and probably likely, that electronic nose data is non-linear [50], for example tin oxide semiconductor gas sensors exhibit non-linear characteristics at high odour concentrations. MLR performs poorly with non-linear data which probably accounts for many of the problems with achieving good classifier performance[8]. Non-linear multivariate regression (NLMR) is a non-trivial technique that requires considerable statistical skills. However, a simple ANN called a multi-layer perceptron can exhibit NLMR [55] but is very difficult to analyse. A MLR classifier (for quantity estimation rather than class membership) provides a good benchmark by which other, similar, classifiers can be judged.

Discriminant Function Analysis

Discriminant function analysis (DFA) is a set of parametric statistical pattern recognition techniques, they previously been applied to electronic nose data [50]. When used for classification, DFA is a supervised technique. Class membership of data is decided using a set of discriminant functions, these functions discriminate between data vectors belonging to different classes. The most basic for of DFA is linear discriminants, first pioneered by Fisher in 1936 [67], which does not require any prior knowledge of the data PDF. Other DFA techniques are quadratic discrimination and logistic discrim-

[8] It is possible to limit this problem by ensuring that the odours measured are at low concentrations

ination[9], these have been applied to the discrimination of coffee odours [65]. These other discriminant functions require some assumptions about the data but provide better performance. DFA showed better performance on coffee odours (81% with cross validation classifier performance estimation) than other techniques, such as PCA and ANNs. DFA has also been applied to alcohol vapours with good results [44, 50] but this application is less 'demanding' than the coffee application, most good classifiers achieved 100% performance on alcohol vapours. DFA is promising because it has been able to discriminate between data vectors that have highly complex and non-linear relationships, as shown with the application to coffee odours. The best performing DFA was the quadratic type which requires a normal PDF, a normal PDF is not a rare occurrence, therefore this assumption has not proved to be a major problem. Wilk's Lambda is used in DFA as a confidence measure, it measures how much of the variation is related to class membership (see chapter 4 for further details).

Cluster Analysis

There are many different algorithms that come under the heading of 'cluster analysis'. In general cluster analysis (CA) undergoes an unsupervised 'learning' phase and is non-parametric. CA has been widely applied to electronic nose data [43, 50, 68, 45], the major CA algorithm has been one based on Euclidean Linear Cluster Analysis. In CA the data vectors are clustered during the 'learning' phase. The clusterings are based on the proximity of the vectors in a domain (usually feature space). A popular CA technique is k-nearest neighbour (k-NN), this has been extensively researched in the statistical community and is a major non-parametric classifier technique [60]. k-NN has been applied to electronic nose data previously [61]. In general CA has performed less well than other techniques, one problem being that CA is very 'sensitive' to the features that are supplied, and therefore pre-processing becomes very important. Another problem with CA is that clustering occurs according to the dominating information content, ad hence it is possible for clusters to reflect the elapsed time at which an odour measurement was taken (due to sensor drift over the duration of data gathering) more

[9] Further DFA types include canonical correlation and stepwise algorithms.

than the quality of the odours measured. CA has been applied to the discrimination of beverages [50] (2 lagers, 2 bitters and 2 spirits). This work showed that although CA performed well (allowing almost perfect discrimination between beverage groups bitter, lager and spirit) it was also 'sensitive' to data pre-processing methods.

Other Statistical Techniques

Apart from the major algorithms previously listed, there are a number of other 'classical' statistical methods such as Auto-regression [69], Principal Component Regression (PCR) [70] and Partial Least Squares (PLS) [70] that have been applied to electronic nose data (prediction of gas concentration in binary mixtures). There are a very large number of different classifier types that could be termed as statistical. However in this research those methods that have been popular and have demonstrated good performance were considered. An exhaustive search of all possible statistical methods was unfeasible. However the methods reviewed here do demonstrate that they should not be ignored totally for any other electronic nose application (it would be unwise to exclusively use ANNs for example).

2.6.2 Artificial Neural Nets

A more detailed description of artificial neural networks (ANNs) employed in this research project are given later in this thesis (chapter 4. This section, however, gives a brief review of ANNs and how they have been previously implemented in the field of electronic nose technology.

The term 'artificial neural network' has been defined in many ways:

"A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Inter-neurone connection strengths known as synaptic weights are used to store the knowledge [59]."

> "A neural network is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates only on local information. Furthermore each element operates asynchronously; thus there is no overall system clock [71]."
>
> "Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge [72]."

No one specification has been adopted by the artificial intelligence community as definitive, the previous quoted definitions however provide a good overall view.

An increasingly popular technique for odour classification has been the employment of ANNs. The development of ANNs was (and is) inspired by biological neural networks. The largest biological neural network known to exist, i.e. the human brain, performs in a manner which fascinates mankind and has driven mankind to formulate many theories as to the nature of thought. It is only recently that the basic structure of the brain has been understood, and it was early this century (1911) that the idea of a network of neurones being the underlying structure of the brain was postulated. The first attempt to construct artificial neurones was performed in 1943 by McCulloch and Pitts [73]. Some success was achieved, but the parameters associated with these ANNs had to be hand-crafted which made using large networks difficult. In 1949 Hebb formulated a learning rule for the ANNs developed earlier by McColluch and Pitts [74]. He also suggested that the brain's connections where constantly changing as it learns throughout its lifetime. Basically the learning rule was based on reinforcement and allowed the parameters associated with a network to be systematically adjusted until a satisfactory performance was achieved. From this point in history, research into the area of ANNs increased and the next major advance occurred in 1958 with Rosenblatt [75]. This work introduced the *Perceptron*, with the possible functionality of a single layer network[10] of perceptrons, taught using the *Perceptron Learning Rule*, was investigated. Classification problems were successfully solved by a single layer perceptron network. Widrow and Hoff in 1960 [76] proposed a neurone model called the *Adaline* (Adaptive

[10] Meaning a network where the input to the network is connected to the same neurones as the output from the network.

Linear Element) which is similar is many respects to the perceptron but is taught using a different algorithm called *Least Mean Square*. Research into perceptrons, adalines and other ANN models bloomed for many years. These networks were able to segment the weight space, by linear discriminant functions, into regions which allowed classification. In 1969 Minsky and Papert highlighted the limitations of single layer ANNs [77], where the XOR problem was proved mathematically to be unsolvable. They also postulated that none of the desirable qualities so far exhibited by single layer networks would also be exhibited by multi-layer networks. The XOR problem is now the 'standard' or 'benchmark' problem that is frequently used in ANN research. For many years after this publication research in the area of ANNs reduced, more accurately they became 'unfashionable'; this slowed progress significantly.

Developments in self-organising ANNs occurred during the 1970's and 1980's [78, 79] which later gave rise to ANN paradigms such as ART (Adaptive Resonance Theory). Hopfield in 1982 published work [80] which described a new ANN paradigm called the Hopfield Network, these networks were to become one of the major classes of ANN. Kohonen, also in 1982, published work [81, 82] which proposed a new ANN paradigm which learnt using an unsupervised (or self-supervised) algorithm, these ANNs were called SOMs (Self Organising Maps) and have also become popular with derived types such as LVQ (Learning Vector Quantisation) being developed as classifiers.

A major breakthrough in ANNs occurred in 1986 with a publication by Rumelhart, Hinton and Williams [83] which described a learning algorithm now known as 'Back Propagation' (or just BP for short). The main reason why this discovery was important was because until this time there was no effective algorithm for training a perceptron (or similar neurone model) network that had more than one layer of adaptive weights. The famous XOR problem was no longer unsolvable by ANNs. It was also from this discovery that multi-layer perceptron (MLP) ANNs became the most popular ANN. A further review of literature concerning BP reveals two further facts; firstly two other publications describing BP were published at around the same time independently [84, 85], and secondly that BP was described as early as 1974 in a PhD thesis [86], which

unfortunately remained unknown for over a decade[11].

In 1988 the first popular alternative to the MLP was discovered by Broomhead and Lowe [87]. A new network based on neurones which were modeled using radial basis functions (RBFs) was described, these ANNs have been applied to numerical analysis and linear adaptive filters, amongst others. Recently RBFs have become popular in control engineering applications where the advantages of RBFs over MLPs are significant.

The number of publications in the area of ANNs has grown significantly, especially since the discovery of BP. The historical review earlier in this section indicated only the major, initial, publications. Many of these publications were followed shortly after with further important research. The number of books and journals available is now enormous[12]. Luckily, there have been numerous quality publications describing the application of many ANN types to electronic nose technology. The first published application of ANNs to electronic nose technology occurred in the early 1990s [46, 88] with the use of BP trained MLP ANNs.

Figure 2.5 shows how the field of ANNs is sub-divided depending on ANN architecture and training algorithm, such categorisation aids the researcher into what paths of investigation have a high probability of success.

The following subsections describe previous ANN research in the field of electronic noses based upon the type of ANN employed. In each case the relative performance of the ANN is reported.

Multi-layer Perceptron

By far the most popular arrangement of neurones within an ANN is the MLP. This is also reflected in the amount of published research where MLPs have been applied to electronic nose technology [46, 88]. This type of ANN was the first to be applied to

[11] Perhaps current writers of PhD theses and their institutions should take note!

[12] A recent valuable source of ANN information and literature is the 'Internet', the FAQ (Frequently Asked Questions) published regularly on USENET: comp.ai.neural-nets and archived at: ftp://ftp.cs.cmu.edu/user/ai/pubs/news/comp.ai.neural-nets and http://asknpac.npac.syr.edu, is well worth reading.

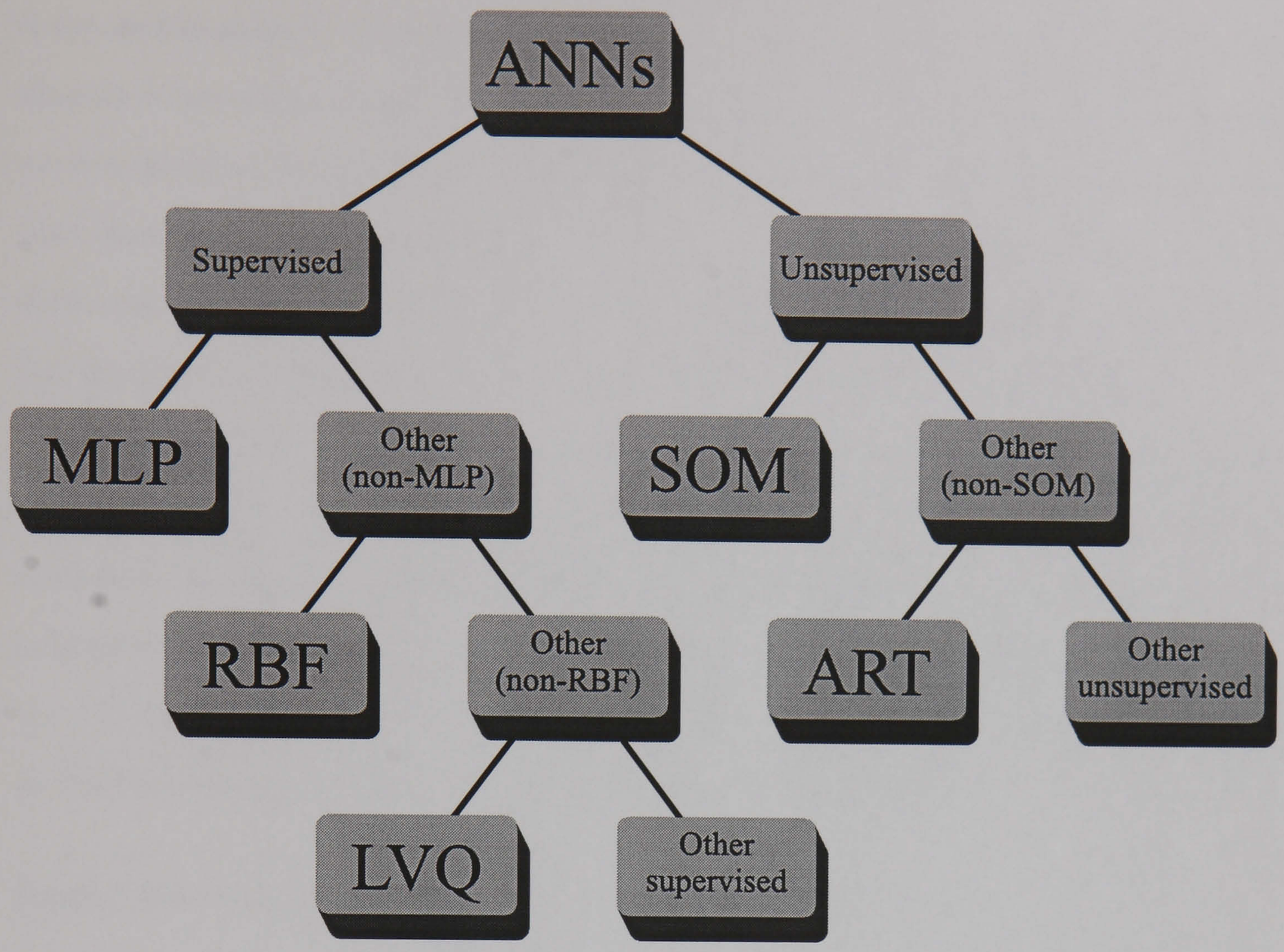


Figure 2.5: Categorisation of the major artificial neural network types.

electronic noses. There are many different 'flavours' of MLP ANNs and BP training algorithms, some of these have previously been applied to classification of odours from an electronic nose. The first 'flavour' of MLP that most researchers have employed can be termed as 'vanilla'. This is a standard, feed-forward, arrangement of neurones trained using standard BP which is described within the book "Parallel Distributed Processing: Explorations in the Microstructure of Cognition" by Rummelhart and McClelland [83]. Much ANN research has been performed by applying 'vanilla' MLPs in order to establish a benchmark performance by which further classification techniques were compared. Gas mixtures have been analysed using MLP ANNs [89], 'vanilla' MLP ANNs were employed and the effect of changes in internal parameters upon classification performance was investigated. Other similar works have been published; enhanced versions of MLP have been investigated when applied to the analysis of pure gases and complex odours [90, 46, 88]. There have been many more publications of this type,

there are too many to cite individually; however, it can be observed that there is a large amount of repetition where, firstly 'vanilla' MLPs were shown to perform better than more established statistical based methods, or secondly empirical parameter derivation were described. The impression from this is that each set of researchers (i.e. each electronic nose project team) are at the same stage in ANN development and the field has not matured sufficiently for diverse ANNs to be significantly researched.

Even if the ANN paradigm employed is of only one type, there are many parameters that need to be empirically derived, such as training parameters, network topology and estimation of classification performance. These parameters, according to the results published so far, seem to be highly dependent on the application. This limits the relevance of published material, therefore the review was biased towards research performed at the University of Warwick or with similar sensor technology.

Radial Basis Functions

Radial basis function (RBF) ANNs are popular in the field of control engineering. Their application to electronic nose data has been limited. An ANN inspired by RBF has been applied to surface acoustic wave (SAW) sensor arrays [91]. Research has recently been published which applies RBF ANNs to electronic nose technology [92]. There is a wealth of material published about RBF ANNs [59, 60, 55]. RBF ANNs have been found to be superior to MLP ANNs in some respects, firstly the training process for RBF ANNs is more reliable, secondly they only need one hidden layer of neurones, and thirdly in some cases they can generalise better[13].

Self-Organising Maps and Learning Vector Quantisation

Self-organised maps (SOMs) and Learning Vector Quantisation (LVQ) ANNs were discovered by Kohonen [81, 82]. They have been applied to a wide variety of problems including classification of odours [93, 94]. SOM ANNs have proved to be a valuable tool for analysis of the information content within a data-set. It is not itself a classifier but

[13]This depends on the quality of the data. MLP ANNs can interpolate between missing data points better than RBF ANNs because of the 'global' approximations that result from training.

a MLP stage can be appended in order to construct a classifier [94]. SOMs have been employed in order to visualise the effects of various data pre-processing techniques and to identify the source of noise. Further, SOMs have shown to be better in some cases for adapting to, and therefore becoming tolerant of, noise effects such as sensor drift. SOMs are more closely related to the neural structures found in the human olfactory system than other ANN types such as MLP and RBF. The olfactory sensor neurones in the human nose are subject to considerable drift and time varying behaviour, therefore SOMs have shown some success in this area [95]. Workers have experienced problems when applying SOMs, notably their highly empirical design and lack of control during the unsupervised learning phase.

A development from the SOM is the LVQ ANN, which is a classifier and undergoes a supervised learning phase, and which has been applied with some success to electronic nose data [96]. In this work, as well as MLP and LVQ ANNs, fuzzy learning vector quantisation (FLVQ) was employed, which is a 'flavour' of LVQ with the incorporation of fuzzy set theory (see section 2.6.3). FLVQ was developed by the authors and shown to yield better performance than LVQ and MLP ANNs when applied to the classification of whiskey odours. It should be stressed that the MLP ANN employed was a 'vanilla' type and was not enhanced in any way. This research does however show that LVQ (or 'flavours' of LVQ) ANNs have potential.

Other Network Types

There have been a small amount of published material detailing the application of ANN types that do not fall into the previous categories.

The performance of recurrent ANNs such as Elman and Jordan have been compared to 'vanilla' MLP ANNs [97]. These recurrent networks were found to perform at best only marginally better than 'vanilla' MLPs, it is possible that an enhanced MLP could outperform all the other types. The problem may not be that recurrent nets are unsuitable for application to electronic nose data, but that recurrent networks require more attention to data pre-processing and overall implementation. Of all the ANN types recurrent networks are most similar to the networks found in the olfactory bulb

(see section 2.3.2). In an effort to utilize the dynamic information within electronic nose data, time-delay networks were employed along with recurrent networks to simple volatile vapours [56]. Time-delay networks performed favourably compared to recurrent networks for this particular application.

Hopfield networks, more precisely Boltzmann and Hamming networks, have been applied to the classification of alcohol and hexane vapours [48] and compared with the performance of classifiers based on statistical cluster analysis. Cluster analysis exhibited superior performance, especially on simple alcohol vapours such as methanol, ethanol and isopropyl alcohol. The poor performance of ANNs was blamed on poor data quality.

Adaptive networks such as ART show great promise. Although ART has only recently been applied to electronic nose data [98], it shows great promise because it allows new odours to be learnt without having to completely rebuild the network. Within the ANN community, ART has received great criticism for lack of statistical foundation [62].

2.6.3 Fuzzy Set Theory

Fuzzy set theory is a set of rules by which a set of numerical values are transformed into or out of a fuzzy domain. Fuzzy theory is well described in many publications [99, 100] and the theory can be applied to many classification techniques [101]. It has been proposed that fuzzy sets enable performance of classification of odours to be improved because the signal conditioning that occurs during 'fuzzification' and 'de-fuzzification' translates many properties of overlapping sensor arrays into parameters that are better handled by classifiers [61]. A discussion of the role of fuzzy logic within the field of electronic nose technology has previously been published [102], where it was concluded that neuro-fuzzy systems were a possible advance towards the goal of an intelligent sensor system. Fuzzy sets have been applied to LVQ ANNs [96], and to MLP ANNs [103]. Fuzzy reasoning has been used in conjunction with MLP ANNs (as opposed to a modified MLP) to classify odour groups (inflammable gases, fragrant smells and offensive odours) [90]. In this study, fuzzy reasoning enabled odour groups

to be distinguished with 100% accuracy. Fuzzy sets are currently popular and although there is less published material than ANNs in general they are possibly the fastest growing area of classifier development within electronic nose technology.

2.7 Previous Applications

This section highlights some of the novel applications which have employed an electronic nose, and an appreciation of the relative complexity of the application that composes this PhD can subsequently be gained. The field of electronic nose technology is still relatively young, having only been the subject of significant research in the last 15 years. Consequently the variety of applications of electronic noses has grown at an almost exponential rate.

The range of types of application of electronic noses is constantly expanding. In many industries today (for example the food, brewing and perfume) the primary method of assessing the flavour and/or smell of various substances is still the human nose and so had been the target of research in the area of electronic noses which has been performed for the past 15 years or so. It is therefore the area of beverage or foodstuff production which is responsible for the majority of electronic nose applications.

At the University of Warwick, the electronic nose has been applied to the grading of coffees [45] and to determining the roasting level of coffee beans [51]. Also, beverages such as whiskey, lager and bitter have been graded [45].

The perfume industry is an obvious target for electronic nose applications [104]. Air quality has been estimated [45], and more specifically, the presence of dangerous gases in the air within mines has been detected using an electronic nose [105]. The detection of boar taint has been reported [106] (the release of volatile compounds that exist within pork fat). Irradiated tomatoes have been screened [17] where CO_2, O_2, water vapour and ethylene were detected. These substances are released from the tomato cells when they are stressed (which occurred during handling). Wine recognition has been previously performed [61], only two different wines were analysed but the results were conclusive with discrimination being easy.

Grain odour resulting from organism activity has been used as the subject for electronic nose application [8, 107]. The organisms mainly responsible for grain odour are insects and fungae, such as moulds. This work is interesting because a large amount of data was collected (235 samples) which allowed many different types of pattern recognition to be investigated, they achieved 90% classification accuracy by employing a 'vanilla' MLP ANN (for two classes: good and bad).

2.7.1 Smelling Bacteria Odours

There are a number of publications which detail the analysis of bacteria odours. However these studies do not treat the odours as being bacteria odours but as being odours such as 'off-flavour', 'spoilt food' and 'food freshness'. In reality, many odours that we perceive as originating from inanimate objects are the result of the metabolism of micro-organisms. An example of this was the analysis of fish freshness [45]. Micro-organisms grow on fish, and the level of perceived freshness depends on the amount of micro-organism activity. If fish is frozen, the activity of the micro-organisms is virtually halted and the shelf life of the fish is extended. Given a particular type of fish, in this case haddock and cod, there is a very limited range of bacteria types that are responsible for spoilage. The study therefore analysed a limited number of bacteria growing in a consistent environment. Also in this study only three fish of each type were used, the relevance of the study to the fish industry as a whole is therefore limited. The principal of fish freshness estimation by electronic nose was proved to be feasible.

The off-flavour of lagers has been estimated using an electronic nose [6]. This off-flavour is caused by the action of micro-organisms on the beer, acid producing micro-organisms (sometimes bacteria) contaminate the beverage. The brewing process in the beverage industry is highly developed and specially bred yeasts are employed that yield a high quality product with low production costs. The activity of these yeasts, under the normal brewing process, is highly consistent and therefore the odour given off by a satisfactory product is also highly consistent. A deviation from the normal odour is therefore simple to distinguish.

Milk quality is related to the acidic products which build up as a result of micro-

organism growth. Fermentation of milk has been monitored using a biosensor array [108]. Similar to beverages, the fermentation of milk is highly controlled with only one yeast being involved.

More closely related to this research project (in that bacteria living on/in humans are involved), is work that was published that described the monitoring of body odours [18]. The flora that exist in the armpits cause body odour, and the effect of deodorant products upon this odour was investigated. This work did not attempt to distinguish between different odours but described the variance in odour concentration. If the quality of odour was effected as well as strength then this would have been missed in this investigation.

The odorous products of pig slurry have been analysed [109], the research was significantly simplified by creating an artificial odour. Pig slurry was analysed initially using gas chromatography (GC), the major individual odorants were identified. From this information, the odour was reconstructed by reconstitution of an aqueous solution where each major odorant was present in similar relative concentrations. This odour reconstruction allowed a more stable odour to be analyse by the electronic nose, which in turn reduced noise within the subsequent data. Visualisation of the odour data was accomplished by using a statistical based clustering technique called Sammon Mapping.

The closest published research was only recently published last year [110] (i.e. during the present PhD research). This work analysed the odour from 12 different microorganisms (including some types employed in this research). The odour sampling method was to place a Petri dish containing the bacteria to be sampled in a plastic bag (previously filled with air) and leaving it for 2 minutes. The air was then pumped out of the plastic bag and into an electronic nose. For some classes, all the vectors were resulting from 1 bag of odour (i.e. all measured on the same occasion). The resultant data set was also imbalanced with some classes having 48 vectors and others only 6. A 'vanilla' MLP ANN was employed to classify the odours, the network had 112 inputs and 13 outputs (one for each class). Although the results seemed impressive[14] the methods

[14]Some classes achieving 100% classification accuracy using cross-validation classifier performance estimation

employed do not have any substantial statistical foundation. The effect of culture age was ignored. This research can only therefore be considered as mildly interesting; other research although having a less similar application, was of more relevance to this research.

Chapter 3

Electronic Nose Development and Initial Data Collection Experiments

This chapter describes the methods of data collection including the design of an automated sampling sub-system, the procedures for testing and characterisation of the apparatus and the procedures employed for data collection itself.

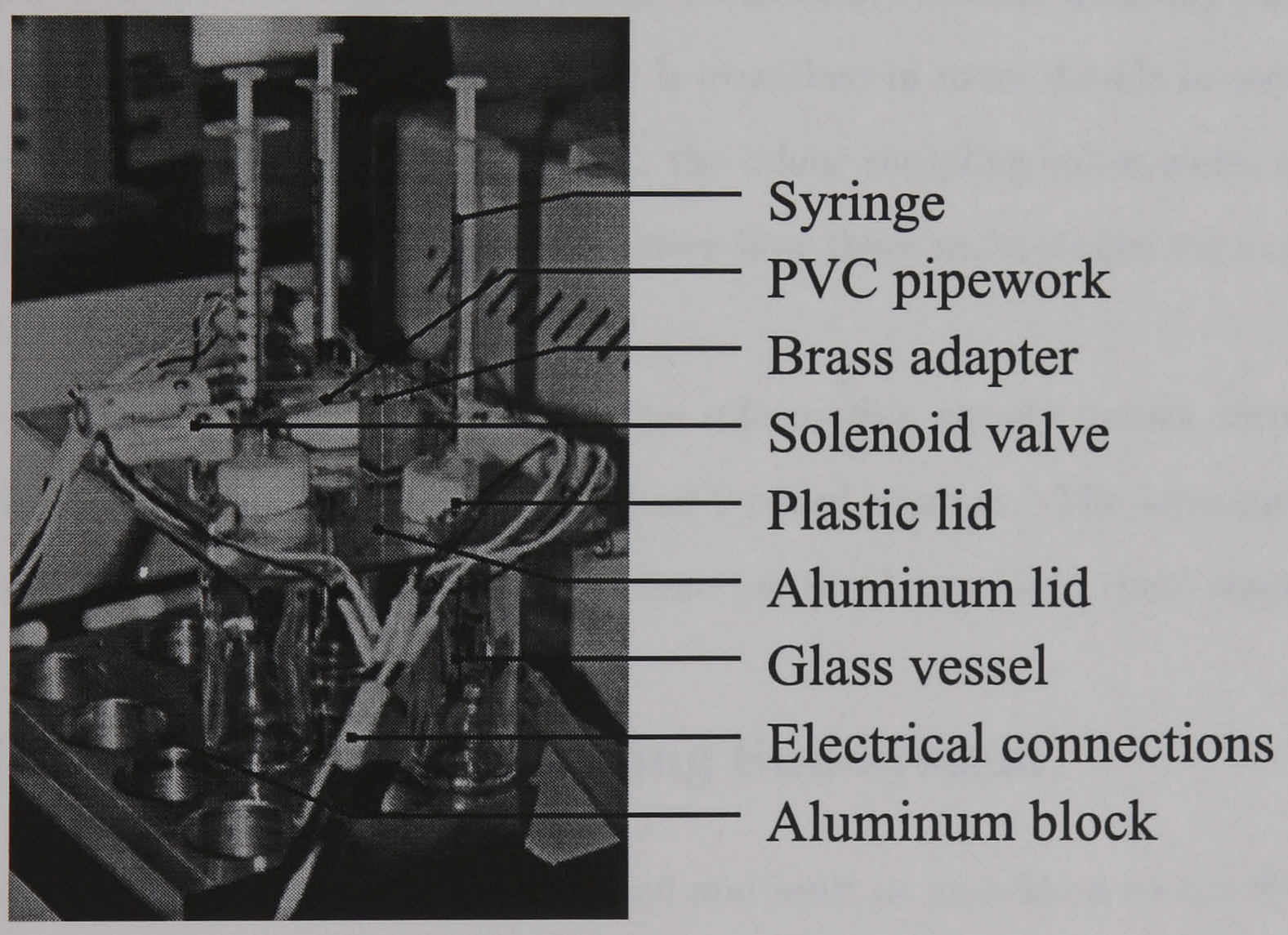
The purpose of data collection was to record signals from the sensors in the electronic nose and store them as data-sets in files[1] on computer disc (i.e. non-volatile storage). Data file storage allowed subsequent analysis and processing to be performed more easily.

3.1 System Overview

Before describing individual system elements (sub-systems), it is necessary to give a brief description of the complete system in order to illustrate the role of each element. The photographs shown in figure 3.1 show the apparatus used for data collection in the Microbiology Laboratory in the Biological Sciences Department at the University of Warwick. These photographs show how the system was constructed, the size of the

[1]The files were in text format where numerical values were represented as ASCII strings.

(a)



(b)

Figure 3.1: Photograph of (a) whole system (scale 9.3:1)and (b) the valve/vessel assembly (scale 2.4:1) within the Front-end, used to collect data in the Biological Sciences Department at the University of Warwick.

system (and therefore its portability), and the environment in which it was used.

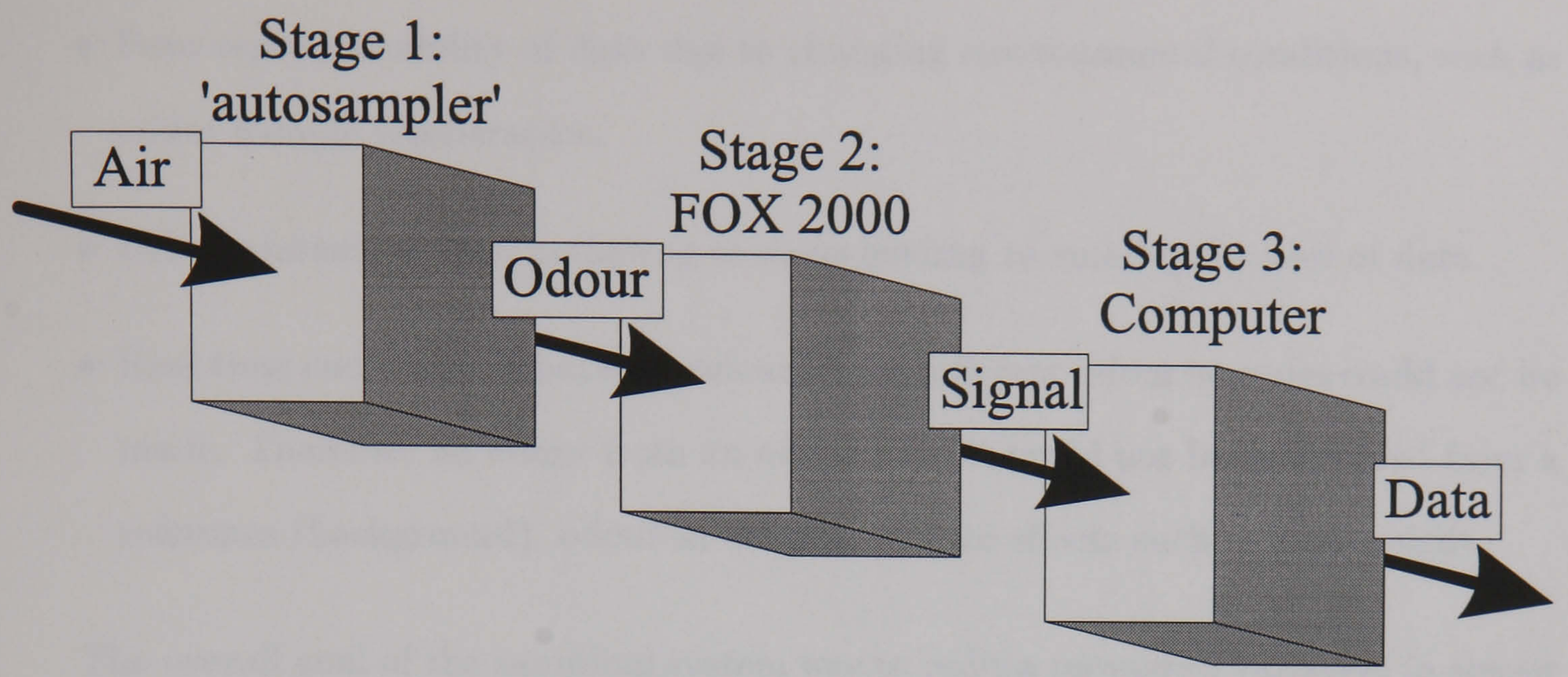


Figure 3.2: Schematic diagram of apparatus showing the three major stages.

The system centered around a FOX 2000 unit. This was a commercial electronic nose which conssisted of a gas sensor array, temperature sensor, humidity sensor, vacuum pump and interfacing electronics. It is described in more details in section 3.4. The system consisted of three main stages: the odour sampling sub-system, the FOX 2000 unit, and the computer. Figure 3.2 shows how these main stages were combined to form the complete system.

From figure 3.2 it can be seen that the information simply passes through the system, from one stage to another, in a feed-forward manner. The structure of the description of the apparatus is based on a function breakdown into these stages.

3.2 Design of Odour Sampling Sub-system

An odour sampling sub-system was designed and built as an add-on to the Electronic Nose. This sub-system was called the 'autosampler'. The purpose of the odour sampling sub-system was to control the delivery of gases to the sensor chamber in the FOX 2000. This was necessary in order to address problems that had compromised performance in previous work [52]. Those problems may be summarised as:

- Poor reproduceability of data due to a significant variance in several sampling parameters, such as the time the sensor array is exposed to the sample.

- Poor reproduceability of data due to changing environmental conditions, such as odour sample temperature.

- Labour-intensive data gathering sessions leading to small quantities of data.

- Real time comparisons between odours from different odour samples could not be made. Therefore an odour from an odour sample could not be subtracted from a reference (background), odour in order to reduce effects such as sensor drift.

The overall goal of the sampling system was to reduce unwanted variation in sensor signals. This was achieved by implementing the following features:

- Computer control of the gas flow using solenoid valves to switch gas flow through one of several vessels, each containing an odour sample.

- Filtering of the air input to the sub-system in order to reduce sample contamination due to external air-born contaminants, such as pollen or fungal spores.

- Using a water bubbler to condition the air that was input to the system through water in order to reduce evaporation from the odour samples (which were aqueous solutions), and to control humidity.

- Placement of sample vessels (and associated gas flow fittings) within a temperature controlled environment, ($\pm$ 0.1°)C.

- Automated data gathering using software that interfaced to the sensor array within the FOX 2000 and autosampler.

Figure 3.3 shows a schematic representation of the autosampler showing the gas and electrical connections. Gas flow was directed through one of three routes by the action of six solenoid valves (Lee Company LFAA1200118H). Each vessel had a pair of solenoid valves associated with it, one on the input side and one on the output side. When power was not applied to a particular pair of valves, the associated vessel was isolated from the rest of the sub-system. When power was applied to a pair of valves the associated vessel was fully connected to the rest of the sub-system and gas flow

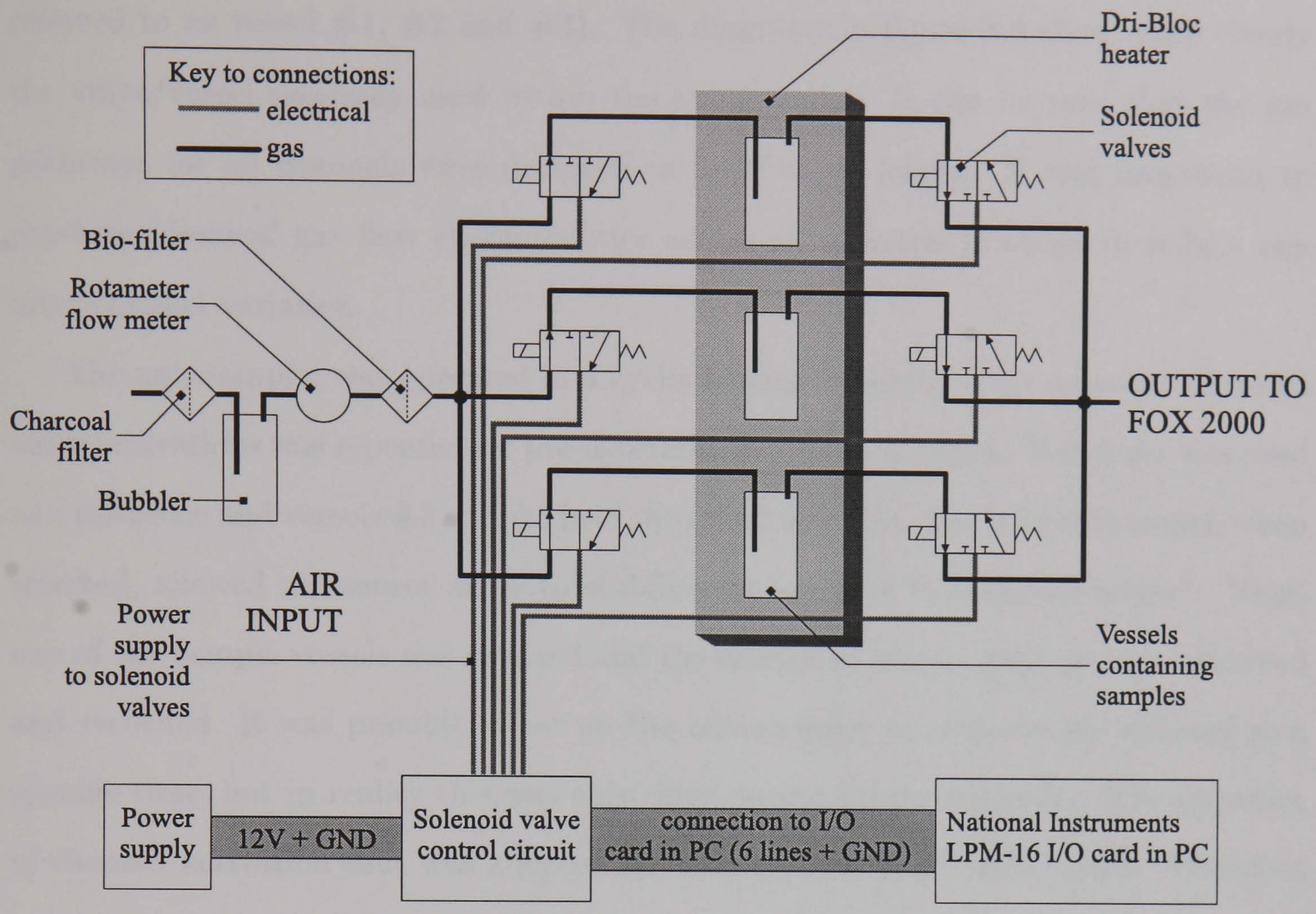


Figure 3.3: Schematic diagram of the 'autosampler' showing gas and electrical connections.

occurred via this route. At any one time, only one pair of valves were powered up. The gas connections to each valve were that the input was connected to the 'common' terminal and the output was connected to the 'normally closed' terminal, the 'normally open' terminal was blocked off (this caused the valves to act as a simple on/off control). Control of the valves was by means electronic signals[2] being applied to each valve via an interface circuit, which was in turn connected to an I/O card in the PC (National Instruments LPM-16). Software run on the PC ultimately controlled the solenoid valves and therefore controlled gas flow (the software is described in greater detail in appendix A). This sub-system therefore had three pathways (called channels) for gas to flow from the input to the output with one vessel per channel (these channels are subsequently referred to as channels #1, #2 and #3. Their corresponding vessels are

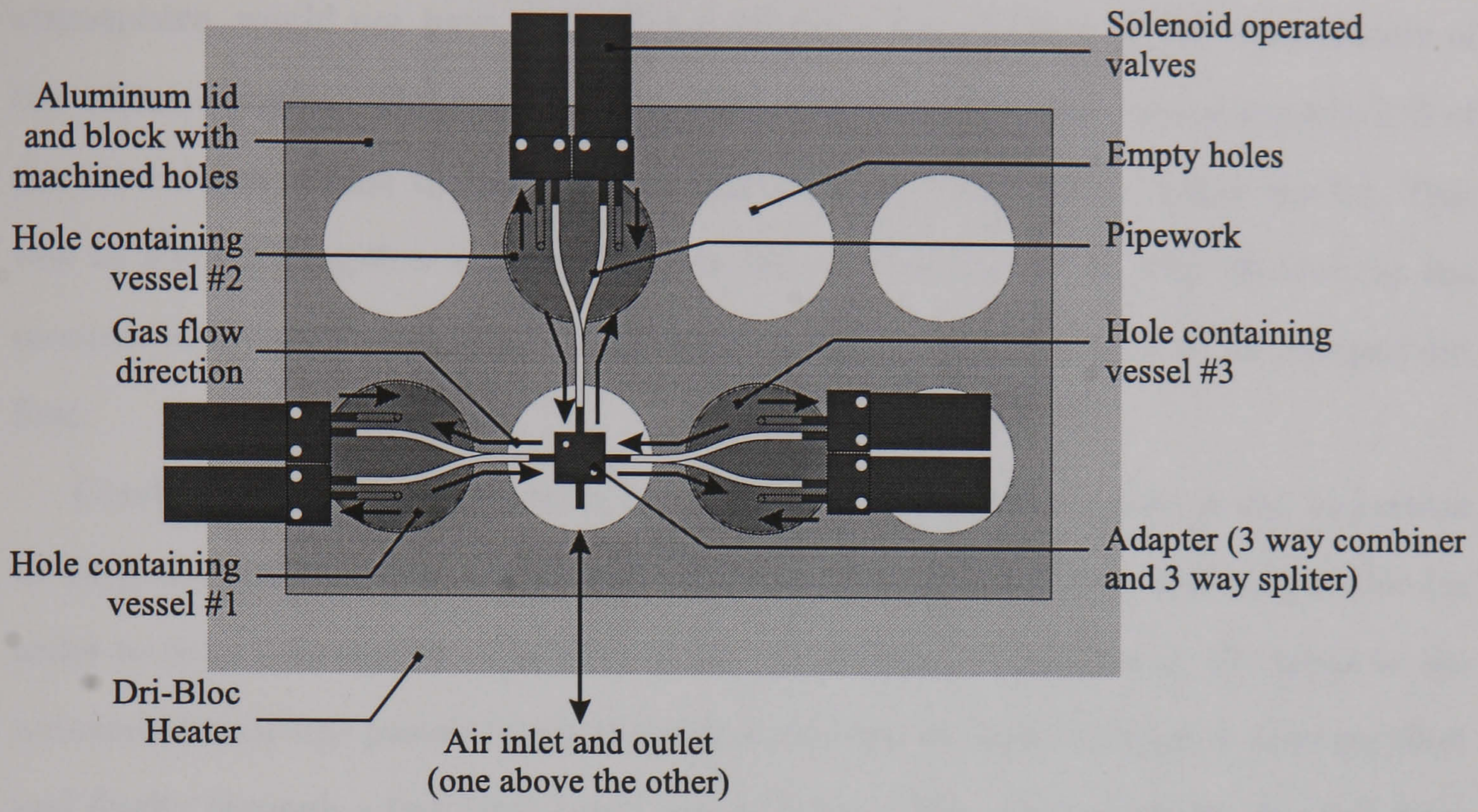
[2]The signals to the solenoid valves were actually power supply connections rather than a control signal separate to the power supply.

referred to as vessel #1, #2 and #3). The diagrams in figure 3.4 show more clearly the valve/vessel assembly used within the autosampler. It can be seen that the gas pathways for all channels were designed to be of equal length. It was important to preserve identical gas flow characteristics across all channels in order to reduce any inter-channel variation.

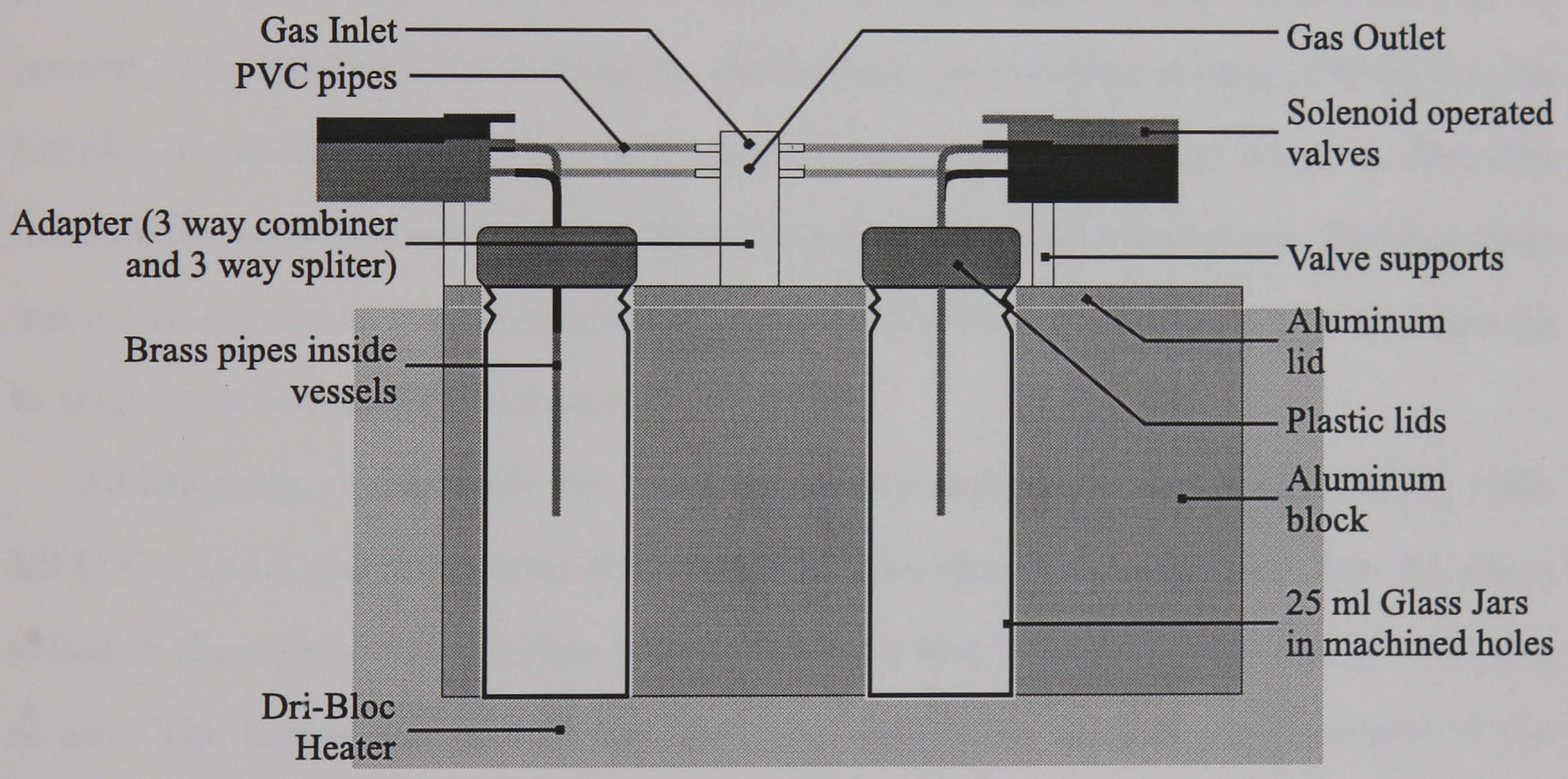
The autosampler was operated in a cyclic fashion, whereby a set sequence of timed valve activations was repeated for pre-determined number of times. Vessel #1 was used as a reference and vessels #2 and #3 held the odour samples. The reference vessel, when selected, allowed the sensor array to stabilise its response to a known odour[3]. Next, one of the sample vessels was selected and the change in sensor response was observed and recorded. It was possible to set up the autosampler to activate any channel at a specific time, but in reality this was only done during checks for faults. The sequence of channel activation that was adopted for each cycle was; #1-#2-#1-#3. Therefore each odour sample was measured, or 'smelt', once during each cycle. The action of activating the reference channel (#1) followed by one of the other channels (#2 or #3) was called a 'smell' and represented one complete measurement of an odour sample. One cycle represented a 'smell' of sample #2 and a 'smell' of sample #3.

The vessels were placed into an aluminium block in a DRI-BLOCK heater (model DB-2p, manufactured by Techne Ltd.). The DRI-BLOCK heater is a commercial product in which the target temperature is set by means of external trimmers to ± 0.1°C, and the internal heating element is powered by a PI controller in order to maintain the target temperature. In the top of the heater was a large inlet, into which apparatus was placed (in this case a machined aluminum block). In the aluminium block holes were machined that were just big enough to accept a glass vessel (a vessel was a 25 ml glass jar which is commonly employed in microbiology research), only the plastic lid of the jar protruded out of the top of aluminium block which acted as a heater jacket. This meant that the heater target temperature was set and the bacteria samples in the glass jars were maintained at the target temperature (± 0.1°C). It was

[3]In some experiments the vessel in the reference channel was empty and so the reference gas was laboratory air, technically this is still an odour because air contains small amounts of odorants.



(a)



(b)

Figure 3.4: (a) Plan view and (b) cross-section view, of the valve/vessel assembly used within the autosampler. Electrical and external connections have been omitted for clarity.

assumed that the heat conduction properties of the aluminium block were such that cooling effects on the block of external environmental factors, such as the laboratory

atmosphere, would not have a significant effect on the stability of the temperature of the odour samples. The input pipe to each vessel extended to approximately 2/3 of the total depth, whilst the output pipe extended very little (2mm) below the lid. This was to allow a plug-flow characteristic within the vessel, this in turn allowed, at the moment of the particular vessel being selected, the static head-space to be pumped out first.

Conditioning of the air entering the system was necessary because it was important to keep the environment in which the bacteria were growing as stable as possible (in order to reduce unwanted variability in the odour data). Therefore on the input to the autosampler air was passed firstly through a bubbler, secondly through a charcoal filter and finally through a fine filter (mesh size of 5 μm). The purpose of the charcoal filter was to reduce the amount of air-borne contaminants entering the system, such as dust, pollen and fungal spores and also to absorb volatile organic compounds that may be present. The bubbler helped maintain the humidity at a constant level. Obviously, the bubbler increased the humidity of the gas entering the system and this was desirable because the aqueous bacterial growth medium was prone to evaporation. The fine filter was a last barrier to stop contaminants entering the system, the mesh was fine enough to stop most common particulates.

Additionally, in line with the input to the autosampler, a Rotameter (series 1100, MFG Co. Ltd.) gas flow meter was installed. This device measured gas flow by channeling it through a vertical glass pipe, in which a metal indicator floated on a column of air. The higher the rate of gas flow, the higher the vertical displacement of the metal indicator. This particular Rotameter was capable of measuring gas flow up to 1.0 litre min^{-1} with an accuracy of 0.02 litre min^{-1}. The main reason for employing the Rotameter was to check for air leaks in the autosampler. For example, when the Rotameter showed a fall in flow-rate but the flow meter within the FOX 2000 showed no change in output, it was found that air leaks in and around the vessels was the cause.

The 'autosampler' was under control of an IBM AT compatible personal computer (PC)[4], via a control circuit, details of the computer system and software are given in section 3.3. The control circuit existed as electronic circuit which was separate from the computer. The circuit itself was simple in design, and figure 3.5 shows the sub-circuit for each valve. The sub-circuit was based on a design recommended by the solenoid valve's manufacturer (Lee Components). The circuit was simply 6 identical sub-circuits, each sub-circuit had a separate input and output. The sub-circuit consisted of a *npn* transistor (2N3055, rated at up to 15 A collector current!) driven in the saturated region which therefore acted as a switch, the transistor was connected in common emitter mode. The power consumption of each valve was 1W maximum, therefore being driven at 11.4V from a 12 V power supply[5], each valve consumed 88 mA maximum. Each digital output from the LPM-16 I/O card was rated at 100 mA maximum, this is more than ample to drive the base of the transistor. An LED was connected, via a current limiting 220 Ω resistor to the base of the transistor. This gave a visual indication of the status of the circuit; LED lit for valve on and LED not lit for valve off.

Additionally it can be noted from the schematic diagram in figure 3.5 that there was a diode connected, across the solenoid valve terminals. The purpose of this was to allow a path for conduction (other than through the rest of the circuit) of current caused by back-emfs (called inductive 'kick') which result from the generation or collapse of the magnetic field within the solenoid when the solenoid was energized or de-energized.

The pipe-work between the valves and the vessels was brass piping with an approximate outer diameter of 1.5 mm and an approximate inner diameter of 1.2 mm. Brass can absorb a finite amount odorants as a result of chemical reactions on its surface. During the 'burn-in' of the system (described in section 3.5), which had a duration of several days, the brass piping was exposed sufficient to odorants that subsequent odorant absorption was negligible (i.e. The effect of odorant absorption during the data collection experiments was negligible). The other piping was PVC (Lee Company

[4]Manufactured by Viglen Ltd, with an Intel 80486DX33 cpu, 8MB of RAM, MSDOS 6.22 and MS Windows 3.11.

[5]When switched on, the collector transistor terminal was approximately at 0.6V, which is normal for a *npn* transistor, the voltage drop across the valve input was therefore 12V - 0.6V =11.4V

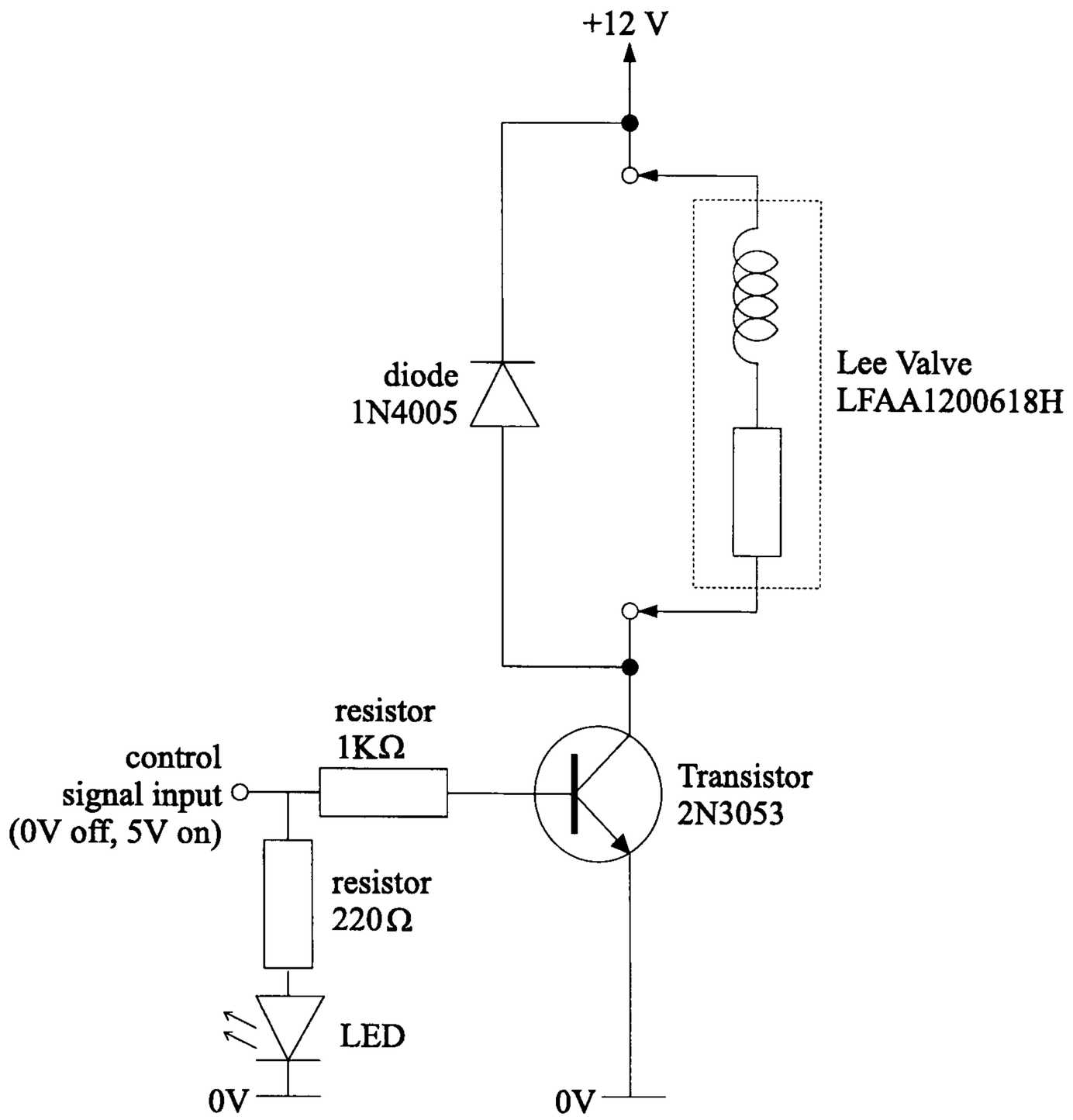


Figure 3.5: The electronic sub-circuit for control of a solenoid valve.

TUVA4220900A), which fit tightly over the brass pipes, inner diameter was approximately 1.37 mm (when unstretched). The small diameter of the pipe-work allowed a small 'dead-volume' and therefore a fast response.

3.3 Computer Control System

The computer (PC) was central to the system. It controlled the supply of power to the valves via a control circuit and sampled the sensor signals within the FOX 2000.

An LPM-16 I/O card was used to connect the computer to the valve control circuit and the FOX 2000. The LPM-16 I/O card allows both digital and analogue signals to be input to and digital signals to be output from the computer. In this case 6 of the digital output lines (and a 0 volt, i.e. digital ground line) were connected to the valve control circuit, that is one digital line per valve. Nine analogue LPM-16 input lines were connected to sensors within the FOX 2000, 6 to gas sensors, 1 to a temperature

sensor, 1 to a humidity sensor and 1 to a gas flow rate sensor.

Software was written using a software package called Labview. Labview is a language sold by National Instruments, aimed at scientists and engineers who wish to control and communicate with external measurement equipment. Labview runs within the Microsoft Windows graphical environment and allows program development via a graphical interface. Labview programs were interpreted by a Labview interpreter in this project[6], this meant that Labview had to be installed on the computer within the system (as well as the computer used for software development). Labview therefore allowed programs to be written that communicated with the LPM-16 I/O card and therefore communicated with the valve control circuit and electronic nose. The main program written was called the 'autosampler Control Program', this program combined the control of the gas flow and the recording of sensor signals from the FOX 2000. The overall purpose of the program was to allow the collection of large quantities of data from the FOX 2000 with the minimum of workload. The user was able to input parameters into the program's graphical user interface (GUI) that determined the duration that each valve was open, the number of times (cycles) that an odour was sampled and where (in which data file) the sensor signals and valve states was stored. Because the valve states were also recorded with the sensor signals it, allowed timing information to be extracted from the data files.

The computer software allowed, in theory, for a data gathering experiment to only need a worker to be present at the start and once the experiment was underway there was no reason for a to attend. Previous work [52] required a worker to be present throughout the entire duration of the data gathering experiment, and since this period of time was many hours human errors were inadvertently introduced. Until this system was employed, precise timed exposure of the sensor chamber to odour was difficult. All the Labview programs written for this project are detailed in appendix A.

The autosampler Control Program, via a user friendly screen (see Figure A.2), gave feedback to the user on the current status of the experiment by presenting information

[6]It is possible to compile the programs in order to make stand-alone applications by buying an extra add-on package

such as number of completed cycles, sensor readings for the previous few minutes and current valve status. The user could therefore check that the experiment was running smoothly at one glance. When the maximum number of cycles was reached the experiment was halted and put in the standby condition (detailed in section 3.6).

The FOX 2000 Electronic Nose is 'shipped' with dedicated software that is developed by the manufacturers specifically for performing data collection experiments. However this software was unable to control the valve control circuit and the work involved in incorporating this extra functionality was deemed to be greater than the work involved in developing a smaller more dedicated program, the later strategy was therefore adopted.

3.4 The FOX 2000

The electronic nose employed was based on an early design of the Alpha M.O.S. FOX 2000. This instrument, in its original form (before modification) consisted of a sensor chamber which contained six metal oxide gas sensors, a temperature sensor (LM35CZ) and a humidity sensor (MiniCap 2). On the output of the sensor chamber was a gas flow-rate sensor, a mass-flow controller (Parker 854TF) and a vacuum pump which exited to an exhaust fitted with a silencer. The Electronic Nose was connected to an LPM-16 I/O card within a PC via an external connector. Figure 3.6 shows the schematic layout of the FOX 2000.

The FOX 2000 contains some analogue op-amp interfacing circuitry, that converts the resistance of the gas sensors (and the signals from the other non-gas sensors) into a DC voltage (0 to 10V) for input to a computer. This circuitry additionally outputs voltages to the front panel circuit and connects to the gain and calibration controls. A photograph of the FOX 2000 is shown in figure 3.1a. On the front panel there were the following controls:

- Six numerical LED displays, 1 per gas sensor. These displays showed the voltage being output for each sensor. These displays allowed the user to check that the output from the sensors was within the range suitable for the I/O card (0 to 10

Figure 3.6: The construction of the Alpha M.O.S. FOX 2000 sensor array.

V).

- Six rotary controls, 1 per gas sensor. These controls altered the gain for the analogue interface circuit for each sensor, thus the magnitude of the output for each of the gas sensors could be controlled. If a particular sensor produced a voltage out of range (< 0 or > 10 volts), its corresponding control could be used to re-tune the output to the desired level. In practice, the base line output was set to be between 6 and 8 volts.

- Six heater controls, 1 per gas sensor. The heater in each gas sensor could be controlled, more details on the heaters within the gas sensors are given in later in this section. The controls were selectors which had three positions; 5 volts, 4 volts and 0 volts. The heater element of each sensor could be supplied with either 0, 4 or 5 volts. The normal heater voltage was 5 volts, supplying a gas sensor

heater at 4 volts resulted in the gas sensor operating at a lower temperature which altered its output characteristics. The heater could be turned off to save power if required. Heater controls were set at 5 volts at all times.

- Pump on/off switch. This switched the rotary vane[7] pump within the FOX 2000 either on or off. During the setup of an experiment it can be useful to momentarily turn off the pump. During this work the pump was left running at all times in order to keep the gas sensors at equilibrium.

- Pump purge/normal switch. This controls the voltage supplied to the rotary vane pump within the FOX 2000. After a measurement, the switch can be set to purge which increases the voltage to the pump. The resulting higher gas flow rate increases the rate at which the gas sensors recover from a measurement. In practice, the purge flow rate was not significantly higher than the normal flow rate, this switch was therefore set to normal at all times.

- Calibration controls. There were two calibration controls; the first selected one of the 6 analogue voltage output channels (i.e. 1 per gas sensor), and the second set one of several resistances. These controls allowed a calibration routine to be performed by setting the resistance for each channel. It was found that calibration, in this manner, was only needed once as previous op-amps and resistors were employed in the FOX 2000 circuitry.

Initially a graphite rotary vane vacuum pump (Type 122, Vacuum Pump Manufacturing Co. Ltd.) was employed (this is the supplied item). However, this was later replaced, after a number of initial experiments had been performed, by a diaphragm action vacuum pump (KNF Neuberger NMP30KNDC). This was because the former unit proved to be unreliable when pumping gas that had a significantly higher level of humidity than normal room air (due to the bubbler and aqueous odour samples in the autosampler). The problem was that within the rotary vane pump small amounts of graphite dust are produced as part of normal operation, and the high humidity levels caused condensation to form inside the pump causing graphite dust to form into a

[7] Now membrane pumps are used for better reliability.

sludge. This sludge consequently prevented the rotor from turning which could lead to poor performance and in the worst case lead to the electric motor (which was driving the rotor) burning out.

Other modifications to the standard FOX 2000 unit included the removal of the bio-filter which is in-line on the input to the sensor chamber. During characterisation tests condensation formed inside the filter (due to higher than normal humidity levels) causing an increase in resistance to gas flow, and this eventually resulted in gas flow being completely halted. Because the autosampler already performed filtration, the bio-filter was completely removed. It is interesting to note that the bio-filter used in the autosampler never suffered from this problem.

Alpha M.O.S. have recently started to market a new FOX instrument. This device differs from the old FOX 2000 in two main ways; firstly the analogue to digital conversion now takes place within the FOX 2000 and the computer interface is a RS232c serial connection from an embedded Motorola 68HC11 microcontroller, secondly the the new FOX series is able to use SAW sensors as well as metal oxides. The arrival of the new FOX was too late to be introduced into this project. Therefore at this time it was felt that the FOX 2000 was a more established model and should be continued to be used for the remainder of this project.

3.4.1 The Gas Sensor Array

The gas sensor type employed within the FOX 2000 was based on a semi-conducting tin oxide reactive element with a heater. There has been much electronic nose research performed at the University of Warwick employing commercial tin oxide gas sensors [45]. Tin oxide gas sensors have some desirable characteristics; firstly they are non-specific and have a broad range of sensitivity, and secondly they available with many different broad sensitivities (achieved by adding different impurities into the reactive element). Another advantage is that they are commercially available, initially the major manufacturer was Figaro Engineering Inc in Japan, and now more recently, Alpha M.O.S. have begun to market tin oxide gas sensors which are compatible (i.e. the same footprint and pin functions) with the Figaro devices. The gas sensors employed in this research

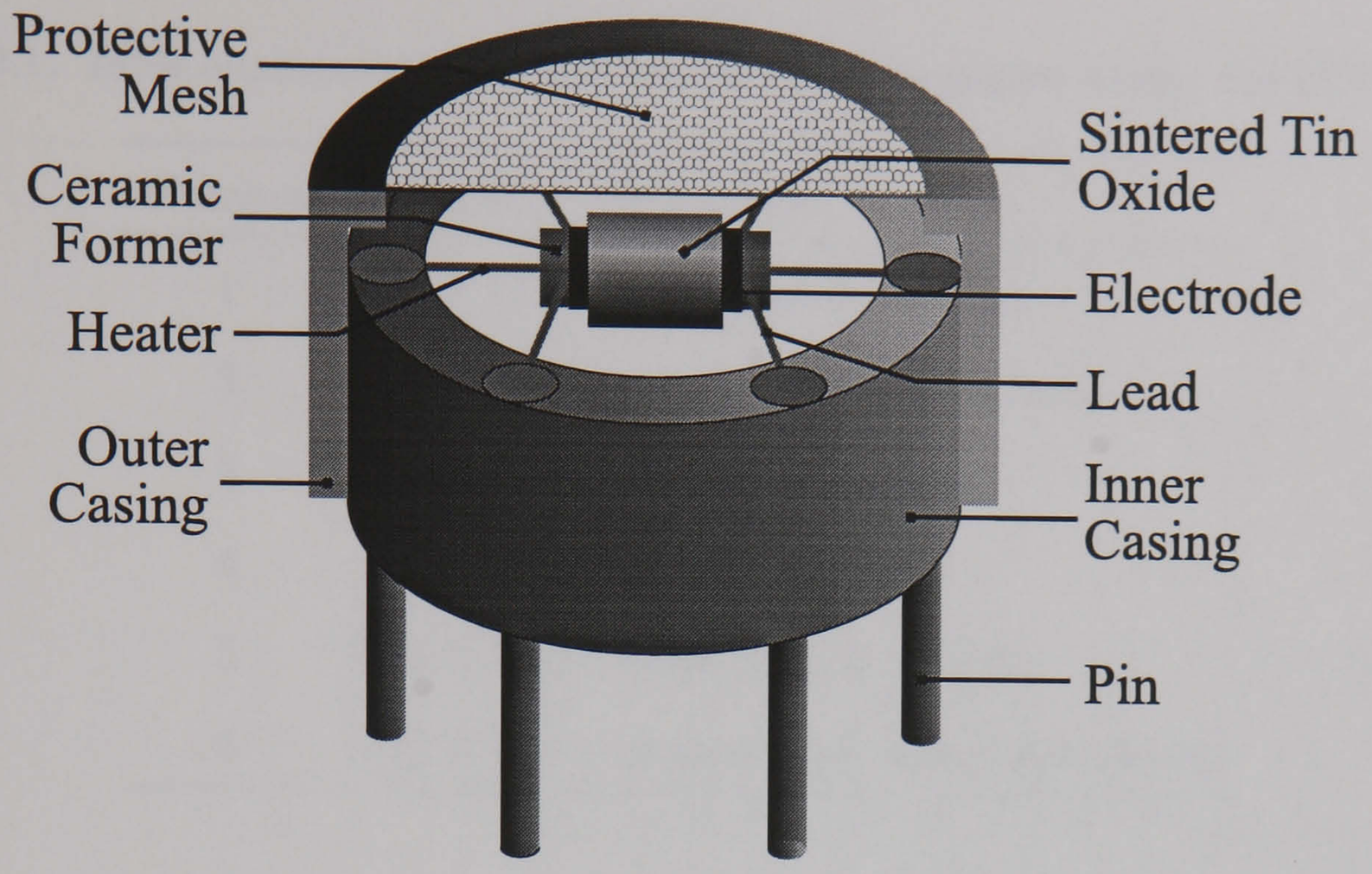


Figure 3.7: Cross-section of a typical tin oxide gas sensor showing the major components.

were supplied by Alpha M.O.S. An exhaustive description of tin oxide gas sensors is not given here but other works exist which treat the subject more thoroughly [111]. A diagram showing the construction of a typical tin oxide gas sensor is shown in figure 3.7, the major components are highlighted. The basis of the sensor is a cylindrical ceramic former through which a heater element passes. On the surface of the former, a layer of sintered tin oxide is deposited with an electrode at each end in order to facilitate electrical connection. The sensor element is then encased and connections made to pins suitable for mounting on a PCB. The relationship between the type of metal catalyst employed an the effect on the broad sensitivity is detailed in other work [111].

The broad sensitivity of a tin oxide gas sensor is determined by the introduction of metal catalysts within the tin oxide, usually by means of doping. The following description of the mechanisms of modification of the conductance characteristics of the sintered tin oxide element is based on material published in a previous book [45]. Tin oxides behaves here as an *n*-type bulk semiconductor[8]. When exposed to oxygen (usually the oxygen present in air), the oxygen is chemisorbed onto lattice vacancies in the non-stoichiometric semiconductor, one oxidation state is represented in equation 3.1.

[8] It behaves as *p*-type in H_2 atmosphere or under other conditions of temperature.

Table 3.1: Table summarising the gas sensor types employed within the FOX 2000.

Number	Model	Sensitivity
1	FIS P.10.2	non-polar compounds
2	FIS P.10.1	hydrocarbons and others
3	FIS T.30.1	polar compounds
4	FIS P.A.2	polar compounds
5	FIS T.70.2	alcoholic compounds
6	FIS P.40.1	heteroatom/chloride/aldehydes

$$[\,]_{site} + e^- + \frac{1}{2}O_2 \rightleftharpoons [O]^-_{site} \tag{3.1}$$

Where e^- is an electron abstracted from the conduction band. A depletion layer is created because the surface layer is depleted of electrons compared to the bulk semiconductor. Equation 3.2 shows the reaction that occurs when an odorant, R, is introduced to the semiconductor.

$$[O^-] + R \rightarrow RO + e^- \tag{3.2}$$

It can be observed from equation 3.2 that the previously chemisorbed oxygen reacts with the gaseous odorant. The reduction of sorbed oxygen causes a change in the amount of depletion and therefore of the conductivity of the tin oxide. Below 200°C water molecules react strongly with the surface of the tin oxide, and this affects the sensor response characteristic. In order to reduce reactions with water molecules, the tin oxide is heated to between 300°C and 500°C. Operating the sensor at a high temperature also speeds up the reaction of the tin oxide with odorants, so that typical response times are in the order to 10 s to 20 s. The heater causes the device, as a whole, to require more power than other, unheated, sensor types (such as conducting polymer). Table 3.1 lists the details of the 6 Alpha M.O.S. tin oxide gas sensors that were employed.

The choice of which 6 sensor models to use was partly determined by what was available and partly by the expected head-space of metabolites. The sensor models

used represented the widest range of sensitivities possible for the types of odorants expected.

3.4.2 Non-gas Sensors

In addition to the array of 6 gas sensors, there were also three non-gas sensors, these were:

1. The temperature sensor used was a LM35CZ. This device is a common temperature sensor and is widely available. It is manufactured by National Semiconductors. The LM35CZ is a 3 terminal integrated circuit temperature sensor which outputs a linear voltage of 10 mV per °C. The CZ version operates over a temperature of -40°C to +110°C. It has a typical accuracy (at 25°C) of ±0.4°C. It can operate from a supply of 4 V to 30 V and its typical quiescent operating current (supplied with 5 V) is 91 μA. The interface circuit consisted of a simple op-amp based amplifier with an adjustable offset.

2. The humidity sensor used was a MiniCap 2 (MC-2) Relative Humidity Sensor. This device is manufactured by Panametrics Ltd and is a general purpose thin film polymer capacitive relative humidity sensor. The dielectric constant of the polymer thin film changes with atmospheric relative humidity. The output is a linear function (rated at 1% accuracy) of relative humidity. This device can operate over a range of 5% to 95% R.H. and has an operational temperature range of -40°C to +180°C. Its capacitance at 25°C and 33%R.H. is 207pF (manufacturing tolerance is ± 15%). The interface circuit was an variable pulse width generator feeding into a pulse width to voltage converter. The width of the pulse is directly and linearly proportional to relative humidity.

3. Gas flow rate sensor used was a AWM3300V micro-bridge. This device is manufactured by Honeywell. Since the flow-rate was nominally constant during all tests and experiments, and also that a Rotameter was used on the input to the autosampler to measure flow rate, the exact characteristics of this device are not

of significant interest. However the output from this sensor was checked for large changes indicating gas leaks in the system, or pump failure.

3.5 System Testing and Characterisation

The system was first assembled in the Sensors Research Laboratory (SRL) at the University of Warwick, and a characterisation test (or 'dry-run') was performed. After these tests it was decided to perform the remaining tests and data gathering experiments in the Department of Biological Sciences at the University of Warwick. The reason for moving the equipment was that it was not possible to handle bacteria cultures safely in the SRL. However, performing the characterisation test in the SRL was beneficial because problems that arose were able to be dealt with much more easily because equipment and expertise was more readily at hand.

During the construction of the autosampler and the computer software, checks were made to ensure correct operation. For example, the autosampler was checked during building by electrical signals being applied to the solenoid valves and the control circuit. In other words the standard electronic fault-finding procedures were adopted. Much attention was paid to the methods employed during the building of the autosampler. The software was checked for bugs regularly during development, the Labview program development software provided comprehensive tools for program debugging.

The 'dry-run' tested all components of the system including the solenoid valves, their associated circuitry and air lines, computer software and the FOX 2000. A special Labview program was developed in order to aid testing, this program allowed direct manual control of the states of the digital output lines from the LPM-16 I/O card. This in turn allowed each solenoid valve to be controlled. Details of this program are given in appendix A.

By using the rotameter at the input to the autosampler in order to independently monitor gas flow through the system, air leaks were detected by comparing the rate of gas flow when different channels were activated. There were no odour samples within the glass vessels during this test. Air leaks in the system were found in channels

#2 and #3, which originated from the interface between the plastic lid and the glass body of the vessels. PTFE tape was then wrapped around the neck of the bottles to ensure an air-tight fit. The reason that channel #1 had no air leaks was that the corresponding plastic lid had a better seal (apparently some lids had better seals than others). However the same amount of PTFE tape was used on all vessels in order to preserve identical gas flow and odour characteristics on all channels. The flow rate through all channels was found to be 0.4 l min^{-1}. This flow rate was valid for all the experiments performed.

Next, 5 ml of 5% aqueous solution of ethanol was used as an odour source in each vessel. The progress of odour was monitored by observing sensor responses. Ethanol caused a strong response from tin oxide gas sensors because of its reactive nature, and as a result the sensor responses were well defined. Delay times of around 1 second were observed from the time a channel was activated to the time the gas sensor outputs started to change. The inside diameter of the pipe-work was small (1.37 mm), and hence the gas flow within the pipes can be considered plug flow. The time delay can be approximated to the time in which it takes for the odour to be propagated through the pipe-work (for a given channel) from the sample vessel to the sensor chamber, see equation 3.3:

$$T_d = \frac{V_d}{Q} \tag{3.3}$$

Here, T_d is the time delay, V_d is the dead volume (in this case the volume of pipe-work) and Q is the flow-rate. The total length of pipe-work was 55mm, therefore the volume of the pipe-work was 81.08mm^3. However, inbetween the head-space and the sensor chamber was a pre-heater (the volume of which was 62.8mm^3) and a pre-sensor chamber (the volume of which was 3455.75mm^3). Assuming plug flow in these components, the total dead volume, V_d, was 3599.63mm^3. The flow rate, Q, was 0.0067 litres s$^-$1. Using equation 3.3, the time delay, T_d, was calculated to be 0.537 s. This model assumes the mixing time within the gas chamber to be negligible, although this assumption cannot be safely made and in reality the time delay is greater than T_d. The original sensor chamber was rectangular, the dynamics of gas flow within this chamber

were dependent on many environmental variables, such as gas temperature, flow-rate, humidity level etc. It was deemed unnecessary to build a mathematical model that would describe the gas flow dynamics within the sensor chamber and a new design initiated which is described later on.

The autosampler Control Program and the FOX 2000 was tested by performing a 'dummy' experiment. The experiment was set up to run for 24 hours, each channel being activated in turn for a period of 10 minutes (this was not the usual 4 phase cyclic manner). The response from the ethanol solution contained in the vessels was stored as a data-set in a data file. The data file was examined using a program called Microsoft Excel, the signal profiles were consistent with correct operation of the system. Additionally, during these tests, the DRI-BLOCK heater was set to 25°C.

The experiments conducted at the Biological Sciences Department will be described in terms of experiment groups with procedural details being described as and when they were employed.

3.6 Initial Tests Performed At Biological Sciences

Firstly the complete system was assembled at the Biological Sciences Department. For the entire duration of these tests the system was left in the standby condition (even between tests), the purpose of which was to ensure that the gas sensors were at their equilibrium. The standby condition was that the FOX 2000 was powered up with the pump running, channel #1 was activated, vessel #1 was empty, the gas conditioning and filtration systems were in place. Excessive sensor drift can occur if gas sensors are used to analyse odour when the sensors themselves have only been powered-up for a short period of time prior to use. Sensor drift was an undesirable occurrence because it introduced unwanted variance into subsequent data (details in chapter 4). During the remaining experiments and tests the DRI-BLOCK heater was set to a target temperature of 36.8°C, this corresponds to body temperature and is the temperature at which bacteria in the human body normally grow.

The valve/vessel assembly up to the input to the sensor chamber, that is the pipes,

solenoid valves and vessels, were first cleaned. This was done by removing the assembly from the system and pumping through it a mixture of 75% ethanol 25% distilled water[9](also known as ethanol/water mix). Clean air was later pumped through the assembly in order to remove and ethanol or water vapour that remained after the mix was pumped out (and would cause unwanted variance in the sensor signals). The clean assembly was then re-fitted to the rest of the system.

The bio-filter was autoclaved. This process involved 'pressure-cooking' the filter in high pressure steam for one hour. The result of this process is that all contaminants (unwanted microbes) within the filter were killed. Autoclaving is a common process employed in the field of microbiology. Equipment used in micro-biology tends to be in one of two classes; that which is autoclaveable and that which is disposable (used once only). Unfortunately the data collection system fitted into neither class which meant that more the difficult methods of cleaning had to be employed.

After the cleaning phase, the system was left in the standby condition for 4 days. Then a test, called the 'burn-in test' because it 'burnt-in' the set of gas sensors, was performed for 22 hours and 20 minutes where one cycle was programmed to be 8 minutes in length, see table 3.2. Alpha M.O.S. recommend, in their documentation, that each measurement (cycle phase) should be no less than 2 minutes in length (i.e. the length of time that the gas sensors are exposed to an odour). This measurement time was therefore used in all subsequent experiments. The 'burn-in test' allowed the gas-sensors to monitored over a long period of time in order to check for stability in the new environment, all vessels were empty (i.e. there was no odour samples). It has been demonstrated that metal oxide gas sensors suffer from long term drift [45] and that reaching a state of equilibrium can take between 2 to 3 days.

The 'burn-in test' lasted for 170 cycles (translating as 81,600 valve operations). During this test ambient conditions, such as room temperature and humidity, varied. One reading was taken from each sensor every second, therefore an 8 minute cycle consisted of 480 measurements from each sensor, making a total of 4320 measurements from all 9 sensors. The signals from the gas sensors were analysed to investigate drift, the

[9]This particular mixture acts as an anti-microbial agent

Table 3.2: Table showing cycle times used for an 8 minute cycle time.

Channel Number	Duration On (minutes)
1	2
2	2
1	2
3	2
Total Cycles Time:	8 Minutes

relationship between temperature and gas sensor output, and the relationship between humidity and gas sensor output. Figures 3.8 and 3.9 show these relationships. Because all the vessels were empty, there was no significant change in gas sensor output during the cycle. The sensor value is the corresponding voltage output for that sensor that was input to the LPM-15 I/O card, i.e. it shows the 'raw' data-set.

The output voltage from a gas sensor at time t can be given as:

$$v_t = \mathcal{F}(g_t, t_t, h_t, v_{t-1}) \tag{3.4}$$

Where g_t is a parameter related to the chemical composition of the gas the sensor is being exposed to, t_t is the ambient temperature, h_t is the ambient humidity and v_{t-1} is the previous voltage output of the sensor (i.e. the sensor has a 'memory'). It can be observed from the plot in figure 3.9, that the variance in humidity appears not to have been significant.

However, analysing the plot in figure 3.8 shows that the variance in temperature is significant (a change of 6.45°C). A visually significant correlation between the output from the temperature sensor circuit and the outputs from the gas sensor circuits can be observed from this plot. Table 3.3 shows the results of correlation analysis on this data. It can be observed that there was high degree of correlation between the gas sensors. Also there was a relatively high correlation between the temperature and humidity sensors outputs and the gas sensors outputs. Negative correlation between two variables simply show that when one variable increased in value the other tended

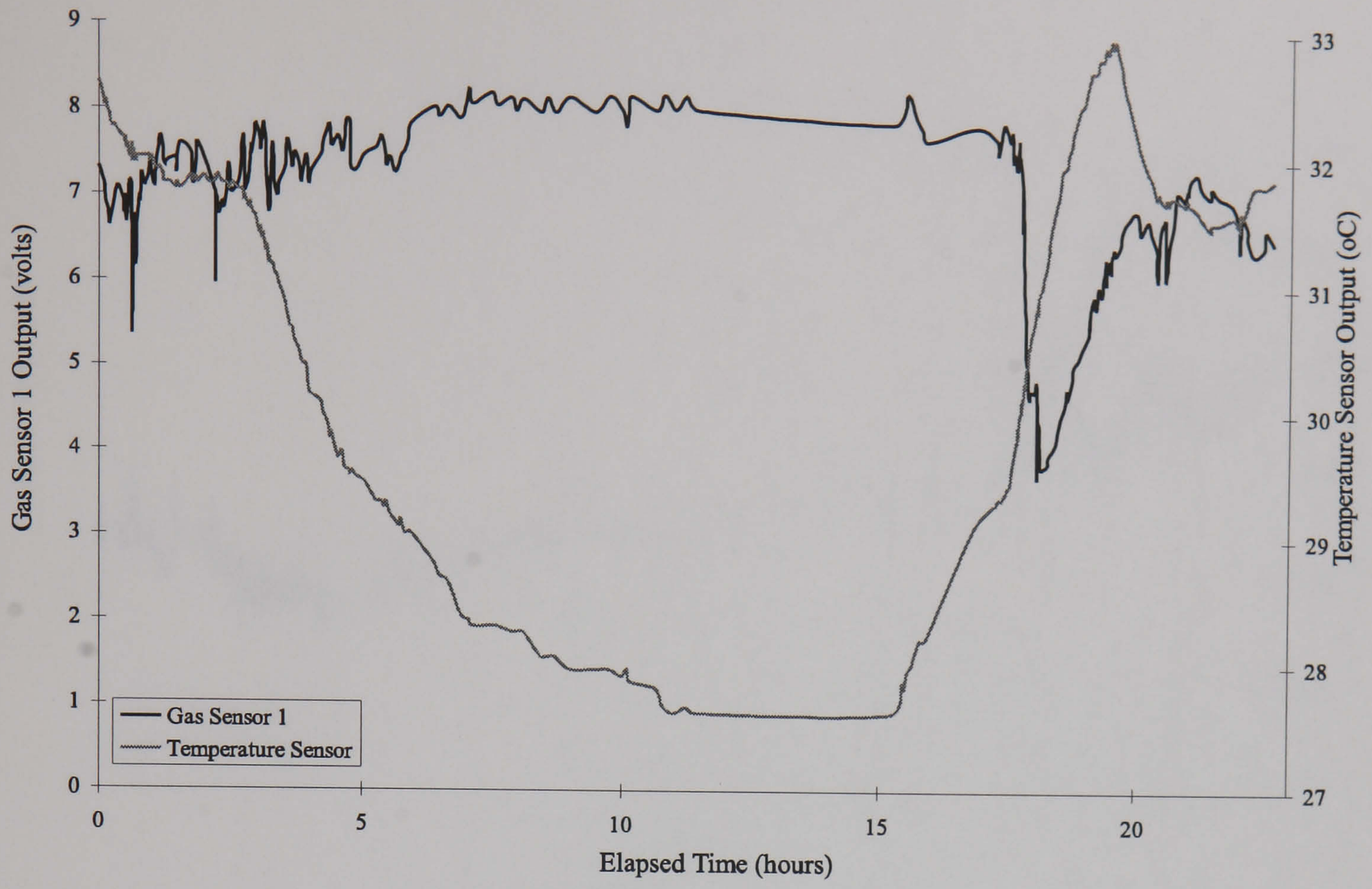


Figure 3.8: Plot showing the relationship between temperature sensor output (grey line) and gas sensor 1 output (black line). The output from the other gas sensor is not shown for clarity.

to decrease in value (i.e. an inverse relationship). It initially appeared from the plot in figure 3.9 that humidity was not a significant source of noise, however from the results of the correlation analysis it can be observed that it could have been as significant as temperature. There was negative correlation between temperature and humidity, the output of the humidity sensor was affected by ambient temperature [10]. Therefore, after further analysis it can be deduced that humidity, in this case, was not as significant a source of noise as temperature.

From the correlation analysis shown in table 3.3, in theory since there was no odour sample in the vessels (and therefore negligable odorants), each gas sensor should have had correlation value of 1.0 with the other gas sensors. It can be observed that the actual correlation values ranged from 0.88 to 0.99, this indicates that each gas sensor

[10] Since the LM35 series of temperature sensors are in sealed packages, they are unaffected by humidity. Condensation was not significant. The MiniCap humidity sensor was not in a sealed package because vapour must come into contact with the reactive element.

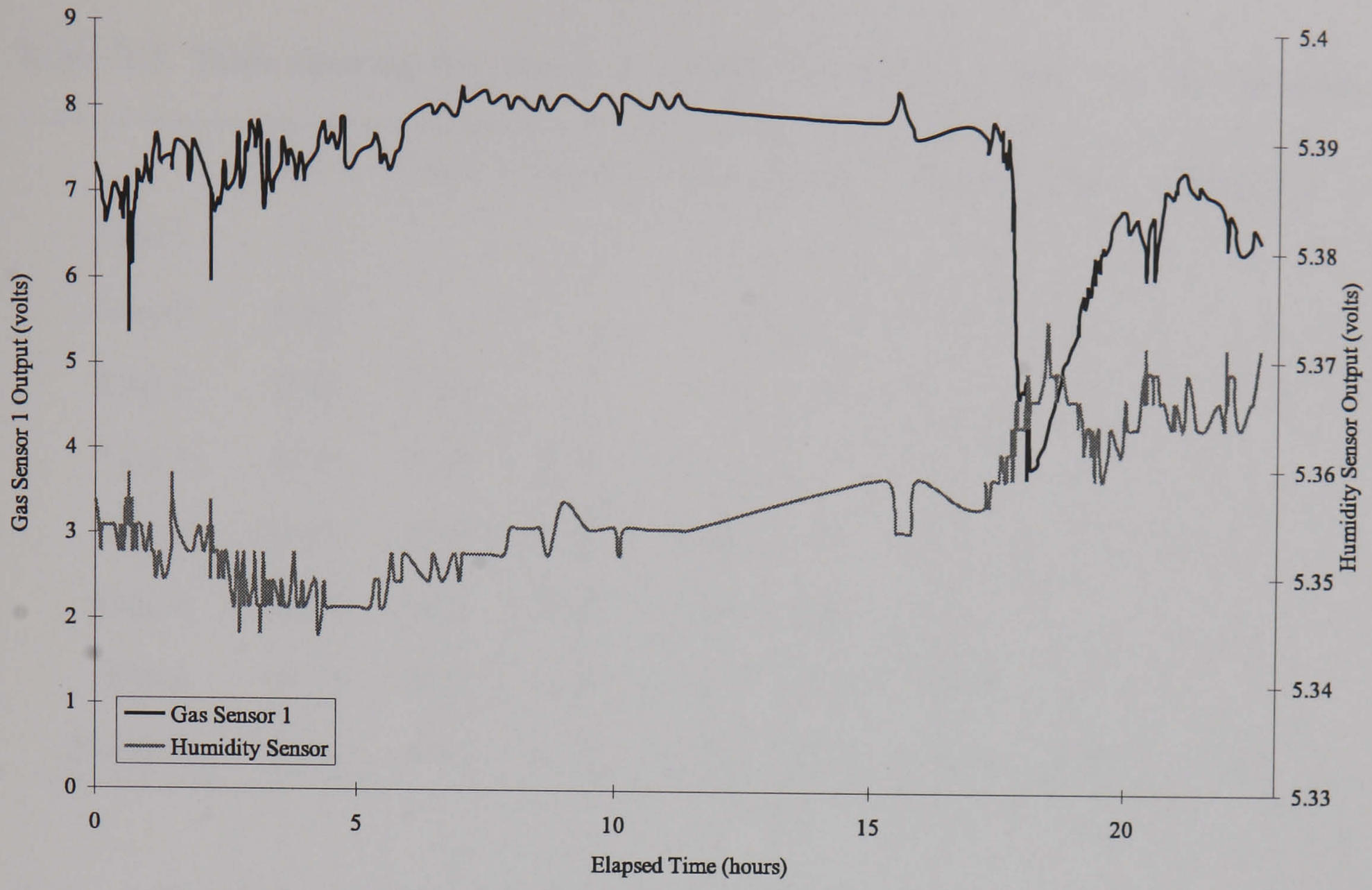


Figure 3.9: Plot showing the relationship between humidity sensor output (grey line) and gas sensor 1 output (black line). It can be seen that there is an inverse relationship between the output from the humidity and gas sensors, this confirms the effect of humidity upon the output of metal oxide gas sensors. Also, the humidity sensor may be responding to differently in the presence of different odours. The output from the other gas sensor is not shown for clarity.

reacted differently to changes in temperature and humidity. This makes temperature and humidity compensation more complex because the same compensation cannot be applied to all gas sensors. In the presence of an odour, we would expect lower values of correlation between gas sensors.

It can be observed from the plots in figures 3.8 and 3.9 that there was a period of relative temperature stability (from 12 to 15 hours, elapsed time), were the output from the temperature sensor was stable. During this period the output from the gas sensors was also relatively stable, showing that if ambient temperature was controlled there would be a reduction in the noise in the odour data.

Table 3.3: Table showing the results of correlation analysis on the 'burn-in' test data.

	Gas 1	Gas 2	Gas 3	Gas 4	Gas 5	Gas 5	Temp	Humidity
Gas 1	1							
Gas 2	0.94	1						
Gas 3	0.95	0.99	1					
Gas 4	0.91	0.99	0.98	1				
Gas 5	0.99	0.92	0.93	0.88	1			
Gas 6	0.94	0.99	0.98	0.98	0.91	1		
Temp	-0.43	-0.52	-0.46	-0.55	-0.40	-0.52	1	
Humidity	-0.65	-0.49	-0.54	-0.47	-0.70	-0.46	-0.26	1

3.7 Biological Experiment Procedures

In order to carry out the experiments using real bacteria cultures it was first necessary to address some issues that pertain to the handling and measurement of growth of bacteria cultures. This section outlines these procedures.

3.7.1 Methods for Culturing Bacteria Samples

In order to take measurements from bacteria samples, it was necessary to be able to grow bacteria colonies. Bacteria cannot be treated as inanimate objects, a particular bacteria culture would have been be useless if all the cells have died, or if the culture has become contaminated with another, unwanted, micro-organism.

The Biological Sciences Department keep in storage reference bacteria cells in order to be able to grow cultures containing known bacteria types. When the system was set up at the Biological Sciences Department, a set of cultures were started. For each bacteria type, two cultures were created (two cultures allowed a margin for problems such as contamination). The growth medium used was nutrient broth (NB), which is a common, multi-purpose, growth medium and which is able to allow the growth of bacteria found in humans. In a standard 25 ml glass jar, 20 ml NB was placed.

A small number of cells (contained in 0.1 ml of inoculum) from the reference storage were placed in the NB. These 'master' cultures were incubated at 36.8°C, after 12 to 24 hours of incubation the cultures were saturated with bacteria cells (i.e. Stationary Growth Phase), after 2 or 3 days the cultures entered the Death Growth Phase and finally after approximately two weeks, most cells in the culture were dead. At the start of each experiment it was necessary to have available a culture containing the desired bacteria type, in which the cells were in the Stationary Growth Phase. This ensured that a given volume of the culture would contain the maximum number of healthy cells (i.e. cells capable of growth). In order to make available such cultures, the day previous to an experiment, 2 sub-cultures would be made from one of the 2 current 'master' cultures by inoculating 2 lots 20 ml fresh NB (in 2 new glass jars) each with 0.1 ml from the one of the current 'master' cultures. These new sub-cultures become the 'master' cultures for subsequent experiments. Using this culturing technique a series of cultures for each bacteria type were created, at any time, cells in the 'master' culture were in the Stationary Growth Phase. The diagram in figure 3.10 shows this technique in diagramatical form.

After each subculturing, the new 'master' cultures were kept in a refrigerator in order to slow the metabolism of the bacteria cells and thus lengthen their useful life.

3.7.2 Viable Cell Counts

In order to accurately estimate the growth phase of the cultures being measured, it was decided to perform a viable cell count of all cultures from the beginning of each data gathering experiment and at intervals of one hour thereafter. The procedure for executing a viable cell count was as follows:

- Extract 0.1 ml from each culture using the syringe assembly in the vessel lid. Care has to be observed in order to prevent contamination. The 0.1 ml culture samples are placed into small, sterile, plastic vials for easy handling. All subsequent handling of culture samples was via these containers.

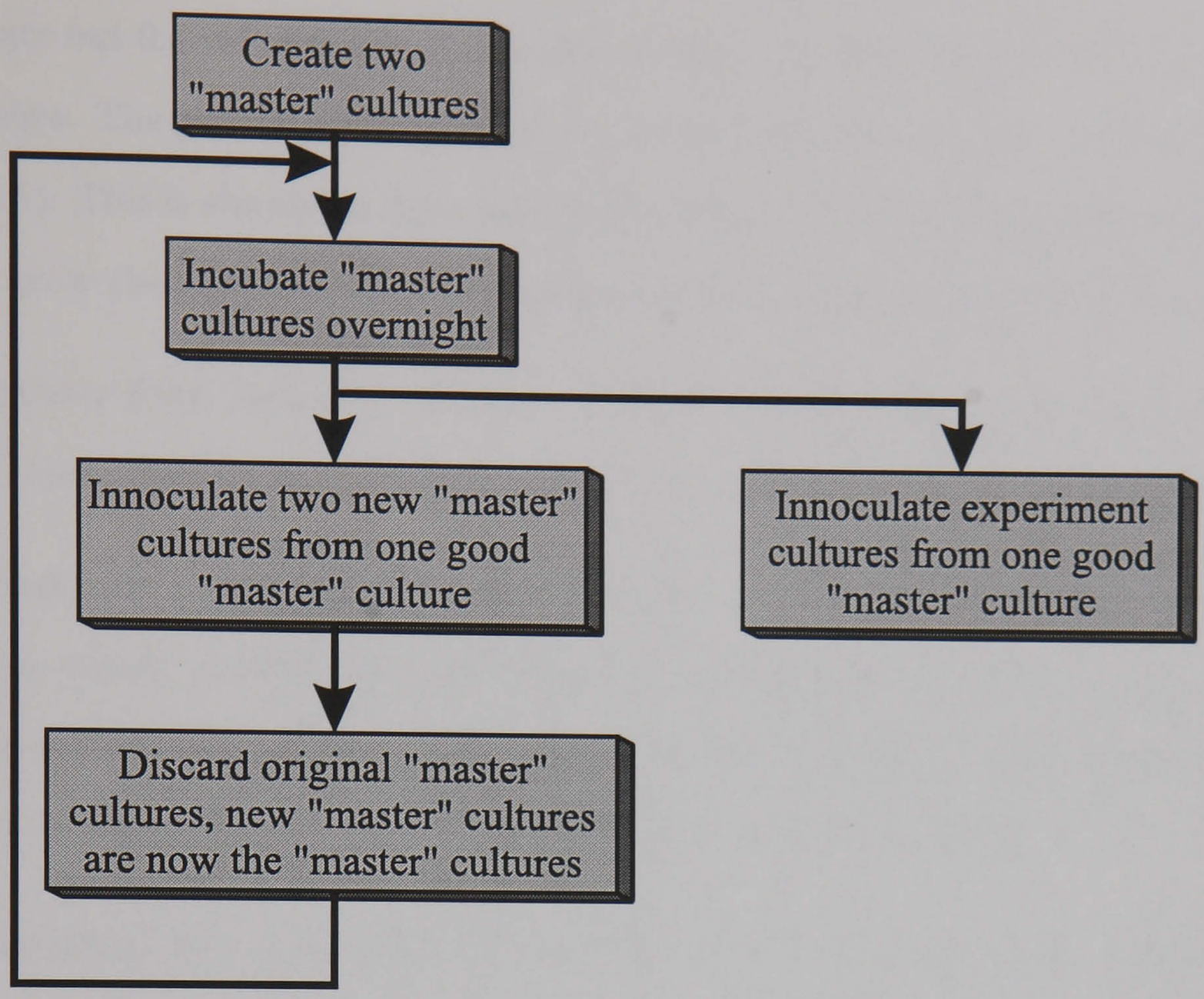


Figure 3.10: Flow chart showing how cultures were created and how a 'master' culture was maintained.

- Perform necessary serial dilutions. Because it is important that only a limited number of colonies develop on the plate, if overcrowding takes place then some colonies do not develop and cause errors in the count. Typically the number of colonies should be up to 300. A process called 'serial dilution' is used to ensure overcrowding does not take place by diluting small sample, which was inoculum extracted via syringe from a vessel, several times. More precisely, 0.1 ml of culture was diluted with 0.9 ml of phosphate buffer solution[11], 0.1 ml of this culture/buffer solution was further diluted with 0.9 ml of phosphate buffer solution, and so on; each dilution reducing the concentration of cells by a factor of ten. The precise number of times that this dilution was performed was determined by the experience of the laboratory technicians (typically from 6 to 10 times).

[11] Phosphate buffer solution is a common substance used to temporarily contain microorganisms undergoing analysis

- Plate out 0.1 ml of the final 4 dilutions onto 4 nutrient agar (NA) medium Petri dishes. The growth medium employed for the viable cell counts was Nutrient Agar (NA). This is simply an Agar Jelly form of the Nutrient Broth (NB) medium used to grow the aqueous cultures employed as odour samples in all experiments.

- Incubate Petri dishes for 24 hours at 36.8°C. It is possible to incubate the dishes for more than 24 hours but less than 24 hours risks colony counting errors.

- Count and record the number of colonies in each petri dish. In the 'standard plate count' method an assumption is made that each viable bacterium grows and divides to produce a colony, this colony is observable, after incubation, as a dot on the surface of a plate of NB agar (within a petri dish).

The counting method employed during for viable cell count was the 'standard plate count'. The viable counts obtained were an indication of the number of colony forming units[12] (cfu), from this the growth phase of the bacteria colony at a particular time was deduced from a growth plot (see figure 1.2 for a typical plot) of cfu against elapsed time. In practice the actual cfu quantity was derived from the number of colonies counted in the Petri dishes corresponding to the final 4 serial dilutions, an average of the 4 individual counts.

3.8 Tests At Biological Sciences Using Bacteria

The next set of tests were concerned with problems of contamination. The bacteria samples were growing in a medium known as nutrient broth (NB), which is a complex mixture of carbohydrates, sugars and minerals. The majority of microorganisms can grow, to some degree, in NB. However because nutrient broth was designed as an all purpose growth medium, it allows contaminants like fungi to grow rapidly and it is possible for a contaminant to become the dominant microorganism in the growth environment. Great attention was given to the design of the system to eliminate contaminants and therefore keep pure cultures pure, for example the extensive use of filters.

[12] Basically colony forming cells are cells that are capable of dividing into two daughter cells, i.e. are alive

It was then necessary to also give great attention to experimental procedure. The bacteria type used in these tests was *Escherichia coli*, the 'benchmark' bacteria used in micro-biology. The adjective 'benchmark' described the fact that this bacteria type is often used to characterise new laboratory techniques. It is easy to grow, it grows in a wide variety of media, it is well studied and when it has come into contact with humans is not particularly dangerous[13].

The autosampler was cleaned in the same manner, i.e. washed with ethanol/water mix, as that used in the 'burn-in test'. In the first experiment using bacteria samples, called 'bio-test 1', 25 ml of pure nutrient broth (NB) was placed in vessel #1 as the reference and 25 ml NB inoculated with 0.25 ml *Escherichia coli* 'master' culture was placed in vessels #2 and #3. The samples were placed into the vessel by injection where a syringe needle was incorporated into the modified vessel lids. This allowed a syringe (with the standard neck size) to be inserted into the mouth of the needle (the needle has a plastic moulding at the top to facilitate this) and the contents of the syringe can be injected into the vessel. This mechanism also allowed for matter to be extracted from the vessel. Figure 3.11 shows this modification. It was necessary to put bacteria in vessels 2 and 3 so that it could be tested whether any cross-contamination occurred between channels.

The system was checked for air leaks, and none were found to be present. The eight minute cycle was used (as described previously). This test was run for 153 cycles and therefore lasted 20 hours and 24 minutes. After this time the samples within the vessels were visually examined.

Before the results of the examination are detailed it is necessary to describe the examination procedure. Pure NB, in solution, is a clear, amber liquid. NB that has microorganisms growing in it is cloudy. This property is often used in micro-biology to count the number of microorganism cells in a given volume of solution, using a method called the Optical Density (OD) measurement. In these experiments the OD measurement was not employed because all that was required was to know if a particular

[13] Recently *Escherichia coli* 157 has been brought to the attention by the media after some food-poisoning outbreaks, this strain of the bacteria is extremely rare.

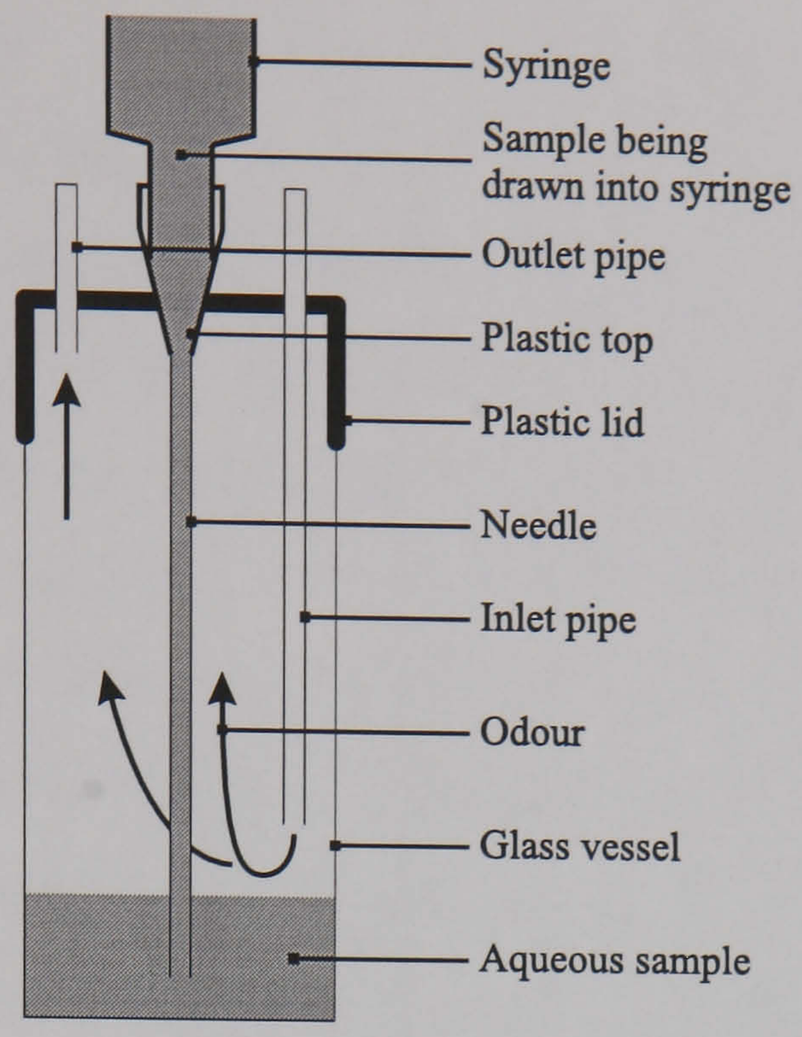


Figure 3.11: Diagram illustrating the modification of the vessel lids within the autosampler by introducing a syringe.

NB solution contained any micro-organisms at all, and after an incubation period of 20 hours or more, if the original solution contained only one or two thousand cells then the solution would appear significantly cloudy (easily observable by the human eye).

All vessels were found to be cloudy and therefore contained microorganisms. However, only pure NB was originally introduced to vessel #1 and consequently this vessel had become contaminated. Since the system was free of air leaks the contamination was either a result of the system not being free of microorganisms at the start of the test or that microorganisms had entered the system via the input and were able to get past the filters.

Following the problem of contamination encountered in experiment 'bio-test 1' a second experiment, 'bio-test 2', was performed where the procedure followed was identical to that used for 'bio-test 1' (including the initial contents of the vessels) except that the valve/vessel assembly was not attached to the nose and all the channels were switched off. In other words no air was passed through the autosampler. The test was run for 18 hours, after this time the samples were examined. Vessel 1 was found to be contaminated, therefore contaminating microorganisms were present in the vessels

before the test was started.

Following the results of experiment 'bio-test 2' it was decided to clean the valve/vessel assembly with 5% sodium hyperchlorite solution. This substance is a bleaching agent and very powerful anti-microbial agent. Therefore experiment 'bio-test 3' was performed where the procedures were identical to those employed in 'bio-test 1' but an extra cleaning cycle was introduced. Before alcohol/water mix was pumped through, the sodium hyperchlorite solution was pumped through. It was not necessary to leave the solution in the assembly because micro-organisms are killed very rapidly (within seconds). Afterwards, when the alcohol/water mix was pumped through the assembly it also removed any traces of the sodium hyperchlorite and therefore also acted as a flushing process. The vessels were examined after the experiment had finished and again vessel #1 was contaminated.

It was decided that a possible additional cause of contamination was the large mesh size of the bio-filter. Therefore a different filter was used with a mesh size of 0.2 μm, a size commonly considered to be small enough to stop all common airborne contaminants. This new filter was autoclaved so that it was sterile. Experiment 'bio-test 4' was started using the new bio-filter, the procedure being identical to that used in experiment 'bio-test 3'. The vessels were examined after the experiment had finished and again vessel #1 was contaminated.

Up to this point samples were injected into each vessel via a syringe in the lid of each vessel, it was thought that the mouths of the needles were a possible site of contamination. Therefore, in the next experiment to be performed ('bio-test 5'), the procedure of putting the samples into the vessels was changed. After the system has been cleaned and re-assembled instead of injecting the samples into the vessels, the vessels were removed from the autosampler, the samples poured into the vessels using a pipette and then the vessels re-fitted. Experiment 'bio-Test 5' was carried using this modified procedure, and the vessels examined at the end. The results were that no contamination took place, i.e. vessel #1 remained clear and hence free from contamination.

In order to test the experimental procedure further more experiments were per-

formed ('bio-tests 6,7,8, 9 and 10'), it was therefore possible to establish whether contamination free odour samples were possible. The reason so many tests were performed was that there were equipment failures during experiments 'bio-tests 7,8 and 9'. During 'bio-test 7' one of the solenoid valves became faulty stopping gas flow through that channel. During 'bio-test 8' the rotary valve pump failed and was repaired, the experiment had to be aborted. During 'bio-test 9' the pump failed again and was replace with a new identical unit. 'Bio-test 10' was run in order to test the whole system with the new components (i.e. vacuum pump) with the experimental procedure developed in earlier experiments. The samples at the end of experiment 'bio-test 10' were contamination free.

The procedure developed, assuming that an experiment had previously been carried out using the same equipment, is described in the following list, the steps were executed in the order they appear in the list:

1. Make sure software exited normally after previous run, i.e. channel 1 is selected. It was important that a channel should be open in order that gas flow should continue to the FOX 2000. If gas flow were to be interrupted then this would have caused the sensors to heat up (changing their characteristics) and would have stopped gas flow through the pump causing the electric motor to overheat.

2. Detach the autosampler from the FOX 2000. Since chemicals were pumped through the pipe-work and vessels in the autosampler, in order to prevent those chemicals from coming into contact with the sensor array in the FOX 2000, it was necessary to detach these two sub-systems.

3. Remove the bio-filter, bubbler and charcoal filter. Place the bio-filter in ethanol/water mix. It is vital that the bio-filter should remain sterile. Placing the filter in ethanol/water mix prevented any microorganisms from gaining access.

4. Remove vessels from the autosampler and fit new, autoclaved (therefore sterile), vessels. The vessels themselves are expendable and since the vessels contain the samples from the previous experiment and therefore probably contain microorganisms the easiest option was to replace them completely.

5. Attach a pump to the autosampler output and a long pipe to the input. Attach a second long pipe to the output from the pump and place the other end of this pipe into a large vessel (1 litre capacity). Also halt the autosampler Control Program and start the LPM-16 output program. Activate all channels.

6. Place the input pipe of the autosampler into a container containing 5% sodium hyperchlorite solution. Pump the sodium hyperchlorite solution through the autosampler for 2 minutes, ensuring that the solution has flowed through all pipes, valves and vessels. Also flush syringe needles in lids with solution. This solution can also leave small crystals if the water is allowed to evaporate, therefore the assembly should be flushed immediately afterwards with ethanol/water mix.

7. Remove the input pipe from the vessel containing the sodium hyperchlorite solution and leave free. Invert the autosampler so that the vessels are upside down. Switch on the pump and pump out all the remaining sodium hyperchlorite solution.

8. Place the input pipe of the autosampler into a container with 75% ethanol/water mix. Turn the autosampler the right way up. Switch on the pump and pump the ethanol/water mix through the autosampler for 5 minutes, ensuring the ethanol/water mix has flowed through all pipes, valves and vessels. Also flush syringe needles in lids with ethanol/water mix.

9. Leave the remaining ethanol/water mix in the valve/vessel assembly for at least eight hours. During this time de-activate all channels to prevent the solution from leaking.

10. Invert the valve/vessel assembly, switch on the pump and pump out remaining ethanol/water mix from the autosampler. Make sure all possible ethanol/water mix has been pumped out of the system. Turn the valve/vessel assembly the right way up.

11. Purge the valve/vessel assembly with air to remove any residual vapours this is done by switching on all channels and pump air through the system for at least

2 hours.

12. Switch off all channels. Remove vessels from the system and fill with samples that are to be analysed. Re-fit vessels (now containing the samples) to the autosampler. Also PTFE tape is wound around the neck of each vessel to ensure that the seal with the lid is air-tight.

13. Re-fit bio-filter, charcoal filter, and bubbler to the rest of the system, also connecting up any pipes that aren't yet connected except the pipe connecting the FOX 2000 to the rest of the system.

14. Activate channel #1 and reconnect the FOX 2000 to the autosampler. At this point a final check of the system is carried out, checking for things like bad pipe connections and that the sensor output voltages are within working levels of the LPM-16 I/O card.

15. Halt the LPM-16 Output Program and restart the autosampler Control Program. Input experimental parameters into the software and start the experiment. The operator should stay with the system for the first few cycles of the experiment in order to ensure that everything is running satisfactorily. Adjust any controls as necessary.

Once a stable and reliable experimental procedure had been developed further experiments could then be carried out.

After consultation with the laboratory technicians in the Biological Sciences department, it was decided to run the data gathering experiments for a duration of 12 hours. This time span is long enough to allow cells in the cultures of all the bacteria types to enter the Stationary Phase. It was decided that it would be impractical to try to take measurements from cultures in the death phase because the length of time required for this would probably be in the order of 2 to 3 days. Because a data gathering experiment requires the worker to be present in order to perform the viable cell counts, an experiment duration of 2 to 3 days would have been impractical.

Therefore for vessels 2 and 3, 13 viable cell counts were performed each. For vessel

1 one viable cell count was performed after 12 hours in order to determine if there were any contaminants present. A total of 27 counts per experiment.

3.9 Experiments Performed Using Bacteria Cultures

Once the apparatus was tested and experimental procedure was developed a series of tests were performed on 2 bacteria types; *Escherichia coli* and *Staphylococcus aureus.* The main reason that these two types were selected were that they are both ENT pathogens, they are readily available 'benchmark' bacteria types and that they are very different organisms[14] and should therefore produce easily distinguishable odours. It was felt at this time to limit the study to 2 bacteria types only in order to facilitate a more thorough analysis.

In order to interpret the growth phases of the aqueous cultures, indicated by plots of colony forming units (cfu); the academic staff and laboratory technicians at the Biological Sciences Departments were consulted as to where the boundaries between growth phases occurred. The boundaries are not clear cut and are only an indication of progress of growth.

3.9.1 Experiments Performed on

In total, 3 experiments were performed using *Escherichia coli* as the sample bacteria. Originally two experiments were planned but one experiment had to be aborted when the vacuum pump in the FOX 2000 failed. At this time that it was decided to replace the rotary vane type vacuum pump with a more reliable diaphragm pump type, see section 3.4 for more details. Subsequent experiments proved the diaphragm pump to be reliable. Therefore the experiment where the pump failed will be discounted from further analysis. The valid experiments were denoted as 'experiment 1' and 'experiment 2' respectively. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters.

[14] *Escherichia coli* is gram -ve and *Staphylococcus aureus* is gram +ve. The gram stain is the most common differentiation test employed in micro-biology

All experiments were identical and were set-up using the procedures outlined in the previous sections, the odour sources were aqueous cultures of 25 ml NB in vessel 1 and 25 ml NB inoculated with 0.25 ml *Escherichia coli* 'master' culture in vessels #2 and #3.

Table B.1 and table B.2 show the results of viable cell counts that were performed. It can be noted that there are small differences between count for colonies at the same age. Bacteria samples have chaotic components to their behaviour, no two samples which given near identical conditions will behave in an identical manner; therefore a small change in initial conditions can lead to a large change after a long period of time (in this case 12 hours). This can be observed as discrepancies in the colony forming unit (cfu) count between bacteria samples that have grown under near identical conditions [15].

Figures 3.12 and 3.14 are plots of the number of colony forming units in 1 ml of culture sample, the data source were the viable cell counts given in table B.1 and table B.2 respectively. The phases of growth in figures 3.12 and 3.14 are indicated by the shaded areas (see figure captions for colour key). It can be observed that, according to the plot, the bacteria samples used in the second experiment had virtually no lag phase at all, in reality there was probably a short lag phase which did not show on the plot because of the count frequency of one every hour was not sufficient in this case.

Figure 3.13 is based on the plots in Figure 3.12, but with two extra plots. For each vessel (i.e. 2 and 3), the magnitude of the voltage change in the signal averaged over all 6 sensors, is shown. The magnitude of the voltage change is the change in output voltage that occurs when a given gas sensor is exposed to an odour. This measure is later referred to as the difference feature model in chapter 4. The purpose of Figure 3.13 is to illustrate to the reader an example of the system response during an experiment. It can be seen that that the trend for the additional plots is to increase with time (and cfu count). There are many measures that could be plotted (see chapter 4), either averaged or not over all 6 sensors. It is therefore not practical to show all possible plots.

[15] Identical cfu may be achieved if the initial conditions were the same, however this criteria is impractical to satisfy

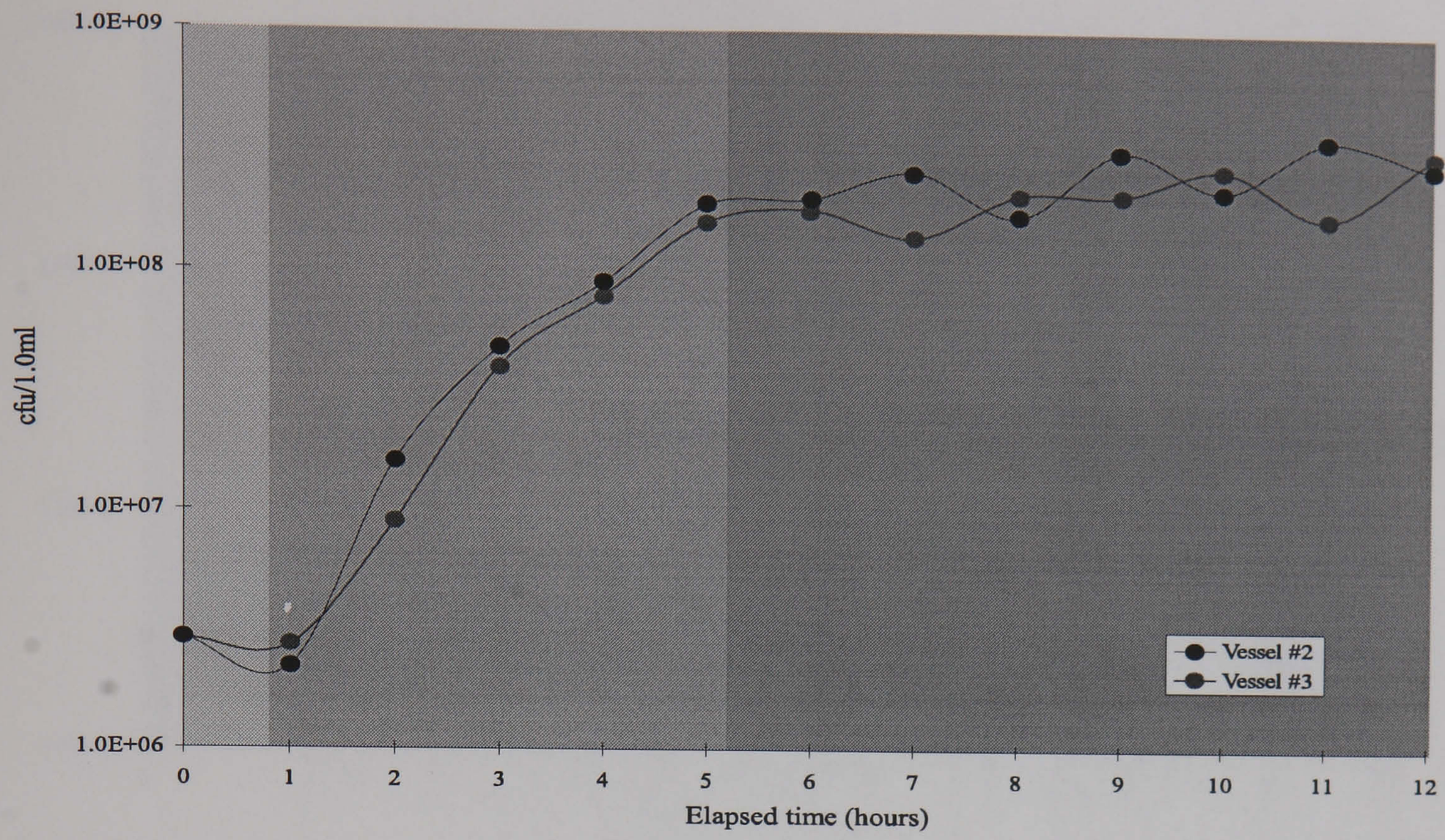


Figure 3.12: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Escherichia coli* experiment 1, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

3.9.2 Experiments Performed on

There were 2 experiments carried out using *Staphylococcus aureus*, the procedures that was used was identical to that employed for experiments 1 and 2, the only difference being that the odour source in vessels #2 and #3 contained aqueous cultures of 25 ml NB inoculated with 0.5 ml *Staphylococcus aureus* 'master' culture. 0.5 ml of 'master' culture was used instead of 0.25 ml (as used for *Escherichia coli* because *Staphylococcus aureus* grows slower in NB than *Escherichia coli.* These experiments were denoted as 'experiment 3' and 'experiment 4' respectively. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters.

Table B.3 and table B.4 show the results of viable cell counts that were performed. Figures 3.15 and 3.16 are plots of the number of colony forming units in 1 ml of culture sample. It can be noted that, in general, the cfu counts for *Staphylococcus aureus* are lower than those for *Escherichia coli* even though more 'master' culture was used to

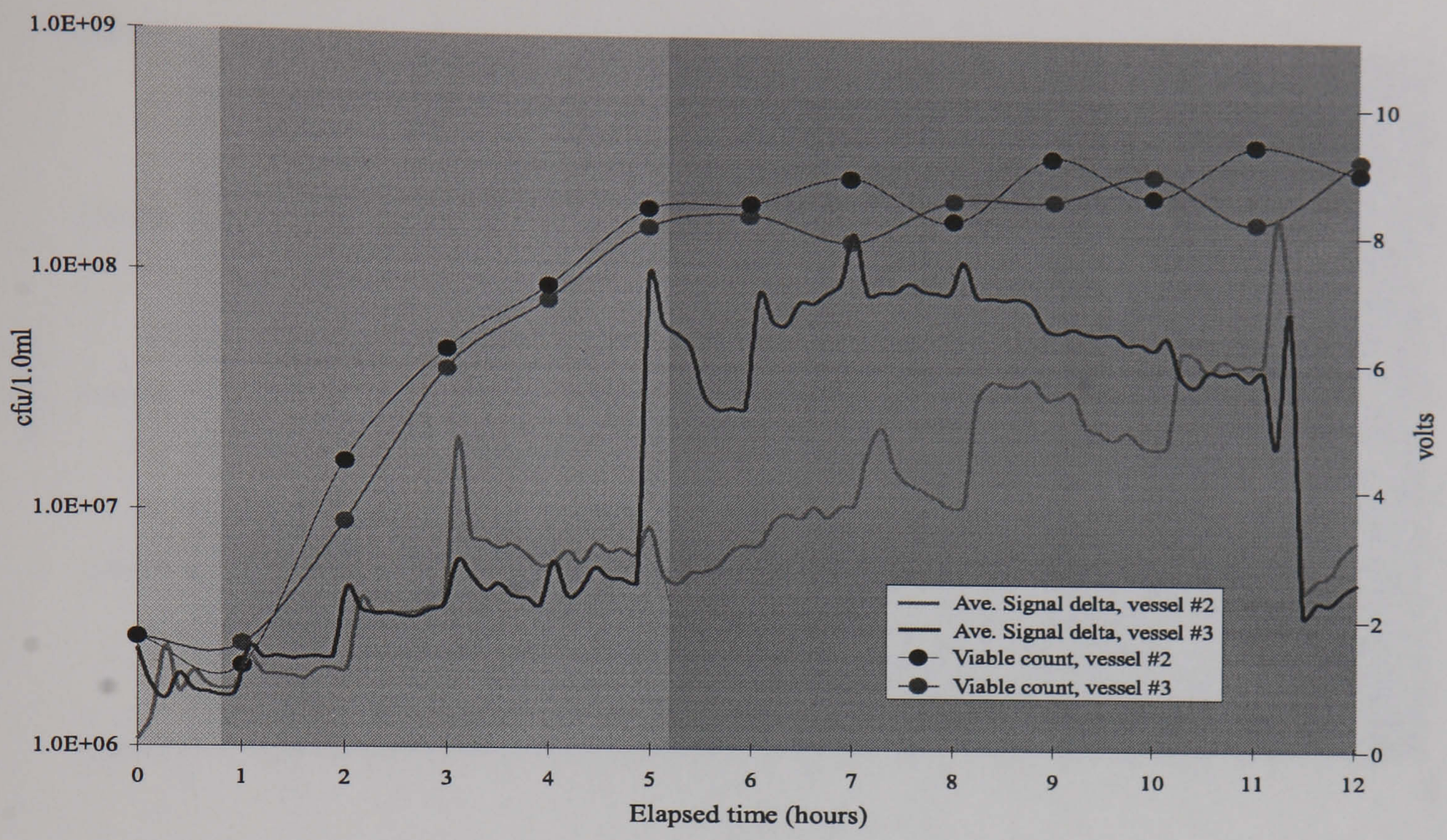


Figure 3.13: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Escherichia coli* experiment 1, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase. Also the magnitude of signal change averaged over all 6 gas sensors for vessels 2 and 3 is shown in order to give an indication of system response during the experiment (See difference feature model in chapter 4 for more details).

inoculate the sample cultures. This confirms that ***Staphylococcus aureus*** is less suited to a NB growth medium. Also ***Staphylococcus aureus*** is nonmotile (which means that it does move under its own propulsion) unlike ***Escherichia coli*** and therefore does not disperse as quickly, so new cells tend to grow in areas where nutrients are already depleted (i.e. in other words it is not suited to growth in an aqueous solution).

3.10 Summary

The existing apparatus was improved in order to meet the needs of this more demanding application. The need for improvement was highlighted in a feasibility study performed prior to this research. The major improvement was automated delivery of odours by means of solenoid operated valves, increased speed of odour delivery to the gas

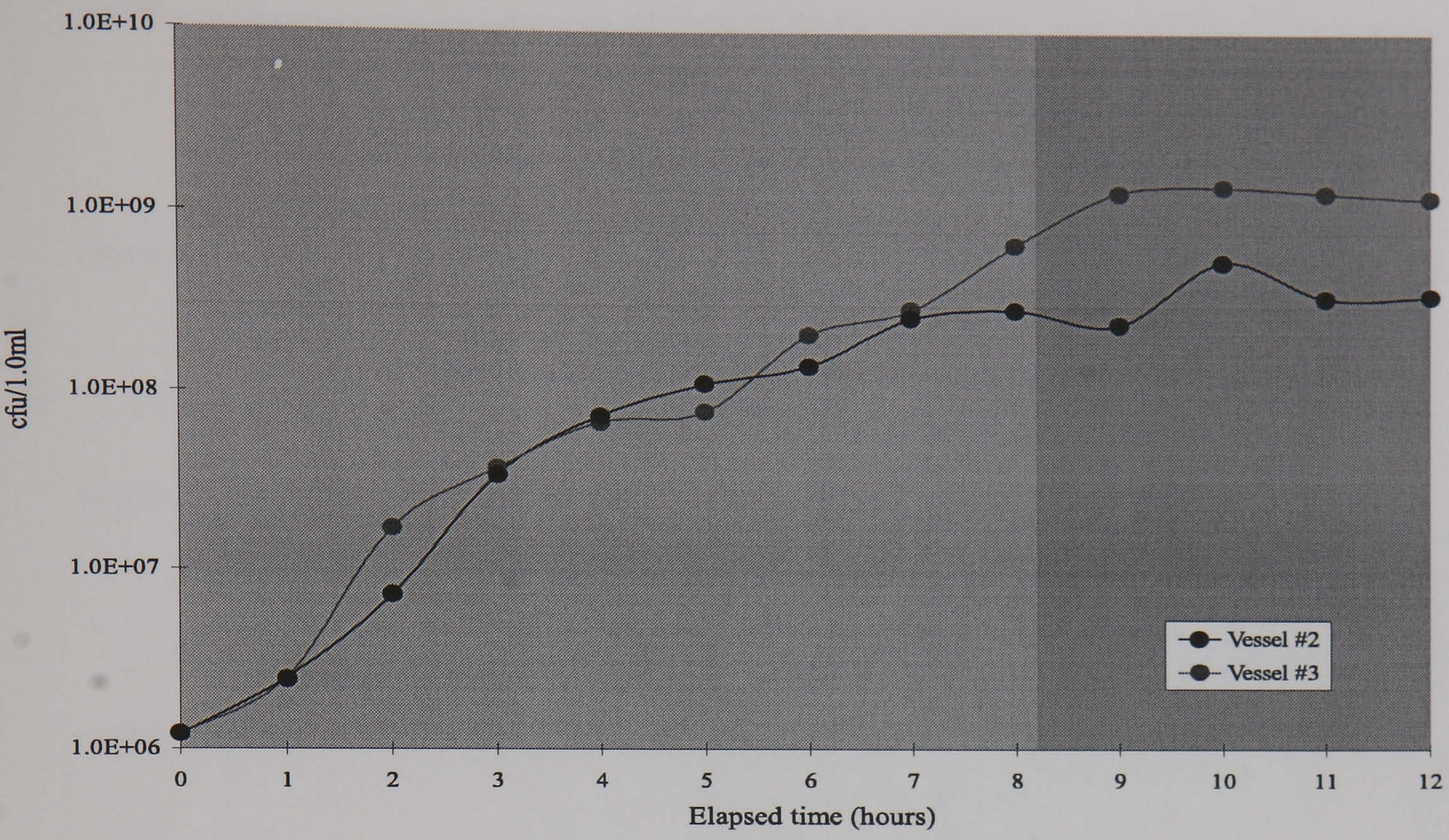


Figure 3.14: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Escherichia coli* experiment 2, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

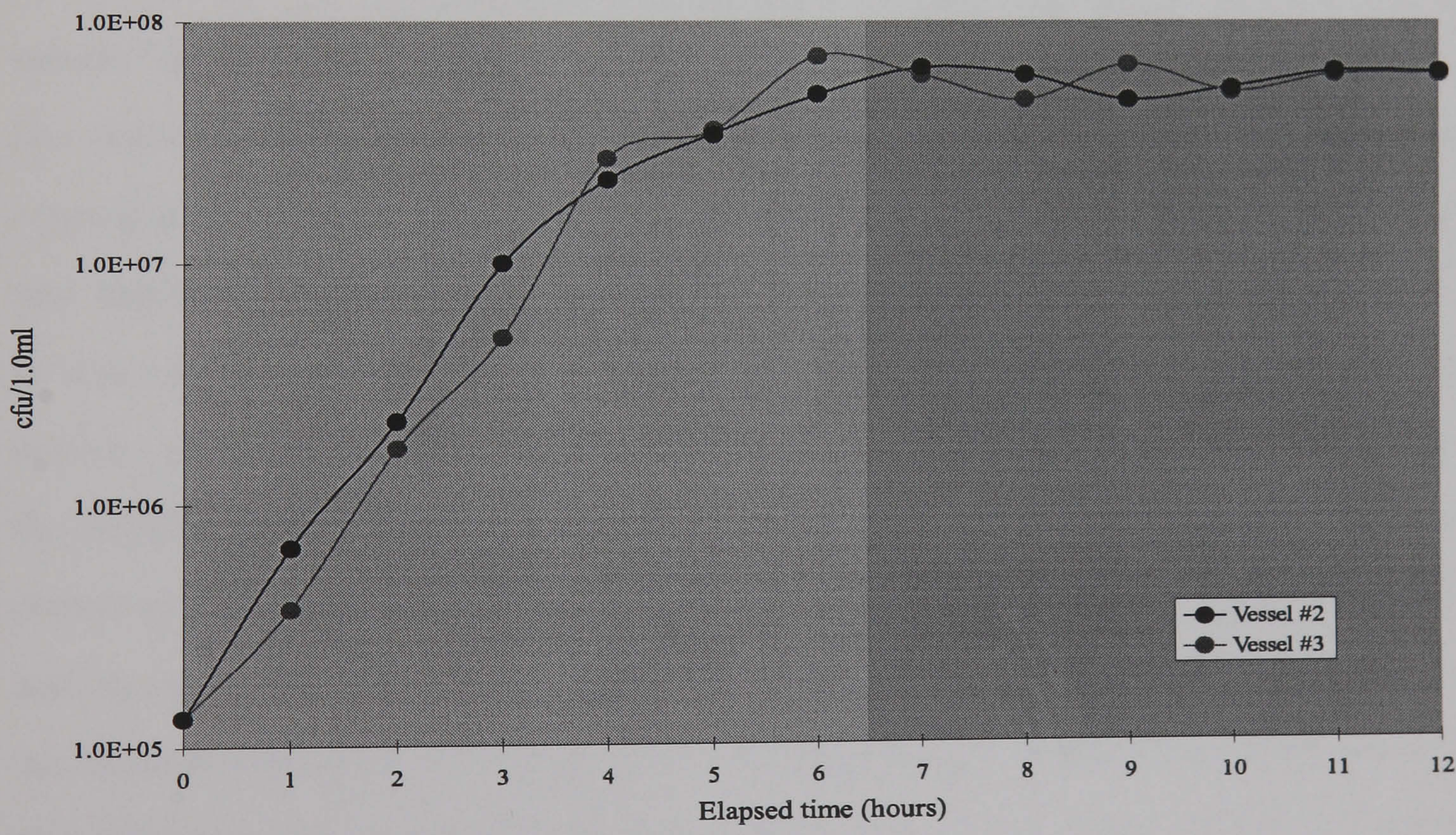


Figure 3.15: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Staphylococcus aureus* experiment 3, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

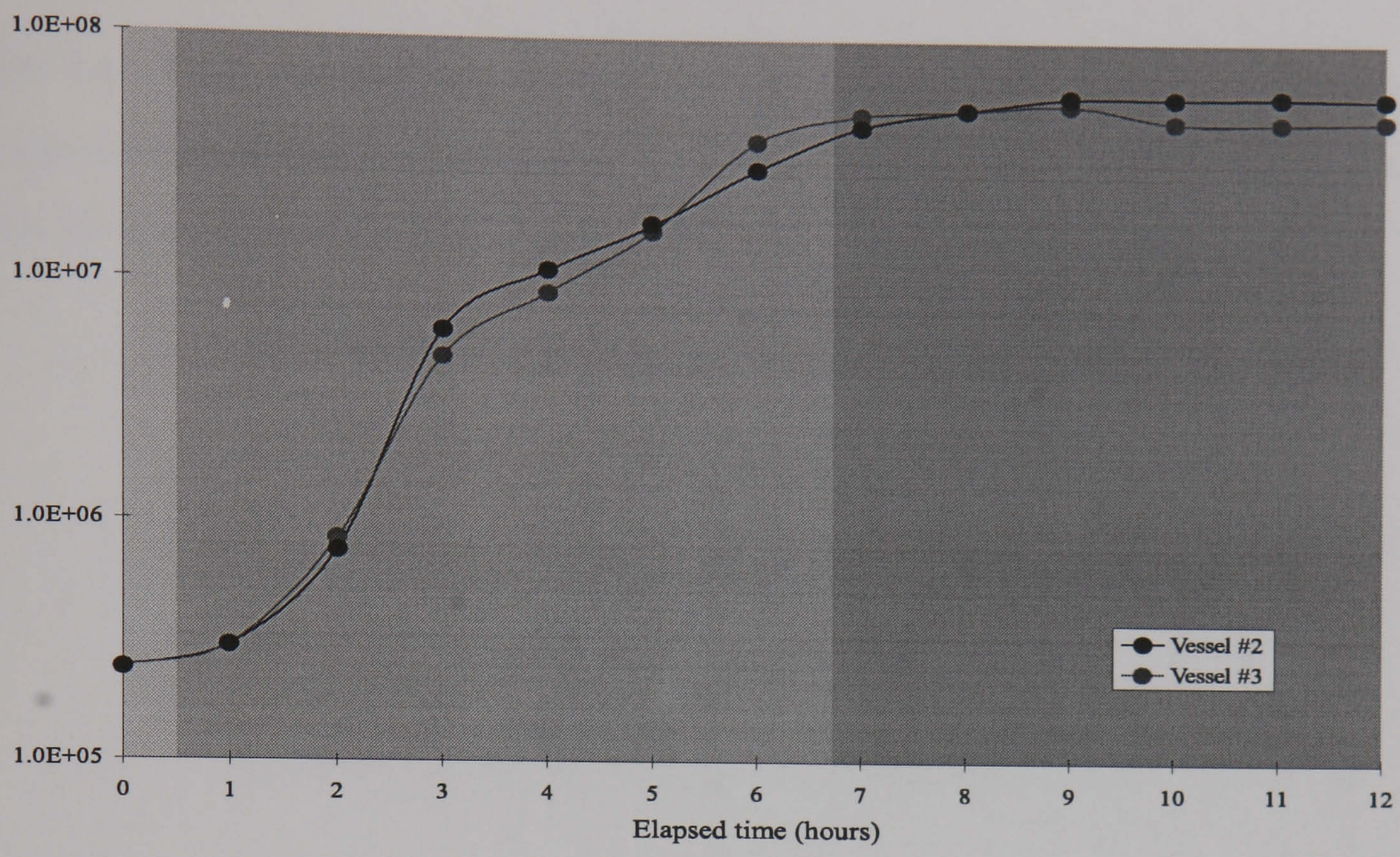


Figure 3.16: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for ***Staphylococcus aureus*** experiment 4, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

sensors (in the sensor chamber) and better filtering of the air input to the system. Gas delivery was controlled using computer software which allowed easy tailoring of experimental parameters, such as the duration of time in which a particular sample was 'smelt'. This meant data gathering experiments could be automated, however, in this case the experiments were not totally automated because the samples had to undergo simultaneous independent testing (viable cell counts). The vessels containing the samples were modified so that small amounts of bacteria culture material could be extracted without interfering with the 'smelling' process. The electronic nose was tested and characterised. The effect of fluctuations of ambient temperature and humidity upon the baseline resistance of the gas sensors was investigated. Experimental methods for data collection were developed, this included methods to eliminate infection of samples by unwanted micro-organisms and the performance of viable cell counts. Finally a set of data gathering experiments were performed over a period of four weeks on two types of bacteria: ***Escherichia coli*** and ***Staphylococcus aureus***, two experiments were performed

Table 3.4: Table summarising the data gathering experiments performed using the temperature controlled sensor chamber. Growth curve quality is an indication to how much the actual growth curve conformed to the ideal curve(see figure 1.2.

Experiment no.	Bacteria types	Growth curve quality
1	*Escherichia coli*	good
2	*Escherichia coli*	good
3	*Staphylococcus aureus*	medium
4	*Staphylococcus aureus*	medium

for each bacteria type. Since viable cell counts were performed during the experiments, the size of the populations could be calculated and the growth curves for each culture plotted. This external control added reliability to the data. Table 3.4 summarises the experiments performed.

Chapter 4

Initial Data Exploration Using Pre-Processing and Classification Techniques

This chapter describes the initial data analysis that was performed on the data-sets collected in the experiments detailed in chapter 3. The purpose of this data analysis was to gain a better understanding of the nature of the data from the electronic nose. Data analysis consisted of pre-processing followed by classification.

Data pre-processing, as employed in this application, was the application of one or more signal processing algorithm(s), such as feature extraction and normalisation, to an 'input' data-set (in this case a 'raw' data-set which were voltages recorded from the sensor array). The resultant 'output' data-set (or Feature-Set) was then applied to subsequent classifiers.

Pre-processing was carried out in order to improve the classification process. The importance of pre-processing has been well documented both in pattern recognition [55] and in the area of electronic noses [50]. The ultimate test of pre-processing methods is to study their effect on classifier performance.

The application of numerous classifiers are described later in this chapter. Two classification tasks were performed; classification of bacteria culture growth phase and

classification of bacteria type. The classification was performed by means of ANNs. Classical statistical methods were employed as benchmarks by which the performance of ANNs were compared.

The boundary between pre-processing and classification is not easy to define because there are processes that can be performed both as an integral part of the classifier or as a separate stage. An example of such a process is the linear scaling of input variables[1]. For the purpose of documentation, the boundary between pre-processing and classification will be a practical one, i.e. those techniques that were applied to the 'raw' data-sets which were not implemented as part of the classification software (since the classification was implemented as computer software). In reality the pre-processing techniques were applied by means of a separate suite of programs other than those used for classification.

4.1 Pre-Processing Techniques

Before pre-processing techniques can be applied, an understanding of the processes involved is desirable in order to predict the effect on classification performance. As employed in this research project, pre-processing can arguably be called 'Feature Extraction' because the input to the pre-processor is the 'raw' data-set and the output is a Feature-Set. In general, the importance of pre-processing has been overlooked in previous work where ANNs have been employed as classifiers (see chapter 2). Research into pre-processing techniques, in this project, was given equal importance as the classification techniques. Pre-processing the 'raw' data-sets from the gas sensors included up to two different types of processing function: feature extraction and normalisation. The following sections discuss these functions. The signals from the non-gas sensors were not pre-processed in the same manner as the signals from the gas sensors; this is discussed in a separate section 4.1.3.

[1]The weights in an artificial neural network can (and do) scale input vectors

4.1.1 Feature Extraction

Essentially Feature extraction is a process whereby the relevant information content of a data-set is preserved as much as possible whilst, at the same time, decreasing the amount of data. Relevant information is information that, when present, improves classification performance. Information that is not relevant can be considered as noise. Therefore if a ratio is visualised of the amount of relevant information contained in a data-set over the total data-set size; feature extraction increases this ratio. The ideal value for this ratio would be 1, but in reality this can never be achieved because a finite amount of relevant information is always lost during feature extraction. An ideal feature extraction technique would be one where relevant information is perfectly preserved, however, this cannot be practically achieved for complex tasks, such as classification of odours.

The question could be asked: Why not input the 'raw' data directly into a classifier? The answer is: It would be impractical to input the vectors in a 'raw' data-set into a classifier because the vectors represent a time series. It initially seems possible that the entire time series for a single measurement phase (or 'smell') could be simultaneously input to a classifier. However, considering only the gas sensors, if a 'smell' contained 120 vectors (for a 4 minute 'smell'), the classifier would need 720 inputs (for 6 gas sensors), which is a very large number. It has been shown that, in general, classifiers that are too complex suffer from poor performance [55]; this has also been shown previously on electronic nose data [48]. This shows that dimensionality reduction of the input is a vital property of feature extraction.

The 'raw' data-sets were processed on a 'smell' by 'smell' basis so that any classifier only had to cope with one 'smell' at any one time. The next step was to extract a feature vector, $\mathbf{X_j} = (x_{1j}, x_{2j}, \ldots, x_{ij})$, from each 'smell'. Each feature vector component corresponds to a single feature, x_{ij}, from a single gas sensor. Because the valve's status was recorded along with the sensor signals, identifying individual 'smells' within the 'raw' data-set was straightforward. This was defined by 'smell' boundaries, which were the points in time where the valves for channels 2 or 3 were switched off and the valve for channel 1 was switched on, see figure 4.1.

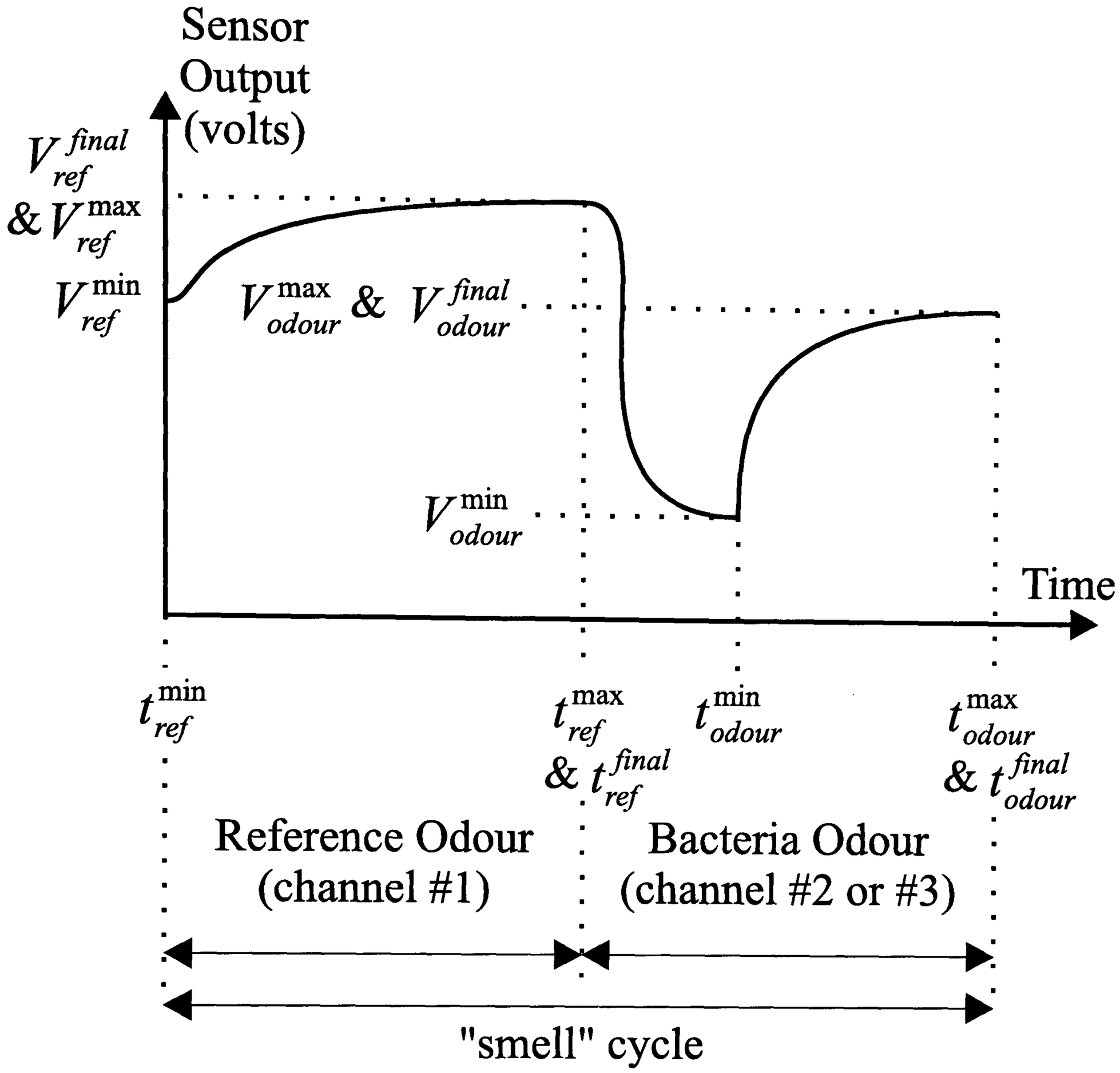


Figure 4.1: Plot of of typical gas sensor response for a single gas sensor during a single measurement phase (a 'smell'). The parameters used in gas sensor feature models are indicated. Note also that because in this example the maximum level is reached at the end of each half-cycle, the maximum and final voltages and times coincide, this does necessarily have to be the case.

In figure 4.1 the parameters V^{final}_{ref}, V^{max}_{ref}, V^{final}_{odour}, V^{max}_{odour}, V^{min}_{ref}, V^{min}_{odour}, t^{min}_{ref}, t^{max}_{ref}, t^{final}_{ref}, t^{min}_{odour}, t^{max}_{odour} and t^{final}_{odour} are indicated. These are the parameters input into gas sensor feature models. Many gas sensor feature models have been tried in previous work [49, 53, 50]. Each different model has associated with it a two letter notation (all notation is shown in italics) in order to ease handling of different feature-sets. Below

are shown the most promising gas sensor feature models:

Difference model (*df*), this is the difference between the maximum output voltage for the reference odour and the minimum output voltage for the bacteria odour:

$$x = V_{ref}^{max} - V_{odour}^{min} \quad (\propto \Delta R) \tag{4.1}$$

Relative model (*rl*), this is the ratio of minimum output voltage for the bacteria odour and the maximum output voltage for the reference odour:

$$x = \frac{V_{odour}^{min}}{V_{ref}^{max}} \quad (= \frac{R}{R}) \tag{4.2}$$

Fractional difference model (*fd*), this is the ratio of the difference model and the maximum output voltage for the reference odour:

$$x = \frac{V_{ref}^{max} - V_{odour}^{min}}{V_{ref}^{max}} \quad (= \frac{\Delta R}{R}) \tag{4.3}$$

Further to these gas sensor feature models a new set of feature models was employed. These models which also use V_{ref}^{final} and V_{odour}^{final} parameters are shown below:

Absolute final output model (*af*); two features, x_1 and x_2, are output, which are the final output voltage for the bacteria and reference odours respectively:

$$x1 = V_{odour}^{final} \; and \; x2 = V_{ref}^{final} \tag{4.4}$$

Minimum output model (*mn*);two features, x_1 and x_2, are output, which are the minimum output voltages for the bacteria and reference odours respectively:

$$x_1 = V_{odour}^{min} \; and \; x_2 = V_{ref}^{min} \tag{4.5}$$

Final relative model (*fr*), this the the ratio of final output voltages for the bacteria odour and the reference odour:

$$x = \frac{V_{odour}^{final}}{V_{ref}^{final}} \tag{4.6}$$

Modified difference model (*md*), this is the difference between the voltage change for the bacteria odour and the voltage change for the reference odour:

$$x = (V_{odour}^{max} - V_{odour}^{min}) - (V_{ref}^{max} - V_{ref}^{min}) \tag{4.7}$$

Modified fractional difference model (*mf*), this is the ratio of the voltage change for the bacteria odour and the voltage change for the reference odour:

$$x = \frac{V_{odour}^{max} - V_{odour}^{min}}{V_{ref}^{max} - V_{ref}^{min}} \tag{4.8}$$

Final fractional difference model (*ff*), this is the ratio of the difference of the final output voltage for the bacteria odour and the reference odour, and the final voltage output for the reference odour:

$$x = \frac{V_{odour}^{final} - V_{ref}^{final}}{V_{ref}^{final}} \tag{4.9}$$

Therefore for sensor i and 'smell' j, there is a feature x_{ij} and consequently $i \times j$ features. Since the sensor interface circuitry outputs sensor signals in such a manner that output voltage V is proportional to resistance R, that is $V \propto R = kV$, therefore these feature models can be considered gas sensor resistance feature models.

None of the gas sensor models so far described use timing parameters; t_{ref}^{min}, t_{ref}^{max}, t_{ref}^{final}, t_{odour}^{min}, t_{odour}^{max} and t_{odour}^{final}. Thus any information in the 'raw' data-set that is directly related to dynamic signal behaviour is not included in any subsequent feature-set. The use of gas sensor models which employ these parameters are explored later in chapter 6.

4.1.2 Normalisation

Normalisation is a generic term which is used to describe any transformation of the variables within a feature-set to lie within certain ranges. There is no general agreement of terms, the technique described here as 'auto-scaling' is also known as 'standardisation'. A set of scaled gas sensor features were produced by applying a linear transformation, the purpose of which was to equalise the vector components in the feature data-sets,

this can improve the performance in many types of classifiers [55]. Notation denoting feature-sets that were not normalised was an n appended to the gas sensor feature model notation.

Auto-Scaling

Auto-scaling is a simple linear scaling of vector components so that each vector component has, across the entire feature-set (i.e. column wise), a mean of zero and a unit variance.

The feature-set contained feature vectors, $\mathbf{X_j} = (x_{1j}, x_{2j}, \ldots, x_{ij})$, each feature vector component was treated as being independent, its mean $\overline{x}_i$ was calculated by:

$$\overline{x}_i = \frac{1}{N} \sum_{j=1}^{N} x_{ij} \tag{4.10}$$

Where N is the number of feature vectors in the feature-set with i components to each vector. The variance σ_i^2 is also calculated over the entire feature-set:

$$\sigma_i^2 = \frac{1}{N-1} \sum_{j=1}^{N} (x_{ij} - \overline{x}_i)^2 \tag{4.11}$$

Therefore the set of scaled feature vectors is calculated such that the vector components have a mean of zero and unit variance:

$$y_{ij} = \frac{x_{ij} - \overline{x}_i}{\sigma_i} \tag{4.12}$$

Auto-scaling assumes that the variables are independent. If the feature-set describes only one class of odour then this assumption may be true, the variables may also conform to a normal distribution. If the feature-set describes many different odour classes then it is more unlikely that the variables are independent, it is more likely that the variables may conform to several normal distributions, each one offset from the other and corresponding to a different odour class. This may not, however, be a problem because the variance is preserved. Notation for auto-scaling was an a, this was appended to the gas sensor feature model notation.

Array Normalisation

Array (or vector) normalisation scales all the components of a feature vector by a constant value so that the vector has unit Euclidean length. For each feature vector, $\mathbf{X_j} = (x_{1j}, x_{2j}, \ldots, x_{ij})$, its components are scaled by a constant, k, given as:

$$k_j = \frac{1}{\sqrt{x_{1j}^2 + x_{2j}^2 + \ldots + x_{ij}^2}} \tag{4.13}$$

Each vector component is scaled by this factor, thus a new feature vector is calculated with a unit length:

$$\mathbf{Y_j} = \sqrt{((kx_{1j})^2 + (kx_{2j})^2 + \ldots + (kx_{ij})^2)} \tag{4.14}$$

Thus, in feature-space, the feature vectors are mapped onto the surface of a unit hyper-sphere. If the odour quality information was related to the angle of the feature vector, and the odour concentration was related to the Euclidean length; this normalisation reduces the concentration dependent information within the feature-set [47]. This assumption may not be valid. However if a particular feature vector is erroneous but small in magnitude, this transformation will amplify the noise it contributes to the feature-set. Therefore the overall noise content of a feature-set may be increased (this may not be a problem for classification by means of ANNs which can be robust to noise). Notation for array (vector) normalisation was a v, which was appended to the gas sensor feature model notation.

Sensor Normalisation

Sensor normalisation scales each feature over the entire feature-set so that it lies in the range [0, +1] (or [-1, +1]). Each sensor feature, x_i, was scaled according to the maximum value, x_i^{max}, and the minimum value, x_i^{max}, for that sensor (i.e. column):

$$y_{ij} = \frac{x_{ij} - x_i^{min}}{x_i^{max} - x_i^{min}} \tag{4.15}$$

Therefore each feature was transformed by a different amount, equalisation occurred. If a particular gas sensor was producing erroneous, but small, values; this scaling would

amplify the noise contributed by that gas sensor. This problem can be countered during the classification process by weighting the relevance of each feature. Notation for sensor normalisation was a s, which was appended to the gas sensor feature model notation.

4.1.3 Non-Gas Sensors

Feature models for the temperature, humidity and rate of gas flow were simple. The output from these sensors were averaged over each measurement phase to give 2 values per sensor per 'smell' cycle. The non-gas signals were a time series of voltages, V_n, and there were N measurements in each measurement phase, therefore the non-gas sensor feature was defined as:

$$x = \frac{1}{N} \sum_{n=1}^{N} V_n \tag{4.16}$$

The frequency of measurement was 1 per second. A 'smell' cycles consisted of 2 measurement phases, each lasting 2 minutes. Therefore in one measurement phase there were 120 individual measurements (i.e. $N = 120$).

4.1.4 Feature-Sets

Table 4.1 and Table 4.2 show how the feature-sets employed in this chapter, were built up from the 'raw' data-sets using the gas sensor feature models and normalisation algorithms presented so far. Each feature-set type had a name and number which identified which features it contained. There were a total of 36 different feature-set types, therefore for the 4 experiments so far conducted, there were a maximum of 144 feature-sets.

Software was written in the C++ language to implement the feature extraction and normalisation. Also, this software allowed the addition of target output vectors to the feature-sets, which were necessary for the training of classifiers. The generation of feature-sets was automated, thus large amounts of data could be generated and handled with more ease.

Table 4.1: Table showing the composition of Feature-Sets using reported gas sensor feature models.

Set No.	Notation	Gas sensor model	Normalisation
1	*dfn*	difference	none
2	*dfs*	difference	sensor
3	*dfa*	difference	auto-scaling
4	*dfv*	difference	vector(array)
5	*rln*	relative	none
6	*rls*	relative	sensor
7	*rla*	relative	auto-scaling
8	*rlv*	relative	vector(array)
9	*fdn*	fractional difference	none
10	*fds*	fractional difference	sensor
11	*fda*	fractional difference	auto-scaling
12	*fdv*	fractional difference	vector(array)

4.2 Artificial Neural Networks: Multiple Layer Perceptron

The most common type of ANN that is used for classification in a wide range of applications, ranging from decision making in insurance companies to the administering of medication, is the multi-layer perceptron (MLP) trained using back-propogation (BP). This is the type of ANN described here. This description is not intended to be comprehensive, MLP ANNs and BP are described in many publications [55, 99]. However from this description the nature of this classification method is highlighted and so the reasons for its widespread application to electronic nose data can be investigated.

A simple mathmatical model has been postulated [73] in order to account for the observed behaviour of the biological neuron, this model is a discriminant function similar to those previously highlighted by Fisher [67]. The following equation describes the neuron model as a non-linear discriminant function for a given neuron i:

Table 4.2: Table showing the composition of Feature-Sets using new gas sensor feature models.

Set No.	Notation	Gas sensor model	Normalisation
13	*afn*	absolute final output	none
14	*afs*	absolute final output	sensor
15	*afa*	absolute final output	auto-scaling
16	*afv*	absolute final output	vector(array)
17	*mnn*	minimum output	none
18	*mns*	minimum output	sensor
19	*mna*	minimum output	auto-scaling
20	*mnv*	minimum output	vector(array)
21	*frn*	final relative	none
22	*frs*	final relative	sensor
23	*fra*	final relative	auto-scaling
24	*frv*	final relative	vector(array)
25	*mdn*	modified difference	none
26	*mds*	modified difference	sensor
27	*mda*	modified difference	auto-scaling
28	*mdv*	modified difference	vecor(array)
29	*mfn*	modified fractional difference	none
30	*mfs*	modified fractional difference	sensor
31	*mfa*	modified fractional difference	auto-scaling
32	*mfv*	modified fractional difference	vector(array)
33	*ffn*	final fractional difference	none
34	*ffs*	final fractional difference	sensor
35	*ffa*	final fractional difference	auto-scaling
36	*ffv*	final fractional difference	vector(array)

$$y_i = \begin{cases} 1 & \text{if } v_i \geq 0 \\ 0 & \text{if } v_i < 0 \end{cases} \tag{4.17}$$

The term v_i represents the level of excitement (or activity) of the neuron which is usually the weighted sum of N inputs, x_j, offset by a threshold θ_i. The term v_i is defined as:

$$v_i = \sum_{j=1}^{N} w_{ij} x_j - \theta_i \tag{4.18}$$

These equations form the basis for the majority of neural network research, including the MLP. The neuron model employed in MLPs is the Perceptron, this does not precisely follow the non-linear discriminant (threshold) model described in equation 4.17, but a 'soft' threshold model, the most popular of which is the sigmoidal logistic function:

$$y_i = \frac{1}{(1 + e^{-\alpha v_i})} \tag{4.19}$$

The term α is a slope term and determines the the slope of the logistic function, it is usually set to 1.0. The effect of these different activation functions, including the sigmoidal logistic function, can be observed in figure 4.2.

Other activation functions include the hyperbolic tangent (tanh) which is another 'soft' threshold function similar to the sigmoid function but is less commonly used (also shown in figure 4.2). The tanh function allows negative inputs to change the activation of the neuron. The sigmoid function 'squashes' the output of the perceptron to the range [0, +1] and the tanh function 'squashes' the output to the range [-1, +1]. The effect of 'squashing' the output is important as it improves the training qualities of the network. Figure 4.3 shows a diagramatical representation of a single perceptron.

From figure 4.3 it can be observed how the models quoted in equations 4.18 and 4.19 combine to model a complete unit. The adaptive threshold, θ_k, is in practice implemented as an extra weighted input fixed at -1. Figure 4.4 shows how a number of perceptrons are connected to form a 'feed-forward' MLP ANN.

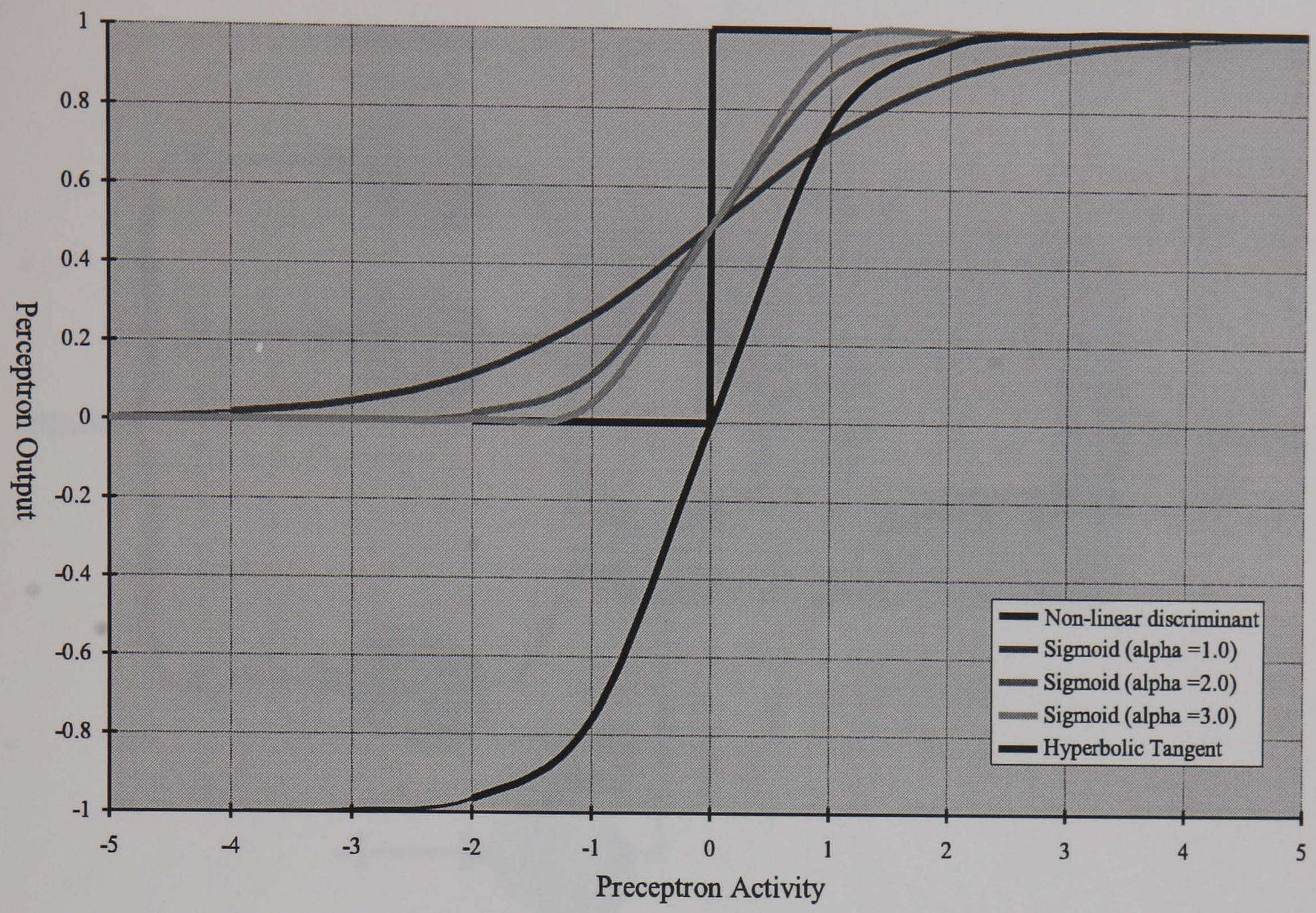


Figure 4.2: A plot showing the different activation functions, including the non-linear discriminant function given in equation 4.17, the sigmoidal logistic function given in equation 4.19 (for 3 different values of α) and the hyperbolic tangent function.

The MLP ANN shown in figure 4.4 is fully connected because all the outputs from the previous layer are input to the next layer. Partially connected MLPs are less common. The term 'feed-forward' is used because each layer feeds forward into the next. It is possible for the inputs to the network to connect directly to the output layer (i.e. bypassing the hidden layer), this arrangement of connections can be beneficial but also lead to problems with training. Also more than 1 hidden layer can be used, however MLPs with more than two hidden layers (or none at all) exhibit different classifying qualities. It has been shown that a MLP with 1 hidden layer, the perceptrons of which have non-linear activation functions (such as sigmoid), can be a universal approximator [99]. Therefore any continuous function can be approximated (including non-linear functions). Adding a second hidden layer can improve classifier performance but in practice is more problematic in training because it increases local minima. With

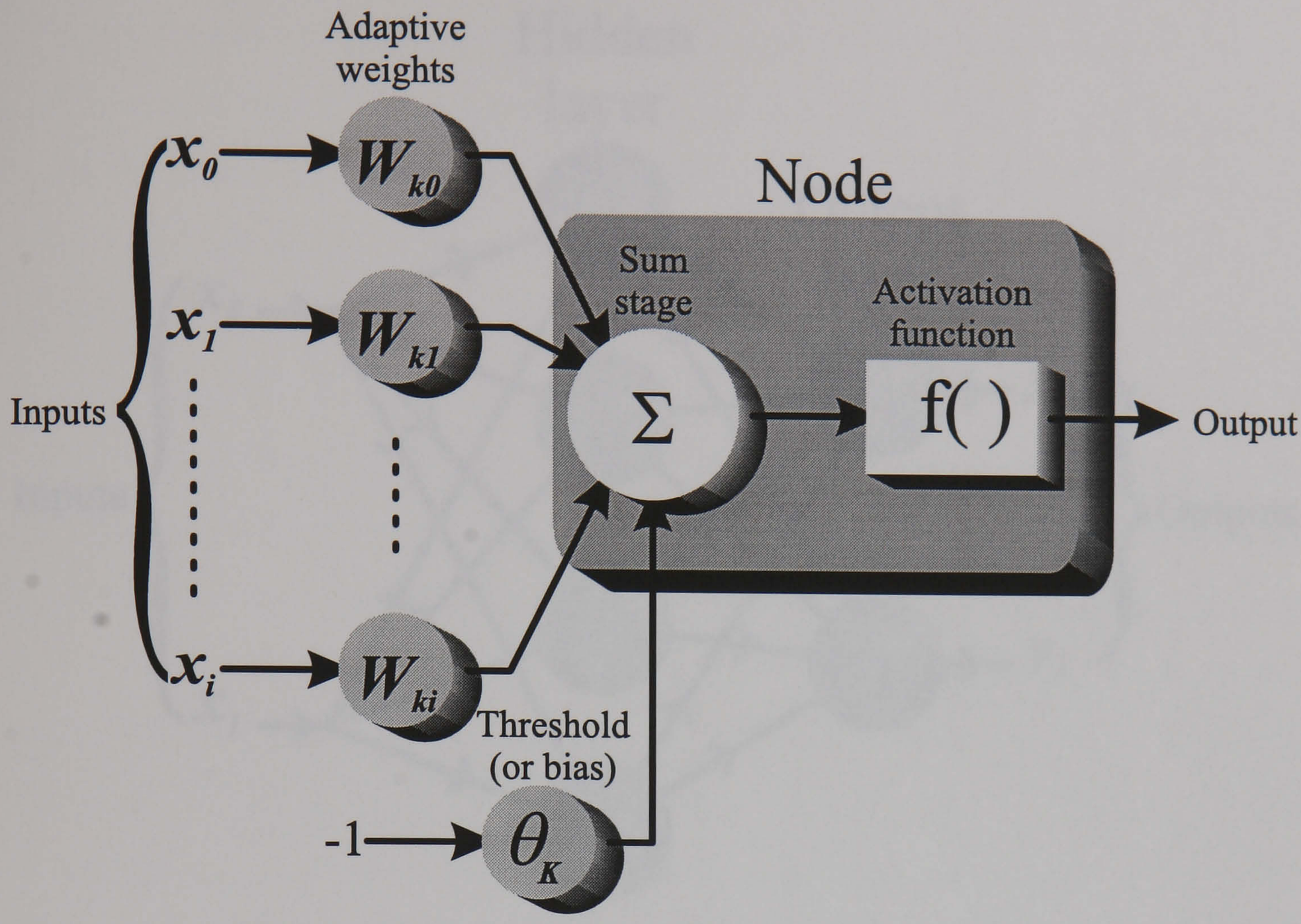


Figure 4.3: Diagram showing the construction of a single perceptron having i inputs and 1 output. The implementation of an adaptable threshold level is also shown.

no hidden layer the network can only approximate linear functions, therefore it would implement a linear discriminant function in feature space. This is not sufficient for complex odour classification [47].

MLPs are usually trained using BP[2]. BP is a supervised training technique where for each input vector, the target output vector is known. The target vector is the desired response from the network to the input vector. Before training can commence the weights have to be initialised. This is usually done by assigning a random number to each weight, often in the range [-1, +1] or [-0.5, +0.5]. Assigning large initial weights can lead to poor training solutions being found. The overall objective of training is to minimise the difference between the actual network output and the target output. This is achieved by adjusting the weights and thresholds for each perceptron. In order

[2]There are other methods but BP is by far the most common. A study of all possible training methods is beyond the scope of this research.

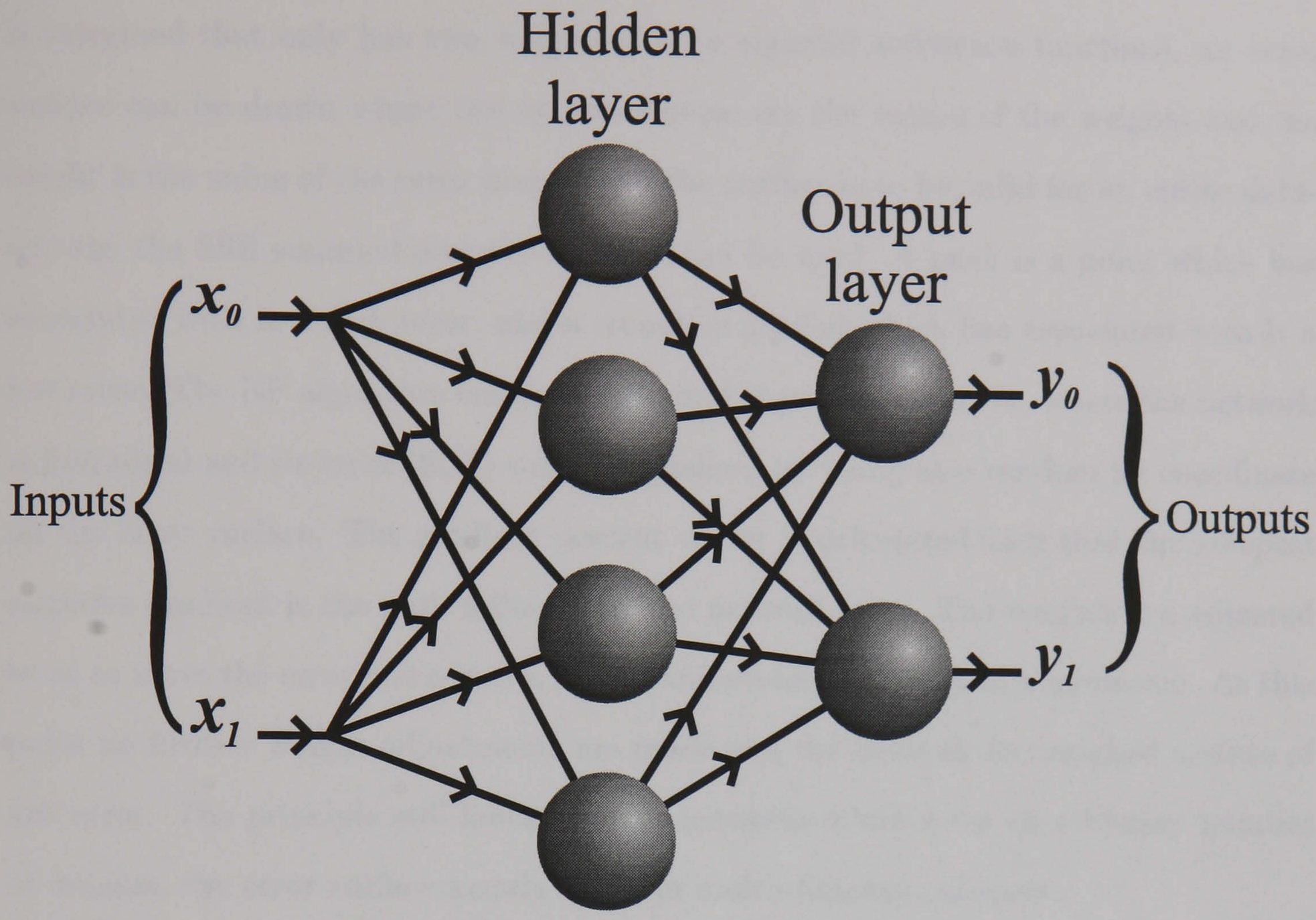


Figure 4.4: Diagrams showing the arrangement of perceptrons to form a 2 layer MLP ANN with 2 input perceptrons, 2 outputs (therefore 2 output perceptrons) and 4 hidden perceptrons. Note that the number of layers of adaptive weights is used to denote the number of layers of a MLP.

to do this, an error measure is computed, the most common measure is the sum of the square of the difference for each network output. This measure, called sum of squared errors (SSE) can be written as (for N outputs):

$$\varepsilon_{sse} = \sum_{n=1}^{N} (t_n - y_n)^2 \tag{4.20}$$

Where t_n is the target for output n and y_n is the actual value for output n. Many adaptions to this measurement are used, such as mean squared error (MSE) which is simply SSE divided by the number of outputs, N, to get a mean value for each output. Also the error measure is computed for each input vector and summed over the whole data-set (this is done throughout this research). A common analogy used in the field of ANNs is the 'error surface', which exists in the weight-space (domain). If a network

is imagined that only has two weights (and a sigmoid activation function), an error surface can be drawn where the xy coordinates are the values of the weights and the height is the value of the error measure. If the surface is to be valid for an entire data-set, the the SSE summed over the data-set can be used. A peak is a point which has associated with it a high error, and a trough is a point which has associated with it a low error. The BP algorithm employs a method of gradient descent, where the network is initialised and its error (SSE) can be visualised by being at a random xy coordinate on the error surface. The gradient descent vector is calculated such that the steepest negative gradient is the path followed by the network error. The weights are adjusted so as to move the error down into a trough and eventually finish in a minimum. At this point no further weight adjustments are made and the network has reached a state of low error. The principle still holds true for networks which have an arbitrary number of weights, the error surface merely exists in multi-dimensional space.

At this point, BP training algorithm can be formally described. First the error derivative is calculated, which is simply the ratio of the rate of change of error, ε_{sse}, over the rate of change of output, y_n:

$$\frac{d\varepsilon_{sse}}{dy_n} = y_n - t_n \tag{4.21}$$

Next the output derivative is calculated for each output perceptron, which is simply the ratio of the rate of change of error over the rate of change of total input, x_n:

$$\frac{d\varepsilon_{sse}}{dx_n} = \frac{d\varepsilon_{sse}}{dy_n}\frac{dy_n}{dx_n} \tag{4.22}$$

Assuming an sigmoid activation function and substituting equation 4.21, equation 4.22 can be simplified further to:

$$\frac{d\varepsilon_{sse}}{dx_n} = (y_n - t_n)y_n(1 - y_n) \tag{4.23}$$

Where $y_n(1 - y_n)$ is the derivative of the sigmoid function. It is no accident that the sigmoid function is so popular; its derivative is a simple function. It can now also be seen why the non-linear discriminant function given in equation 4.17 is not used.

Its derivative is either 0 or ∞ which makes calculation of the gradient descent vector impossible. From this, the weight derivative for each weight can be calculated, which is simply the ratio of the rate of change of error over the rate of change of a weight for the corresponding output perceptron, $w_m n$:

$$\frac{d\varepsilon_{sse}}{dw_{mn}} = \frac{d\varepsilon_{sse}}{dx_n}\frac{dx_n}{dw_{mn}} \tag{4.24}$$

Simplifying and substituting equation 4.23:

$$\frac{d\varepsilon_{sse}}{dw_{mn}} = (y_n - t_n)y_n(1 - y_n)y_m \tag{4.25}$$

Where y_m is the output from a hidden perceptron. From this the error derivative for the output from a hidden neuron can be calculated. This is the important stage because it is here that the error is 'back-propagated'. This product of derivations is called the chain rule. If the output from a single hidden perceptron changes, it effects all the output perceptrons (assuming full connection), the effect on each output perceptron is therefore summed:

$$\frac{d\varepsilon_{sse}}{dy_m} = \sum_{n=1}^{N}\frac{d\varepsilon_{sse}}{dx_n}\frac{dx_n}{dy_m} \tag{4.26}$$

Simplifying and substituting equation 4.23:

$$\frac{d\varepsilon_{sse}}{dy_m} = \sum_{n=1}^{N}(y_n - t_n)y_n(1 - y_n)w_{mn} \tag{4.27}$$

These steps can be progressed further to find the weight derivative for a hidden perceptron. Thus all the derivatives for all weights in an MLP can be calculated. Summarising, a MLP has a vector applied to its inputs, from the resultant output the error measure is computed. Given that each weight currently has a particular value, and using the weight derivative calculations for the hidden and output layers, a gradient descent vector, ∇, is computed. This vector has a many components as there are weights in the MLP. The weights in the output layer are updated using the following rule:

$$w_{mn}(t+1) = w_{mn}(t) - \eta \nabla_{mn} \tag{4.28}$$

Where $w_{mn}(t)$ is the value of the weight between output perceptron n and hidden perceptron m at time t. Also ∇_{mn} is the gradient descent vector component for weight w_{mn} and η is the learning rate constant. This same equation can be used for any two layers by substituting the corresponding weight value and gradient descent vector component value. The threshold value is considered to be an extra weight in each perceptron.

Each input vector used in the training process is called a training vector and has an associated output target vector, i.e. a training vector pair. The data-set that contained the training vector pairs was called the training data-set. Each training vector pair in the training set was presented in turn to the network. In general weights can either updated with every training pair (called on-line method) or after all the training pairs have been applied (called batch method). Batch mode is the less popular mode. For training in on-line mode, each training vector pair is applied to the network, the error is calculated and thus the gradient descent vector is calculated. From this the network weights are adapted according to the update rule (see equation 4.28). After all the training vector pairs have been applied to the network, a criterion is measured in order to determine if the training should stop or carry on (by applying the training pairs once more). This criterion can be that the weights have ceased to be adjusted or that a particular minimum error has been reached.

There are a large number of variants of BP. Despite this, there is a more advanced version of BP that is probably more popular than the original algorithm. This advanced version is called BP with momentum. The reason for its popularity is because of a serious problem with the original BP. Previously the idea of an error surface was described with peaks and troughs. The problem is that the network can get stuck in a trough which is not the deepest (called a local minimum), i.e. the network stabilises without reaching the global minimum error. There is no method of training so far (BP or otherwise) for a MLP that guarantees that the global minimum is reached. BP with momentum helps the network pass through high local minima and encourages the

network to stabilise in deep minima (possibly the global minimum). The weight update rule is modified thus (with $\Delta w_{mn}(t) = -\eta \nabla_{mn}(t)$):

$$w_{mn}(t+1) = w_{mn}(t) + (\Delta w_{mn}(t) + \alpha(\Delta w_{mn}(t-1))) \tag{4.29}$$

Where α is the momentum coefficient. It can be observed how the weight change depends not only on the current gradient descent vector but also the one for the previous stage. There are limitations to BP with momentum, the network can become difficult to stabilise, and the training time may be increased. In reality, most learning algorithms improve some features at the expense of others.

Now that the basics have been covered, some limitations of using BP trained MLPs for classification can be identified and their importance to classification of odours can be explored.

ANNs were modeled using a software package called SNNS (Stuttgart Neural Network Simulator), version 4.1. This package allows a wide range of ANN types to be modeled and applied. ANNs were designed using a graphical interface, these ANNs were then trained and tested on data using an internal processing language (which also allowed automated systematic training and testing of ANNs). Lastly, ANN performance was analysed using built-in analysis software. This program is public domain (i.e. freely available) and is supplied with source code (written in C language), modifications to the software were therefore possible. Although many other 'professional' packages exist, for example NeuralWorks (by NeuralWare), the user base of SNNS has become the biggest of any ANN software package and has therefore become the most tried and tested. Also, the author's previous experience with SNNS meant quick implementation of ANN models.

4.2.1 Classification Of Bacteria Type

Combined data from the first experiments with *Escherichia coli* and *Staphylococcus aureus* (i.e. experiment 1 and 3) was used in conjunction with MLP ANNs which were trained using BP with momentum. These ANNs were tested using combined data from the second experiments with *Escherichia coli* and *Staphylococcus aureus*

(i.e. experiment 2 and 4), this is called split-sample validation. MLPs trained and tested with such a combination were denoted using '13/24' as notation. The reason for combining the data in this manner was that it was a more challenging test for a classifier to classify data from one experiment when being trained from another. In this way, variances between experiments for the same bacteria type cannot be learned (which would have biased the test results), the classifier is forced to learn the variances due to bacteria type only. No doubt more impressive performance figures could have been attained if the classifiers were both trained and tested with data from the same experiment, this would have been less useful. Great effort was invested into obtaining data of sufficient amount that more realistic performance estimates could be calculated, rather than relying on other performance estimation techniques such as cross-validation (see [58]). All the feature-set types listed in Table 4.1 and Table 4.2 were used, therefore there were 36 training feature-sets and 36 testing feature-sets, each feature-set contained 360 vectors (180 for each bacteria type). Each ANN was trained and tested using feature-sets of the same type (i.e. the same notation). Target output vectors were added to all feature-sets, this was easy since the prior class membership was obviously known in all cases.

The network topology was 6 (12 for feature-sets using the Absolute Final Output and Minimum Output gas sensor feature models) inputs, 20 hidden nodes and 2 output nodes. The classes themselves were encoded using 1-of-C where each output node corresponded to 1-of-C classes, there were 2 classes therefore 2 output nodes. This method of encoding unordered categories (classes) is standard. The transfer function for the input nodes was unity (i.e. the output was the same as the input), because these nodes simply distribute the input vectors to the hidden nodes. The hidden and output nodes had hyperbolic tangent (tanh) transfer functions instead of the more common sigmoid function because some input vector components had negative values[3]. Therefore the maximum output value for both hidden and output nodes was 1 and the minimum was -1.

[3] A sigmoid function would output 0 for all negative inputs, thus all information contained in negative input vector components would be lost and classification performance reduced.

The large number of hidden nodes was considered necessary because an early-stopping training technique was used. Early-stopping is a common technique used to improve the generalisation performance of an ANN. Generalisation is the ability of the ANN to correctly classify input vectors which have not been used for training. The usefulness of an ANN that can correctly classify all of its training vectors without consideration for the testing vectors, is very limited. The number of hidden nodes has a significant influence of the ability of an ANN to generalise [55] and yet no reliable method for estimating the optimum number of hidden nodes has so far been discovered. A MLP with too few hidden nodes can 'under-fit' the data. Essentially the MLP is approximating a mapping function from the input vector (the feature-set) domain to the output (class membership) domain. 'Under-fitting' relates to the fact that the function approximation can be inaccurate due to the inability to accommodate enough of the training vectors. Conversely, a MLP with too many hidden nodes can 'over-fit' the data where outlying training vectors unduly influence the function approximation and introduce error. The number of cycles the MLP is trained also affects the generalisation performance, a MLP trained with too few cycles will 'under-fit' the training data, a MLP with trained for too many cycles may 'over-fit' the training data. This problem was tackled by using a MLP with a large number of hidden nodes (e.g. 10 times the number of classes), which was trained in the standard way for up to 1000 cycles, but after each cycle the SSE over entire testing data-set was computed, a 'snapshot' of the weights of the network which yielded the lowest SSE was recorded to allow later use. This method was not strictly early-stopping because it did not stop when the SSE started to rise after the initial fall. The software that was employed allowed a 'snapshot' of an ANN to be recorded, it was possible to train over a fixed period.

Table C.1 and Table C.2 summarise the results obtained from the application of a MLP to the classification of bacteria type, the former table shows the results for the feature-set types previously employed by other workers, and the later table shows the results for the new feature-set types. Correct network outputs were calculated using the '402040' rule [112], in this case a lower output band was defined as the range [-1, -0.5] and an upper output band was defined as the range as the range [0.5, 1]. A

pattern was correctly classified if precisely 1 output was in the upper band, all the other outputs were in the lower band, and the highest teaching output corresponded to the highest output. A pattern was incorrectly classified if precisely 1 output was in the upper band, all other outputs were in the lower band, and the highest teaching output did not correspond to the highest output. A pattern was unknown if it was neither correctly classified nor incorrectly classified. Specifying output vectors in this manner allowed for unknown feature vectors to be more accurately detected. A rule such a winner-takes-all (WTA) forces possible unknown classifications to be classified thus possibly increasing the number of incorrectly classifications[4]. The training parameters used were η =0.001, α =5.0 (see equations 4.28 and 4.29), c =0.1 (flat-spot elimination constant added to derivations) and d_{max} =0.1 ($d = (t_n - y_n)$, maximum tolerance of error per output). The relatively low value for μ and high value for α help reduce the effect of local minima, which is necessary for MLPs with a large number of hidden nodes, such as those used here. The vectors in the training set were shuffled into a random order for each training cycle, therefore there was no constant order in which the vectors were input to the net for training, this can improve generalisation. Also each training/testing data-set pair was used to train a 10 MLPs, therefore if 1 of the MLPs did not converge well due to the initial random weights, there was a good chance of a better MLP being produced. The random weights were assigned in the region [-0.5, +0.5].

The tables summarising the performance of the MLP classifiers are given in appendix C, in Table C.1 and Table C.2. The SSE term was calculated over the entire testing feature-set and the other performance measurements are given as a % of the total number of patterns (feature vectors) in the testing feature-set. Further to these results, it was decided to repeat the same neural net analysis but reversing the training and testing feature-sets. Consequently, the MLPs were tested with feature-sets from experiments 1 and 3, and tested with data from experiments 2 and 4, the results of this

[4]Although it can also give a more immediately impressive figure for the number of correct classifications. It does not increase the difference between the number of correct classifications and incorrect classifications, which is a more important measure.

Table 4.3: Average and standard deviation of performance of bacteria type classification by means of MLPs using data from different experiments for training and testing.

Train/	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Test	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
13/24	552.12	228.47	47.70	28.22	10.76	7.62	41.54	30.51
24/13	512.31	202.00	51.84	28.57	9.94	7.64	38.22	31.79

analysis are given in Table C.3 and Table C.4 in appendix C. MLPs trained and tested with such a combination were denoted using '24/13' as notation. This reversal of training and testing data-sets was valuable because it further tests the robustness of MLP classifiers. It is possible for the vectors in a feature-set from an experiment to describe a larger space in feature-space than a feature-set from a different experiment, thus a MLP trained with former feature-set can exhibit better generalisation. Comparing the results for the same feature-set pairs in Table C.1 and Table C.2 to those in Table C.3 and Table C.4, it can be seen that reversing the training and testing feature-sets did effect performance. Table 4.3 lists the averages and standard deviations of different performance measures for both sets of results. Using the average, $\bar{x}$, and standard deviation, σ, of performance measures (SSE over the whole training feature-set, % correct, % incorrect and % unknown), 13/24 MLPs tended to be marginally inferior to 24/13 MLPs. A large value of σ highlights a quantity that is prone to high variance, i.e. not particularly consistent. Because there was no great difference in performance it can be assumed, at this point, that no one experiment provided data that was significantly better or worse than the others.

Table 4.4 shows not the average and standard deviation of performance measures (Table 4.3) but the minimum and maximum. Again, no significant difference between the two sets of MLPs seems to be evident. However the corresponding minimums and maximums for the two sets of MLPs did tend to occur with different feature-set types. For example, the best % correct classifications for 13/24 MLPs occurred for the ***mnn*** Feature-Set, whilst that for 24/13 MLPs occurred equally for the ***frs*** and ***ffs*** feature-set

Table 4.4: Minimum and maximum of performance of bacteria type classification by means of MLPs using data from different experiments for training and testing.

Train/	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Test	min	max	min	max	min	max	min	max
13/24	74.21	1085.51	0.28	96.11	0.00	34.17	0.00	99.72
24/13	52.90	947.90	0.00	93.06	0.00	24.44	4.44	100.00

types.

SSE is an indication of the 'depth' of the minimum in weight space that was converged upon during training. In general the lower the SSE, the higher the % of correct classifications, the lower the % of incorrect classifications, and the lower the % of unknown classifications. When unknown and incorrect classifications tended to differ greatly with the target outputs, the SSE increased even though the % correct classifications remained similar. MLPs where the SSE was very high and the % unknown classifications was also high (90% or more) can be judged to not have trained correctly, no significant minimum on the error surface was converged upon. The better MLPs had a low SSE along with, high % correct, low % incorrect and low % unknown classifications. These criteria were used to define a 'good' MLP.

From results so far obtained, an idea of the effect of different pre-processing algorithms upon the performance of bacteria type classification can be obtained. For different gas sensor feature models, Table 4.5 and Table 4.6 summarise the results for both MLP sets.

Looking at the results from Table 4.5, the *df* feature model resulted in the best MLP performance, and the *mf* feature model resulted in the worst MLP performance. The relatively low value of σ for *df* indicates that it consistently performs well compared to the other gas sensor feature models. Because these statistics cover all the different normalisation algorithms, it is possible for a particular feature model to perform well with one normalisation algorithm and badly with another. Such behaviour increased the corresponding value for σ. This fact is further enhanced by observation of the

Table 4.5: Average and standard deviation of performance of bacteria type classification by means of MLPs for different gas sensor feature models (see equations for model notation).

Model	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
df	346.94	128.53	77.22	9.19	10.28	4.30	12.50	6.73
rl	547.24	124.13	40.00	25.68	8.99	7.07	51.01	30.37
fd	543.13	125.22	43.54	22.97	10.17	4.07	46.28	24.75
af	388.86	164.16	61.11	14.36	7.22	6.42	31.67	13.28
mn	417.96	177.99	60.14	24.86	9.86	6.90	30.00	24.67
fr	558.73	223.16	36.01	39.92	6.74	8.95	57.26	45.69
md	685.88	250.56	61.35	13.93	21.08	9.04	17.57	12.34
mf	705.44	188.79	23.89	26.69	8.54	4.98	67.57	29.37
ff	595.86	264.34	44.69	32.56	10.24	7.71	45.07	34.81

minimums and maximums listed in Table 4.6. Thus good individual performance (i.e. performance for a particular feature model and normalisation algorithm) was achieved for *df*, *fd*, *mn*, *fr* and *ff* gas sensor feature models, some of which also yield low values for $\bar{x}$ and high values for σ.

As well as feature model, it was also important to consider the effect of normalisation upon classifier performance. Table 4.7 and Table 4.8 are similar to Table 4.5 and Table 4.6 respectively except that performance was measured for all feature models, for a specific normalisation algorithm.

From Table 4.7, auto-scaling appears to perform well. However its low values of σ also highlight that it performed consistently and not the best. This fact is confirmed by observation of Table 4.8. Also, all the other normalisation algorithms gave both good and bad results. From this information it can be said that auto-scaling did not improve MLP classification performance, in fact it tended to decrease performance.

The best MLP used the Minimum Output feature model and sensor normalisation

Table 4.6: Average and standard deviation of performance of bacteria type classification by means of MLPs for different gas sensor feature models (see equations for model notation).

Model	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
	min	max	min	max	min	max	min	max
df	82.18	514.93	64.72	93.89	2.50	15.83	3.06	20.28
rl	330.09	680.35	0.00	71.11	0.28	21.39	18.61	99.72
fd	304.56	723.62	15.28	79.72	3.89	17.78	10.28	75.00
af	226.19	750.79	42.22	82.22	2.50	21.94	12.22	55.00
mn	74.21	625.15	20.28	96.11	1.67	19.44	0.00	78.06
fr	52.90	714.34	0.00	93.06	0.00	21.11	6.67	100.00
md	292.80	1085.51	44.17	78.33	3.89	34.17	4.44	36.11
mf	422.78	933.21	1.11	71.39	0.00	13.33	16.11	98.89
ff	52.90	947.90	0.00	93.06	0.00	21.11	6.67	100.00

Table 4.7: Average and standard deviation of performance of bacteria type classification by means of MLPs for different normalisation algorithms. Key to notation: n = none, s = sensor normalisation, a = auto-scaling and v = array (vector) normalisation.

Norm.	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Type	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
n	533.29	214.35	46.94	32.38	9.60	8.01	43.46	35.92
s	528.41	279.02	54.21	30.32	10.65	8.29	35.14	31.07
a	515.39	109.54	61.88	10.58	14.40	5.35	23.72	12.78
v	551.80	237.97	36.05	29.55	6.74	6.87	57.21	30.94

(*mns* Feature-Set), 96.11% of all vectors were correctly classified, 2.22% were incorrectly classified and 1.67% were unknown, SSE was 74.21, and used 13/24 Feature-Sets. A good method for visualisation of classifier performance is a confusion matrix, Table 4.9

Table 4.8: Minimum and maximum of performance of bacteria type classification by means of MLPs for different normalisation algorithms. Key to notation: n = none, s = sensor normalisation, a = auto-scaling and v = array (vector) normalisation.

Norm.	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Types	min	max	min	max	min	max	min	max
n	271.12	1085.51	0.00	88.33	0.00	34.17	0.00	100.00
s	52.90	933.21	0.00	96.11	0.28	27.50	1.67	99.72
a	349.96	680.35	46.11	76.39	4.17	24.44	7.78	44.44
v	82.18	947.90	0.00	93.89	0.00	25.58	3.61	100.00

Table 4.9: Confusion matrix for best single result for bacteria type classification, using Minimum Output feature model and sensor normalisation, experiments 1 and 3 for training and experiments 2 and 4 for testing.

Actual Output	Target Output	
	Escherichia coli	*Staphylococcus aureus*
Escherichia coli	166	0
Staphylococcus aureus	8	180
Unknown	6	0

is a confusion matrix for the output of the best MLP. From this matrix, it can be observed that ***Staphylococcus aureus*** was correctly classified 100% of the time and that ***Escherichia coli*** was correctly classified 92.22% (166 correct out of 180 total) of the time. Conversely, when ***Escherichia coli*** was actually output, the MLP was correct 100% of the time, and when ***Staphylococcus aureus*** was actually output, the MLP was correct 91.01% of the time. This highlights the fact that there may be unwanted variance within the data representing ***Escherichia coli.***

4.2.2 Classification of Culture Growth Phase

Bacteria culture growth phase was classified (i.e. predicted) using similar methods to those used to classify bacteria type. It was postulated that classification of bacteria type might be improved by inputting, along with the sensor feature vector, information relating directly to culture growth phase. Growth phase could be classified (or predicted) by a different functional phase (e.g. MLP). It could be argued that a classifier that was classifying bacteria type could also internally learn to become invariate to information that showed variance due to culture growth phase, however it could also be argued that such a classifier would be more prone to complications during design and training. Separation of classification of growth phase and type might therefore be beneficial.

The MLP design, topology, training methods and testing methods employed were identical to that used for classification bacteria type with the exception of the number of output nodes. Since there were 3 growth phases, it was necessary to employ 3 output nodes instead of 2. Another important consideration was the imbalance of class membership within feature-sets, where the target vectors reflected growth phase. Previously, there were equal numbers of vectors for each class (bacteria type), this was not the case here. Training MLPs where class membership within the training data is imbalanced can cause problems because the network learns the vectors associated with the predominant class more forcefully. Generalisation of the less predominant classes could therefore suffer. This problem was remedied by simply training using the vectors corresponding to the less predominant class more often. For example, if class A has 25% of the total number of vectors and class B has 75%, the vectors for class A are used 3 times for each use of the vectors for class B[5].

Similar to bacteria type, for growth phase, 2 sets of MLPs were modeled, 13/24 and 24/13. Table 4.10 and Table 4.11 summarise the results over both sets of MLPs.

From Table 4.10, there does not appear to be any significant performance difference between the two sets of MLPs, however from Table 4.11, the MLPs from the 24/13

[5] Another method is the change the learning rate parameter for vectors of different classes, unfortunately the software employed did not allow this.

Table 4.10: Average and standard deviation of performance of culture growth phase classification by means of MLPs using data from different experiments for training and testing.

Train/	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Test	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
13/24	699.69	119.88	24.96	21.87	8.22	7.63	66.27	28.24
24/13	741.41	165.91	25.68	23.65	10.14	10.27	64.18	31.10

Table 4.11: Minimum and maximum of performance of culture growth phase classification by means of MLPs using data from different experiments for training and testing.

Train/	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Test	min	max	min	max	min	max	min	max
13/24	529.03	997.97	0.00	70.00	0.00	26.11	11.67	100.00
24/13	390.71	1168.41	0.00	80.28	0.00	31.67	6.11	100.00

data-sets did show slightly better performance. The effect of different gas sensor feature models upon MLP performance was highlighted by compiling statistical measurements that described each different feature model, these are given in Table 4.12 and Table 4.13.

From these results, the modified fractional difference (*mf*) feature model did not perform well, MLPs using this model never really converged on any minimum in the error surface. It also appears that the Absolute Final Output feature model (*af*) performed almost consistently better than other models. Apart from these two feature models, there was little significant difference between the performance of the other feature models.

It can be noted, looking at the results so far for bacteria type classification and growth phase prediction, that different feature models perform better for different classification problems. This further suggests that separate classification of culture growth phase and bacteria type might be advantageous compared to a single, complex, classi-

Table 4.12: Average and standard deviation of performance of culture growth phase classification by means of MLPs for different gas sensor feature models (see equations for model notation).

Model	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
df	677.72	115.47	41.88	10.80	12.08	7.86	46.33	16.33
rl	732.95	127.83	20.52	28.40	7.12	10.19	72.36	38.03
fd	688.99	116.26	23.26	15.47	9.58	8.89	64.65	20.67
af	603.51	147.95	44.65	31.43	9.83	6.21	45.52	36.38
mn	829.38	152.64	25.83	19.00	14.86	9.16	59.31	26.32
fr	627.06	101.37	25.28	19.73	7.33	8.94	67.40	27.45
md	849.05	150.60	22.64	18.68	13.72	11.42	63.65	28.15
mf	771.28	31.34	0.45	1.28	0.24	0.48	99.31	1.74
ff	671.25	127.66	23.37	21.15	7.85	8.80	68.79	28.60

fier.

Different normalisation algorithms were also investigated, Table 4.14 and Table 4.15 list the statistics for the performance measurements for the MLPs over different feature models but with the same normalisation algorithm.

A ranking for normalisation algorithms was derived using the same criteria as that employed for ranking feature models:

1. Auto-scaling (a)

2. Sensor normalisation (s)

3. No normalisation (n)

4. Array (vector) normalisation (v)

In common with the ranking table for feature model, this ranking is different from the corresponding ranking for classification of bacteria type.

Table 4.13: Average and standard deviation of performance of culture growth phase classification by means of MLPs for different gas sensor feature models (see equations for model notation).

Model	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
	min	max	min	max	min	max	min	max
df	478.14	870.27	25.28	53.61	4.17	26.11	25.83	70.28
rl	529.03	898.21	0.00	58.06	0.00	25.56	23.89	100.00
fd	579.11	909.35	0.28	48.33	0.83	26.67	25.00	98.89
af	390.71	835.83	0.00	80.28	0.00	18.33	6.11	100.00
mn	695.03	1168.41	0.00	57.78	0.00	28.61	24.17	100.00
fr	531.55	846.49	1.11	52.50	0.56	27.50	21.39	98.33
md	664.87	1121.57	0.00	48.89	0.00	31.67	30.83	100.00
mf	731.27	823.08	0.00	3.61	0.00	1.39	95.00	100.00
ff	531.55	846.49	0.56	52.50	1.11	27.50	21.39	98.33

Table 4.14: Average and standard deviation of performance of culture growth phase classification by means of MLPs for different normalisation algorithms. Key to notation: n = none, s = sensor normalisation, a = auto- scaling and v = array (vector) normalisation.

Norm.	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Type	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
n	742.03	101.54	16.17	16.63	6.84	6.56	76.99	21.51
s	656.26	139.80	31.85	23.75	7.64	6.57	60.49	28.79
a	776.29	205.34	40.43	22.75	17.76	10.68	40.69	28.56
v	692.62	61.16	12.82	15.42	4.46	5.53	82.72	20.00

It has been suggested [47] that array normalisation reduces the concentration dependent information, since the growth phase is related to the size of the bacteria pop-

Table 4.15: Minimum and maximum of performance of culture growth phase classification by means of MLPs for different normalisation algorithms. Key to notation: n = none, s = sensor normalisation, a = auto-scaling and v = array (vector) normalisation.

Norm.	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Types	min	max	min	max	min	max	min	max
n	581.18	928.13	0.00	48.89	0.00	20.83	30.83	100.00
s	390.71	878.47	0.00	76.11	0.00	21.11	13.89	100.00
a	423.38	1168.41	0.00	80.28	0.28	31.67	6.11	99.72
v	572.03	780.29	0.00	47.78	0.00	19.44	43.61	100.00

ulation within a culture, it is logical to speculate that greater numbers of bacteria produce a greater concentration of odour. Therefore concentration dependence information was being learnt in order to classify growth phase rather than qualitative odour information.

The best MLP used the Absolute Final Output feature model and auto-scaling (*afa* feature-set), when 80.28% of all vectors were correctly classified, 13.61% were incorrectly classified and 6.11% were unknown, SSE was 423.38, and used 24/13 Feature-Sets. Table 4.16 is the confusion matrix for the best MLP, where the imbalance in the class representation within the test feature-set can be observed (the Lag phase has only 14 patterns). The figures represent the number of patterns classified. Classification of the Lag phase was the most problematic (only being correctly classified 14.29%), most often the Lag phase was mistaken for the Log phase. The other classes performed well, the Log phase was correctly classified 95.10%, and the Static phase was correctly classified 73.10% of the time. It was subsequently discovered that most of the erroneous classification occurred for patterns that existed near the boundary between growth phases. This may indicate that the hard boundaries used to derive the target output vectors are unsuitable. The fact that the Log phase was most easily classified may be related to the particular metabolism of the bacteria cells in this phase.

Table 4.16: Confusion matrix for best single result for culture growth phase classification, using Absolute Final Output feature model and auto-scaling, using experiments 2 and 4 for training and experiments 1 and 3 for testing.

Actual Output	Target Output		
	Lag	Log	Static
Lag	2	2	0
Log	12	154	32
Static	0	1	133
Unknown	0	5	17

4.3 Principal Component Analysis

Principal component analysis (PCA) has been used [50] as a tool for data visualisation. The feature vectors generated by the gas sensor models described in section 4.1.1 have many dimensions, although viewing these vectors in 2 or 3 dimensions is problematic. PCA is essentially a dimension reduction technique where vectors were linearly projected onto a two dimension plane. Generally PCA analysis reduces the amount of information in the data, the objective is to achieve minimum discriminatory (i.e. relevant) information loss. Also PCA is an 'unsupervised' technique where 'target' (i.e prior classification knowledge) information is not used, but rather correlations between the clustering from PCA plots and 'targets' can be analysed. The idea behind PCA is relatively straightforward.

Let a feature-set containing feature vectors $\mathbf{X_j} = (x_{1j}, x_{2j}, \ldots, X_{ij})$ have the covariance matrix $\sum$ with eigen values $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$.

Let a vector $\mathbf{X}$ be transformed using linear coefficients z:

$$\begin{aligned}
\mathbf{Y_{1j}} &= z'_{1j}\mathbf{X_j} = z_{11}x_{1j} + z_{21}x_{2j} + \cdots + z_{i1}x_{ij} \\
\mathbf{Y_{2j}} &= z'_{2j}\mathbf{X_j} = z_{12}x_{1j} + z_{22}x_{2j} + \cdots + z_{i2}x_{ij} \\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
\mathbf{Y_{ij}} &= z'_{ij}\mathbf{X_j} = z_{1p}x_{1j} + z_{2n}x_{2j} + \cdots + z_{ii}X_{ij}
\end{aligned} \tag{4.30}$$

Let the variance for a particular transformation be:

$$Var(\mathbf{Y_{nj}}) = z'_{nj} \sum z_{nj}, \qquad where\ n = 1, 2, \cdots, i \tag{4.31}$$

And the covariance be:

$$Cov(\mathbf{Y_{nj}}, \mathbf{Y_{mj}}) = z'_{nj} \sum z_{mj}, \qquad where\ n, m = 1, 2, \cdots, i \tag{4.32}$$

Those uncorrelated combinations, $\mathbf{Y_{ij}}$, which, when their variance is calculated across the whole feature-set, have as large a variance as is possible can be considered as the principal components for that feature-set.

The first principal component is that value $\mathbf{Y_1}$ which has a set of coefficients that maximises the variance as defined by both equation 4.31 and $z'_1 \times z_1 = 1$ (in order to eliminate indeterminacy of scaled of coefficient increasing the variance, by limiting attention to coefficient vectors of unit length).

The second principal component is that value Y_2 which has a set of coefficients that maximises the variance as defined in equation 4.31, that the covariance of Y_1 and Y_2 as defined in equation 4.32 is zero and that $z'_2 \times z_2 = 1$.

So the n^{th} principal component (PC) can be defined as:

$$\begin{aligned} PC_n = \quad & Y_n\ that\ maximises\ Var(Y_n)\ with \\ & z'_n z_n = 1\ and \\ & Cov(Y_n, Y_m) = 0\ for\ m < n \end{aligned} \tag{4.33}$$

It therefore follows that for a feature-set containing feature vectors with i dimensions (components), PCA will yield i PCs. The PCs are then ranked so that PC_1 is the PC with the highest variance and PC_i is the PC with the lowest variance. A threshold can be set, for example at 85%, such that only PCs which have a variance within 85% of the maximum (i.ethe variance of PC_1) are considered. This limiting of PCs is often necessary because it simplifies analysis of PCs whilst keeping the error introduced by information loss to a minimum.

After the PCs are calculated, the vectors are plotted on a PCA plot. The PCA plots are plots where the two axes are 2 different PCs. For example if only the first

4 ranked PCs are being analysed, a total of six plots (i.e 1 vs 2, 1 vs 3, 1 vs 4, 2 vs 3, 2 vs 4 and 3 vs 4) can be plotted. The method of plotting is to transform each feature vector, using the coefficients corresponding to the 2 PCs being plotted, into a set of 2D coordinates. A point is then plotted at these coordinates. It can be observed from PCA plots that data points corresponding to the same class can cluster. This is desirable because it shows that the variance within the data strongly describes class membership. If the data were noisy, the clustering would be less strong, and data that contained large amounts of noise may result in PCA plots that show no obvious clustering at all. Also the clustering may reflect other information, for example it would be possible for the clusterings to reflect the elapsed time of measurement from the start of the data gathering experiment. Such clusterings indicate significant variance within the data due to factors such a sensor drift or ambient temperature change.

4.3.1 Analysis Of Culture Growth Phase

The feature-set type which gave the best result when used with an MLP to classify culture growth phase was also used with PCA. This was in order for the information content to be visualised so that greater insight to the nature of the classification problem could be gained. The feature-set employed used the Absolute Final Output feature model with auto-scaling (*afa*). Data from experiments 1, 2, 3 and 4 were combined to form one large feature-set. This was the feature-set that was subject to PCA.

Table 4.17 shows the results of PCA, the % of the variance described by a PC to the total variance within the feature-set is shown. Those PCs which describe a comparatively large amount of variance, also correspond to the variance which most affects classification. Since there were two features per gas sensor, each feature vector had 12 components, there were therefore 12 PCs in total. It can be observed that the first 5 PCs describe 98.95% of the total variance of the feature-set.

Figure 4.5 is a PCA plot of the first 2 ranked PCs. The target classes are indicated by different plot colours (see caption for colour key). It can be observed from this plot that there was significant (meaning obvious to the eye) clustering amongst points of the same class. This indicates that there was significant variance within the feature-set

Table 4.17: Table showing the results of the application of PCA to the combined feature-set (*afa*) of experiment 1, 2, 3 and 4; by ranking the PCs in order of the % of total variance and % accumulated variance.

PC No.	% of Total	% Accumulated	PC No.	% of Total	% Accumulated
1	38.29	38.29	7	0.31	99.66
2	28.77	67.06	8	0.15	99.81
3	23.71	90.77	9	0.08	99.89
4	5.75	96.52	10	0.06	99.95
5	2.43	98.95	11	0.04	99.99
6	0.40	99.35	12	0.01	100.00

that is related to culture growth phase. Areas of overlap between clusters highlight potential sources of classification error. An ideal PCA plot would show clusters which have no overlapping regions.

Also from figure 4.5, it can be seen that there were more than one clusters for each class, for example the Lag phase class showed three clusters (one across the top and two more circular ones beneath), this was caused by variance between experiments. The two Static phase clusters at the top of the plot appear to merge into lines of Log phase, these lines were the result of sensor drift, the earlier measurements occurred on the left side and moved, with time, to the right side.

4.3.2 Analysis Of Bacteria Type

In common with the PCA performed for culture growth phase, PCA was also used with the feature-set type which facilitated the best results when used with a MLP in order to classify bacteria type. The feature-set employed used the Minimum Output feature model with sensor normalisation (*mns*). Data from experiments 1, 2, 3 and 4 were combined to form one large feature-set, which was subsequently used for PCA.

Table 4.18 shows the results of PCA, the ratio of the variance described by a PC to the total variance within the feature-set is shown. Similar to the feature-set used

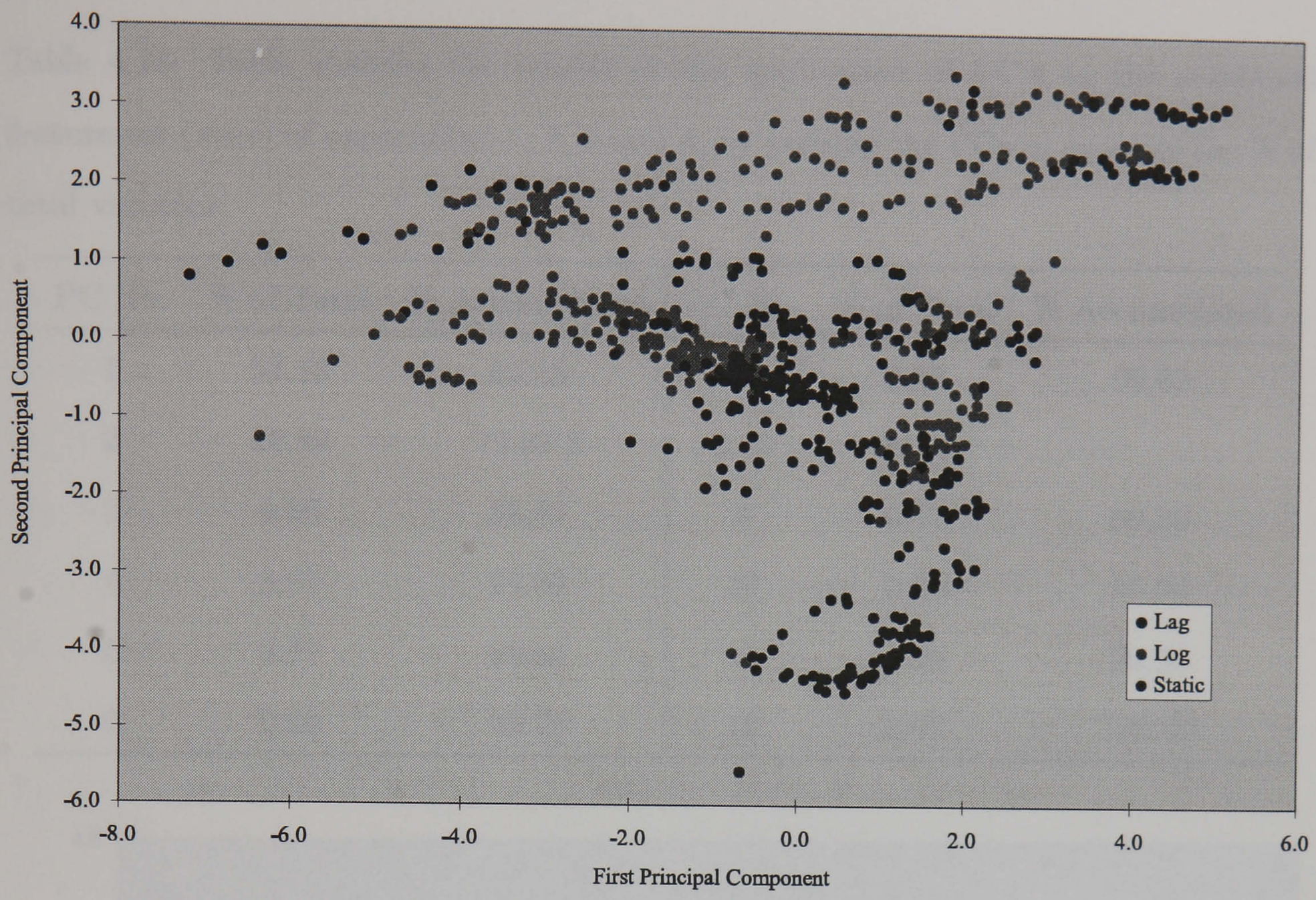


Figure 4.5: PCA plot of the first 2 ranked PCs of the combined feature-set (*afa*) of experiment 1, 2, 3 and 4. The 'target' data class is culture growth phase and is indicated by the plot colour, blue = lag, magenta = log and green = static.

for PCA analysis of culture growth phase, each feature vector had 12 components and therefore there were 12 PCs in total. It can be observed that the first 5 PCs describe 98.61% of the total variance of the feature-set.

Figure 4.6 is a PCA plot of the first 2 ranked PCs. The target classes are indicated by different plot colours (see caption for colour key). Again, in this PCA plot several clusterings can be observed. There are two clusters for *Escherichia coli*, corresponding to the two experiments which were performed with this bacteria type (i.e. experiment 1 and 2). There was one large cluster with many smaller clusters for *Staphylococcus aureus*, the large cluster corresponded to experiment 3 and the smaller clusters corresponded to experiment 4. The long lines indicate sensor drift, thus there was less sensor drift during experiments 1 and 2, than there was for experiments 3 and 4. There were areas of overlap in the central region of the plot, although most clusters were not overlapping. In general, it would be expected that feature-sets exhibiting less overlap

Table 4.18: Table showing the results of the application of PCA to the combined feature-set (*mns*) of experiment 1, 2, 3 and 4; by ranking the PCs in order of the % of total variance.

PC No.	% of Total	% Accumulated	PC No.	% of Total	% Accumulated
1	52.18	52.18	7	0.33	99.62
2	26.89	79.07 8	0.16	99.78	
3	9.80	88.87	9	0.11	99.89
4	5.97	94.84	10	0.05	99.94
5	3.77	98.61	11	0.04	99.98
6	0.68	99.29	12	0.02	100.00

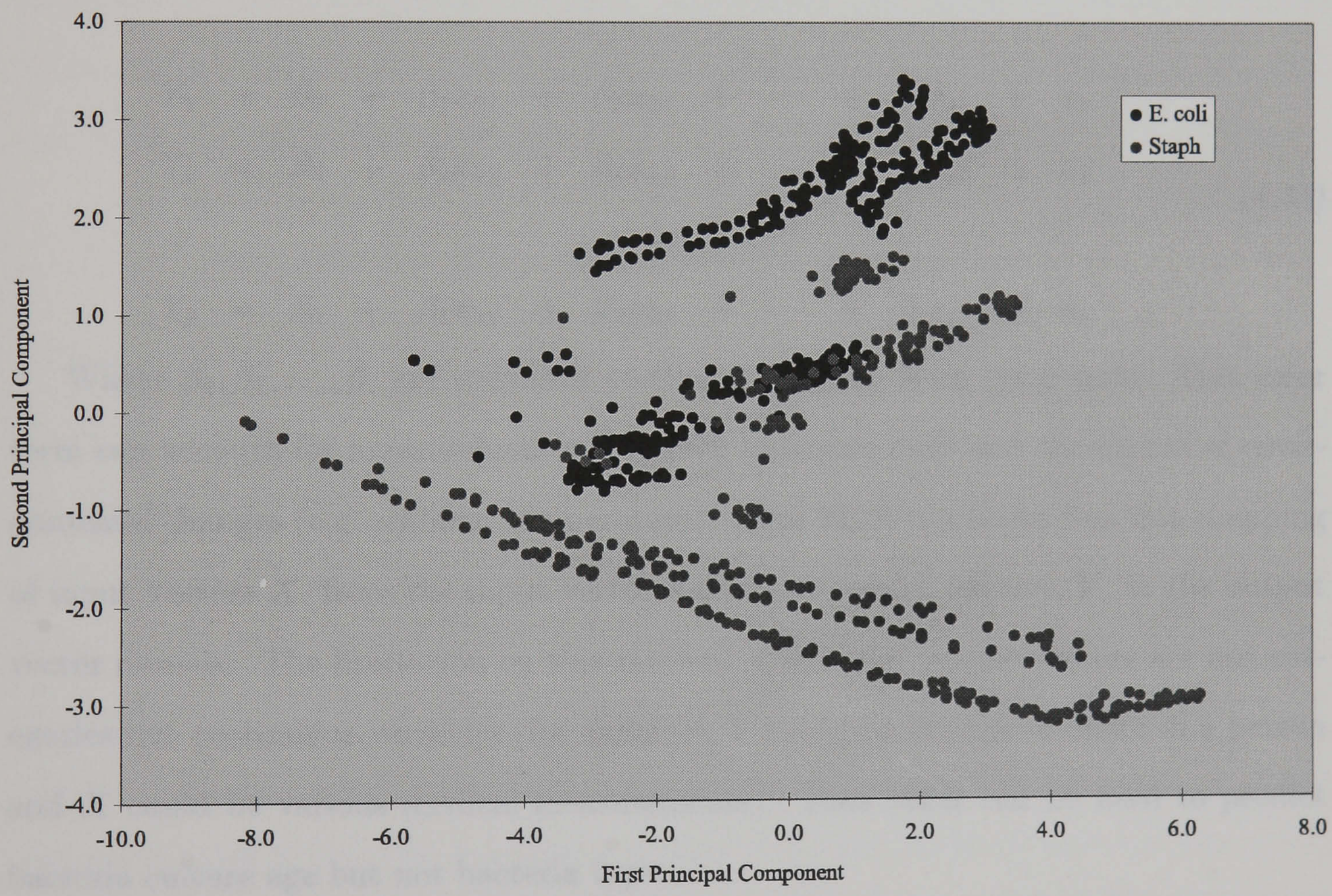


Figure 4.6: PCA plot of the first 2 ranked PCs of the combined feature-set (*mns*) of experiment 1, 2, 3 and 4. The 'target' data class is bacteria type and is indicated by the plot colour, blue = *Escherichia coli* and magenta = *Staphylococcus aureus*.

would also give good classification performance.

4.4 Multi-Variate Linear Regression

One of the most common 'classical' statistical techniques employed for classification problems with multi-variate data-sets is multi-variate linear regression (MLR). One (or more) dependent variables (i.e. class membership) are predicted from a set of independent variables (i.e. gas sensor features and possibly non-gas sensor features). This subject is well documented and many books have been published which describe MLR [113, 114].

A brief description of MLR is given here. First consider a single dependent variable, Y_j, that is predicted using a linear regression model:

$$\begin{aligned}
Y_1 &= \beta_0 + \beta_1 x_{11} + \beta_2 x_{21} + \cdots + \beta_i x_{i1} + \varepsilon_1 \\
Y_2 &= \beta_0 + \beta_1 x_{12} + \beta_2 x_{22} + \cdots + \beta_i x_{i2} + \varepsilon_2 \\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
Y_j &= \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \beta_i x_{ij} + \varepsilon_j
\end{aligned} \tag{4.34}$$

Where $\beta_0, \beta_1, \ldots, \beta_i$ are unknown coefficients and ε_j is an error term. This error term can account for measurement errors such as sensor drift and the effects of environmental changes (e.g. ambient temperature). Thus MLR is a linear function mapping of input vectors X, from the input vector domain to output vectors, Y, in the output vector domain. The limitation on this method is that the output vectors are not categories but continuous variables (for example, Y could be the age of death of a person and X could be various medical measurements). Thus MLR can be used to predict bacteria culture age but not bacteria type.

Derivation of the coefficients is performed using the least squares method. The best fitting coefficients produce a prediction equation where the squared differences between the target and predicted outputs (SSE) are minimised. A detailed description of this process is not given here, but the reader is referred to many publications [60, 113, 114] which contain such descriptions. The MLR model can be scored using the determination coefficient, r^2, which indicates how much of the variation of the dependent variable is

Table 4.19: Regression coefficients derived for prediction of culture growth phase.

Coefficient	Value	Coefficient	Value
β_0	2.709580	β_7	-0.26833
β_1	0.341223	β_8	-1.23841
β_2	-0.03336	β_9	-0.45654
β_3	0.946706	β_{10}	0.001384
β_4	0.520986	β_{11}	-0.20216
β_5	-0.46379	β_{12}	-0.00924
β_6	0.957773		

described by the independent variables. The higher this figure the better the MLR model. Also the correlation coefficient, r, is often quoted which is simply the square root of the determination coefficient.

4.4.1 Prediction of Bacteria Culture Growth Phase

MLR was used to predict culture growth phase. The feature set which yielded the best result when used with an MLP was also used with MLR, this gave some indication of the relative performance of MLPs compared to other, more straightforward methods, in this case MLR. The feature-set employed used the Absolute Final Output feature model with sensor normalisation. Data from experiments 2 and 4 were used to calculate the coefficients, and the data from experiments 1 and 3 were used to test performance.

Encoding the dependent variable, i.e. growth phase, was straightforward; giving the Lag phase a value of 1, the Log phase a value of 2 and the Static phase a value of 3, thus Y had one component which progressed from 1 to 3 (bacteria type could not be ordered in this manner, it being categorical variable).

Table 4.19 lists the coefficients that were derived from the training feature-set. Since the vectors within the feature-sets had 12 components (2 features per gas sensor), there were a total of 13 coefficients.

Once the coefficients were derived, they were used to predict the culture growth

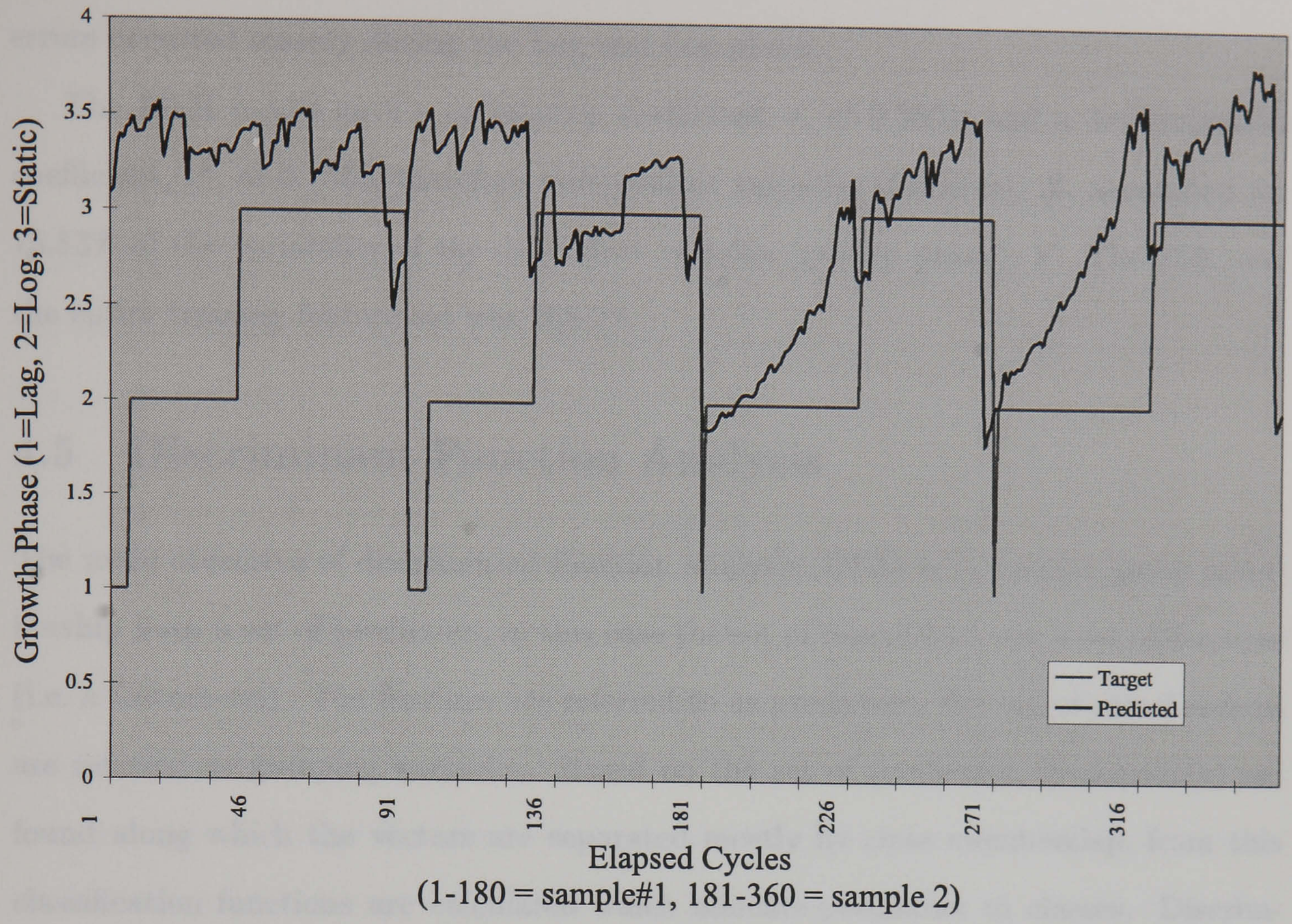


Figure 4.7: Performance of MLP when applied to culture growth phase prediction. The target output is shown along with the actual output. It can be noted that the data had 360 vectors which consisted of 4×90 feature-sets, each feature-set corresponded to an individual culture.

phase of a feature-set that was not used in training, i.e. the testing feature-set. The output from the MLR model was rounded to the nearest integer, if the output was the same as the target output then the answer was deemed to be correct, if they were unequal, the output was deemed to be incorrect. The plot shown in figure 4.7 shows both the target and actual outputs for the MLR model. The output was correct for 56.39% (203 correct outputs for a total of 360 vectors), this does not compare favourably with the figure for the MLP, which was 80.28%.

From figure 4.7, it can be observed that the output from the MLR model was more accurate in the later half of the plot, this corresponds to features from experiment 3 (*Staphylococcus aureus*), where the errors mainly occur around the boundaries between phases. The first half corresponds to features from experiment 1 (*Escherichia coli*), the

errors occurred mainly during the Lag and Log phase.

The MLR model gave a correlation coefficient, r, of 0.8691; and a determination coefficient, r^2, of 0.7553; therefore independent variables (features), X, accounted for 75.53% of the variability of the dependent variable (growth phase), Y. The SSE over the entire training feature-set was 268.27.

4.5 Discriminant Function Analysis

The main objective of discriminant function analysis (DFA) is to predict group membership from a set of predictors, in this case the set of descriptors was a set of features (i.e. a feature-set). The features are referred to as predictors, the target memberships are referred as grouping variables. Based on the set of predictors, dimension(s) are found along which the vectors are separated mostly by class membership, from this classification functions are calculated which allocate predictors to classes. Discriminant functions (DFs) are not to be confused with classification functions (CFs), the former are used to calculate the later. The first DF is found which best performs class separation, then the second DF is found which best performs class separation but is orthogonal to the first DF. This process is repeated, with each new DF being orthogonal to the previous ones, the total number of DFs is equal the the number of degrees of freedom of class membership, i.e. the number of dimensions of the feature vectors. Often, only the first few DFs are reliable enough to be used. DFA is parametric, the predictors are assumed to be randomly sampled from a parent population, and that the predictors are normally distributed. DFA is robust to some types of non-conformity, such as skewness, but is not robust to other types, such as outliers.

The following text is a basic description of the method for DFA, for a more detailed explanation several publication exist [113, 114]. Initially the data is screened for outliers, if any are found they are either removed or limited. The data are then tested for normality, if serious non-conformity is discovered, the data are transformed. The variance within the set of predictors is attributed to either inter-class variance and intra-class variance. This is done by deriving cross-product matrices:

$$M_{total} = M_{inter} + M_{intra} \tag{4.35}$$

The total cross-products matrix, M_{total}, is partitioned into the matrix containing the cross products associated with inter-class variation, M_{inter}, and the matrix containing the cross products associated with intra-class variation, M_{intra}. From this the Wilks' Lambda is calculated using their determinants:

$$\Lambda = \frac{|M_{intra}|}{|M_{inter} + M_{intra}|} \tag{4.36}$$

The parameter, Λ, is a measure of the amount of variance that is not involved in classification, $|M_{intra}|$, divided by the total variance, $|M_{inter} + M_{intra}|$. The higher this parameter the more problematic classification becomes because the greater amount of variance is not attributed to class membership. From this the approximate F ratio is derived which is used to test the significance the variances, if the critical F is exceeded then the predictors can be used to distinguish between the classes[6]. The DFs are found such that the F ratio is maximised. DFs are similar in construction to regression functions (see equation 4.34), each predictor is weighted by a coefficient and summed:

$$DF_i = d_{i1}p_1 + d_{i2}p_2 + \cdots + d_{in}p_n \tag{4.37}$$

Where there are n predictors, p, and i DFs, therefore in DF coefficients. The coefficients are found in a similar manner to those for canonical variates (with the DFs corresponding to canonical variates), with it being a problem of canonical correlation similar to PCA. Basically the DFs are found which maximise the inter-class variance relative to the intra-class variance. The coefficients are auto-scaled so that, for all data, the mean of each DF is zero and the standard deviation is 1. Each class has associated with it, different means for each DF, the magnitude of which indicate the relevance of each DF. Thus the DFs are ranked in order of relevance.

Once the DFs have been computed, the CFs can be calculated, a CF has the form:

[6] In this test confidence levels, such as 95% or 99% can be specified.

$$CF_j = c_{j0} + c_{j1}p_1 + c_{j2}p_2 + \cdots + c_{jn}p_n \tag{4.38}$$

Where there are j CFs (j = no. of classes) and n CF coefficients, c for each CF. The coefficients are calculated from the means of the predictors , M_j, and the pooled intra-covariance matrix, W. The classification procedure is simple, each set of predictors, i.e. feature vector, is input to each CF, the vector is assigned to a class, the CF of which outputs the highest value. It can be seen that the CF is linear in nature, therefore classification degrades where non-linear relationships describe class membership.

A useful tool in DFA is a discriminant function plot, this is similar to a PCA plot except the axis correspond to the highest ranked DFs instead of PCs. The performance of DFA can be indicated by the separation of classes observed in such plots.

There are many descriptive statistical measures used to predict the reliability of a set of DFs, two of which, Mahalanobis distance and Fisher's F distance, were employed. The Mahalanobis distance, D^2, is based on the distance between pairs of class centroids, which is then generalisable to distances over multiple pairs of classes. The larger the measure, the higher the classification performance. Similarly, Fisher's F distance is a measure of class separation based on centroid distances, a large result is desirable.

4.5.1 Classification of Culture Growth Phase

DFA was used to classify culture growth phase for the same feature-sets as those used for PCA and MLR (i.e. Absolute Final Output gas sensor feature model, auto-scaled). The DFs and CFs were calculated using data from experiments 2 and 4, and were tested using data from experiments 1 and 3. Table 4.20 shows the results of DFA, the coefficients of the CFs are listed.

Table 4.21 shows the corresponding Mahalanobis and Fisher's distance measurements. The value for Wilk's Lambda was 0.1271. Reclassification of the training feature set was 90.56% correct (326 correct out of 360 total), testing using the test feature-set yielded a classification performance of 67.50% correct (243 correct out of 360 total). This result was worse than classification using a MLP (80.28%), but better than MLR (56.39%).

Table 4.20: Classification function coefficients calculated using discriminant function analysis for culture growth phase (using *afa* feature-sets).

	Culture Growth Phase		
Coefficient	Lag	Log	Static
c_0	-7.60	-1.50	-3.73
c_1	3.16	-1.88	2.63
c_2	-5.69	0.37	-0.16
c_3	-18.59	-2.87	5.69
c_4	25.90	-15.64	21.88
c_5	20.51	-3.16	3.31
c_6	-29.19	-0.52	2.89
c_7	-1.03	4.73	-7.11
c_8	4.26	12.28	-18.95
c_9	-14.70	1.40	-1.07
c_{10}	16.19	0.04	-1.23
c_{11}	3.85	1.54	-2.61
c_{12}	-10.32	1.76	-1.93

Figure 4.8 shows a plot of the test feature-set against the first two ranked discriminant functions. The target class membership is indicated by plot colour (see figure caption). From this plot, clusterings can be observed between points corresponding to the same target class, and therefore class separation was possible. It can also be noted that there were some outlying points corresponding to the Lag phase near the bottom of the plot, it is likely that these were a source of error.

4.5.2 Classification of Bacteria Type

DFA was used to classify bacteria type for the same feature-sets as those used for PCA (i.e. Minimum Output gas sensor feature model with sensor normalisation). The DFs and CFs were calculated using data from experiments 1 and 3 and were tested

Table 4.21: Mahalanobis, D^2, And Fisher's F distances between class centroids calculated using discriminant function analysis for culture growth phase (using *afa* feature-sets).

		Lag	Log	Static
Lag	D^2	0	13.93	31.31
	F	0	128.11	183.90
Log	D^2	13.93	0	19.88
	F	128.11	0	111.39
Static	D^2	31.04	19.88	0
	F	183.90	111.39	0

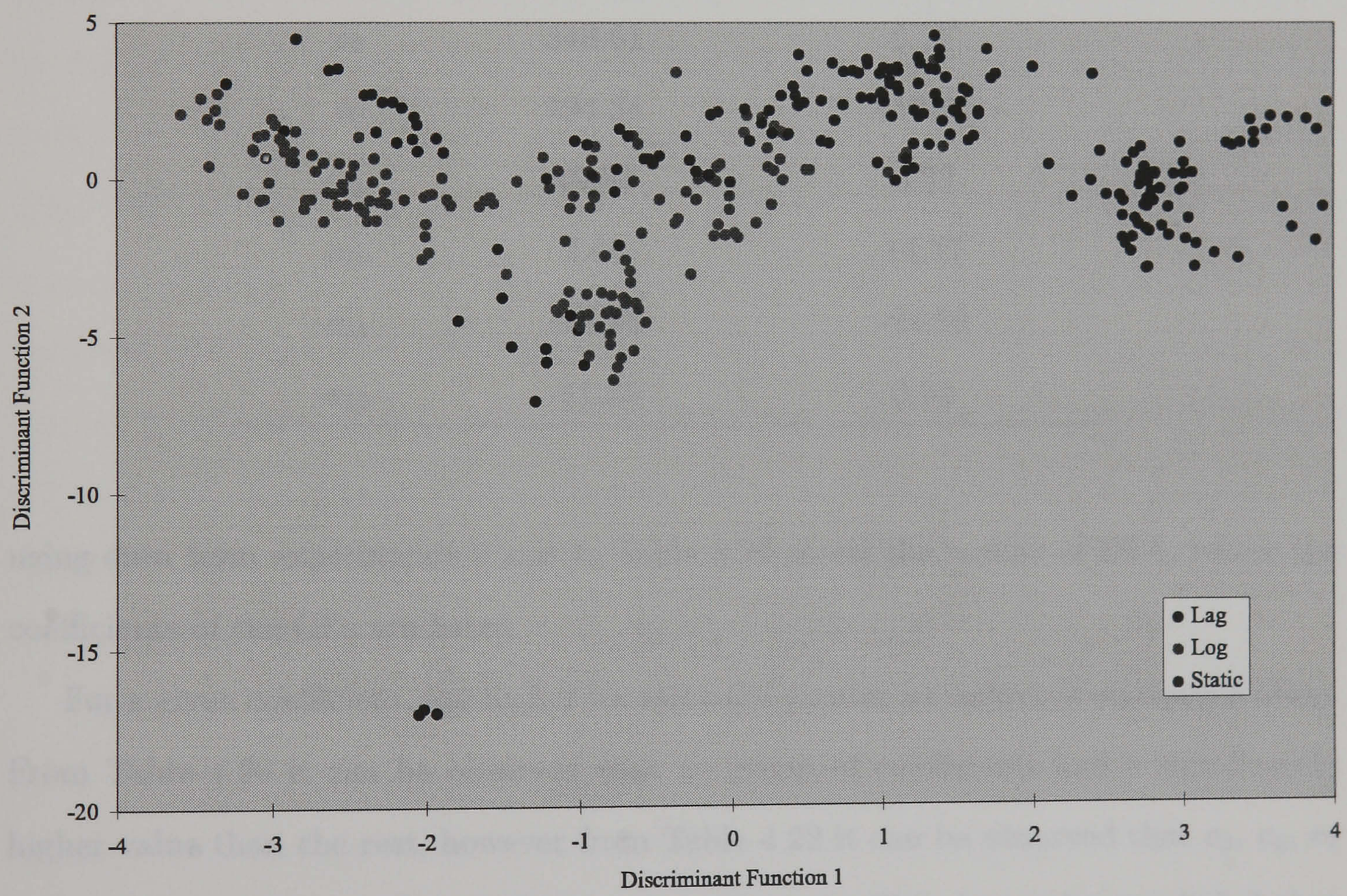


Figure 4.8: Results of discriminant function analysis of culture growth phase (using *afa* feature-sets). Colour key: blue = lag phase, magenta = log phase and green = static phase.

Table 4.22: Classification function coefficients calculated using discriminant function analysis for bacteria type (using *mns* feature-sets).

	Bacteria Type	
Coefficient	*Escherichia coli*	*Staphylococcus aureus*
c_0	-36.56	-14.54
c_1	-6.76	-7.68
c_2	11.75	-6.38
c_3	-318.89	-124.18
c_4	-418.06	-57.92
c_5	40.59	156.64
c_6	-19.78	-50.02
c_7	348.61	9.77
c_8	194.34	117.75
c_9	49.98	6.32
c_{10}	2.80	14.77
c_{11}	-72.16	-24.79
c_{12}	-11.35	0.01

using data from experiments 2 and 4. Table 4.22 shows the results of DFA, where the coefficients of the CFs are listed.

For a given coefficient, the higher its value the greater its influence on classification. From Table 4.20 it can be observed that no group of coefficients had a significantly higher value than the rest, however from Table 4.22 it can be observed that c_3, c_4, c_7 and c_8 have values significantly higher than the rest. Thus is can be concluded that these coefficients are the most significant[7]. Coefficients c_3 and c_4 correspond to gas sensor 2 (FIS P.10.1 sensitised to hydrocarbons and others), and that coefficients c_7 and c_8 correspond to gas sensor 4 (FIS P.A.2 sensitised to polar compounds). These

[7]Similarly coefficients with small values are not significant for classification and may be a source of noise.

Table 4.23: Mahalanobis, D^2, and Fisher's F distances between class centroids calculated using discriminant function analysis for bacteria type (using *mns* feature-sets).

		Escherichia coli	*Staphylococcus aureus*
Escherichia coli	D^2	0	41.37
	F	0	300.76
Staphylococcus aureus	D^2	41.37	0
	F	300.76	0

sensors seem to be the most significant ones for DFA.

Table 4.23 shows the corresponding Mahalanobis and Fisher's distance measurements. The value for Wilk's Lambda was 0.0877, this is less than the figure calculated for culture growth phase. This indicates that there was more variance corresponding to bacteria type (for feature-sets *mns*) than for culture growth phase (for feature-sets *afa*). This theory is further strengthened because the distance measures for bacteria type are considerably greater than those for culture growth phase. Reclassification of the training feature set was 98.89% correct (356 correct out of 360 total), testing using the test feature-set yielded a classification performance of 65.83% correct (237 correct out of 360 total). This result was considerably worse than classification using a MLP (96.11%). This suggests that there were significant sources of error that originated from variances between different data gathering experiments.

Figure 4.9 shows a plot of the test feature-set against the first two ranked discriminant functions. The target class membership is indicated by plot colour (see figure caption). The clusters observable from this plot show well defined class membership clusters. The cluster corresponding to *Escherichia coli* is smaller than that for *Staphylococcus aureus* indicating that the intra-class variance for the later was greater than that for the former. Given such well defined clusters and the poor classification performance it can be concluded that although the set of DFs calculated during the DFA separated classes well, the subsequent CFs were not optimal.

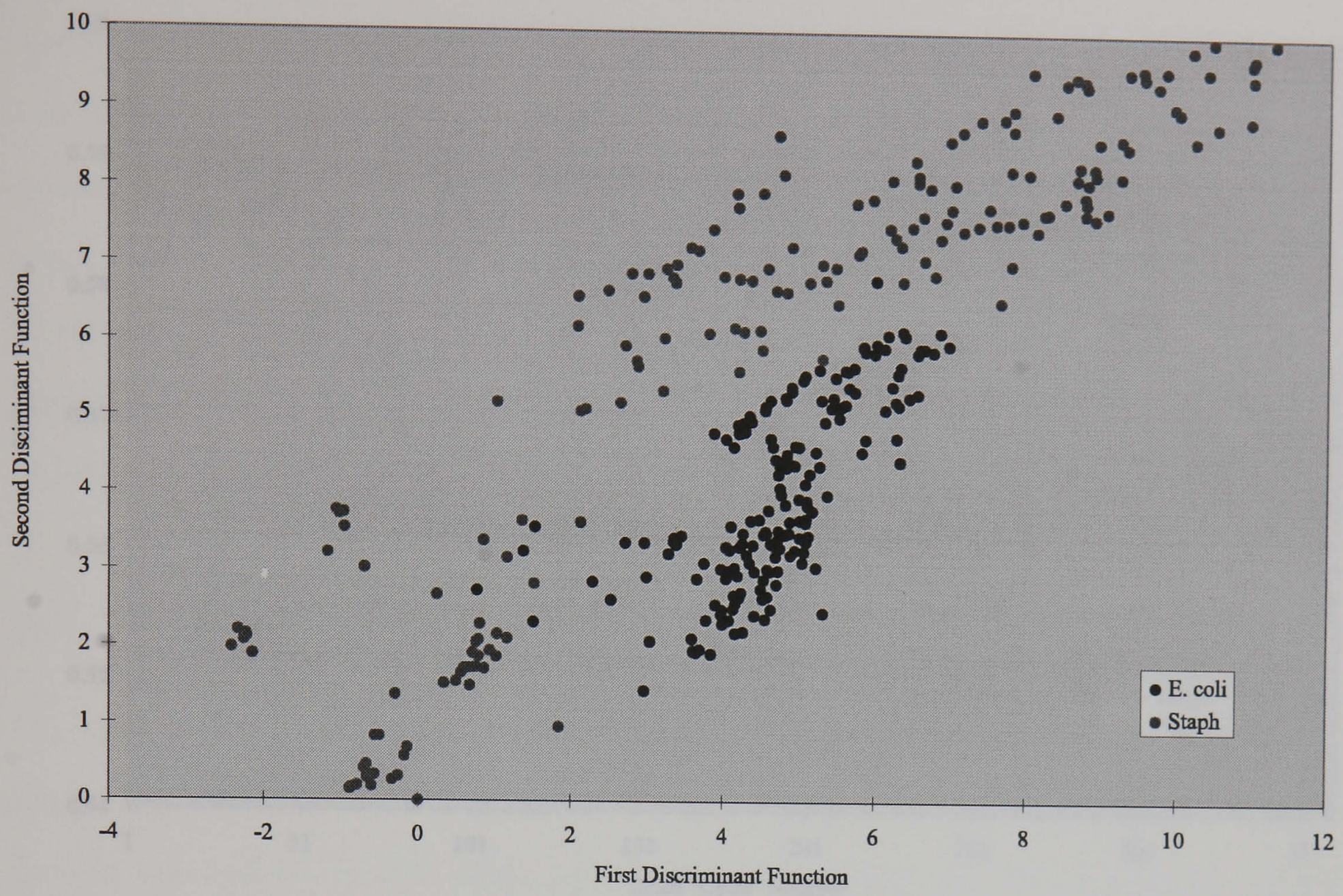


Figure 4.9: Results of discriminant function analysis of bacteria type (using *mns* feature-sets).

4.6 Temperature Dependence of Baseline

The baseline response from a gas sensor is defined as the final output before the odour of interest is measured. Referring to the plot in figure 4.1, the baseline response can be specified as the V_{ref}^{final} parameter. This parameter is closely related to sensor drift because the reference odour was constant. It has already been established that air temperature was prone to drift (see figure 3.8). It was postulated that if the air temperature could be predicted from the baseline of the gas sensors, then a relationship existed between the two. In chapter 3 (section 3.6) correlation analysis revealed a significant level of correlation between gas sensor output and air temperature. A MLP ANN trained using BP with momentum, was employed as universal function approximator. The network had 6 inputs, one for each of the gas sensors, the baseline measurement was input. The network had 10 hidden nodes and 1 output node. The training parameter values were the same as those used previously for bacteria type and culture

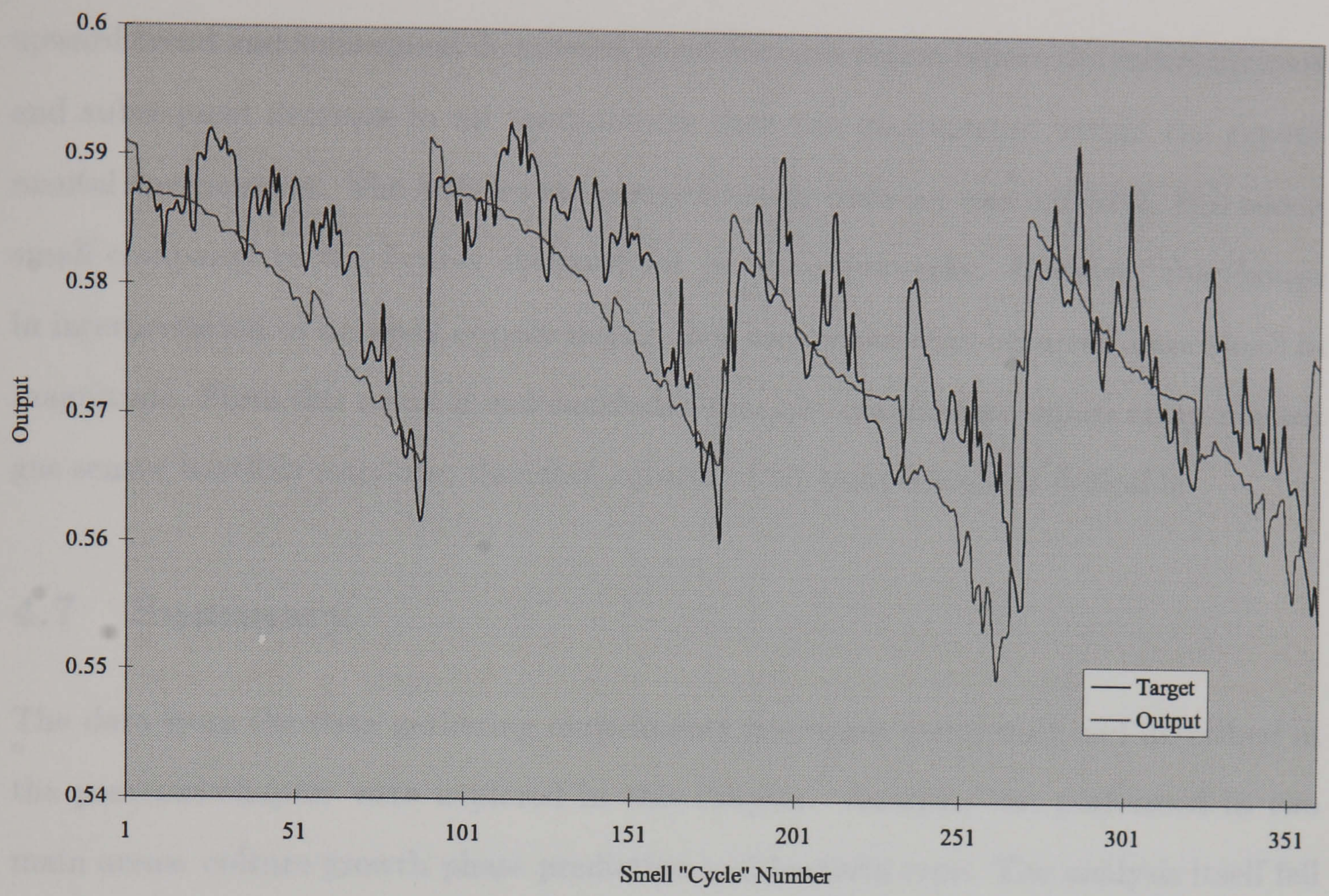


Figure 4.10: Air temperature prediction using a MLP, the plot shows both the target output and actual output.

growth phase classification. Also the procedure for modeling and testing the network were the same as those previously employed. The MLP was trained using data from experiments 1 and 3 and tested using data from experiments 2 and 4. The interpretation of results was different to that previously adopted because the output did not represent a categorial variable (i.e. class membership), but a continuous variable. For each 'smell' cycle, there was an input vector, which was the baseline response, and an output, which was the overage temperature sensor reading over the cycle. The target output for the MLP was the average temperature sensor reading. Figure 4.10 shows a plot where, for the training data, the target output is shown along with the actual output.

It can be seen from this plot that the MLP was able to predict air temperature with reasonable accuracy. Data from each experiment contained data from 2 vessels (the cultures in vessels #2 and #3), therefore the plot clearly shows 4 distinct regions, each region corresponds to one culture in one experiment. The general initial

upward trend and subsequent downward trend for each region reflect the initial increase and subsequent decrease in air temperature that was encountered within the experimental environment. The SSE for the entire training data-set was 0.015205, this seems small compared to the figures obtained for previous networks. However, the change in interpretation of network output meant that any errors that occurred were small in magnitude. From this result it was concluded that air temperature significantly effected gas sensor baseline response, therefore control of air temperature is desirable.

4.7 Summary

The data from the data gathering experiments previously performed and described in the previous chapter were explored in this chapter. Analysis was performed in two main areas: culture growth phase prediction and bacteria type. The analysis itself fell into two major phases: data pre-processing and classification. Pre-processing involved feature extraction and normalisation. Feature extraction was simply a method to extract 'features' from the data where the amount of data was reduced by the amount of information (related to the classification task) was retained. Normalisation re-scaled and transformed the features in order to make classifier training easier. The different types of classifier tried were: multivariate linear regression, discriminant function analysis and multiple layer perceptron. Each classifier was trained and tested on data from different data gathering experiments, thus the need to classifier performance estimating techniques, such as cross-validation, was removed.

The bar chart in Figure 4.11 shows the relative performance of each type of classifier for growth phase predication and bacteria type classification for the number of correct classification. From this diagram it can be seen that the MLP performed best in both cases. The results were very promising with up to 96.11% of bacteria types and 80.28% of culture growth phases being correctly classified. It was found that different combinations of features and normalisation performed best in each case. Principal component analysis was used to analyse the variance in the data and to identify sensor drift.

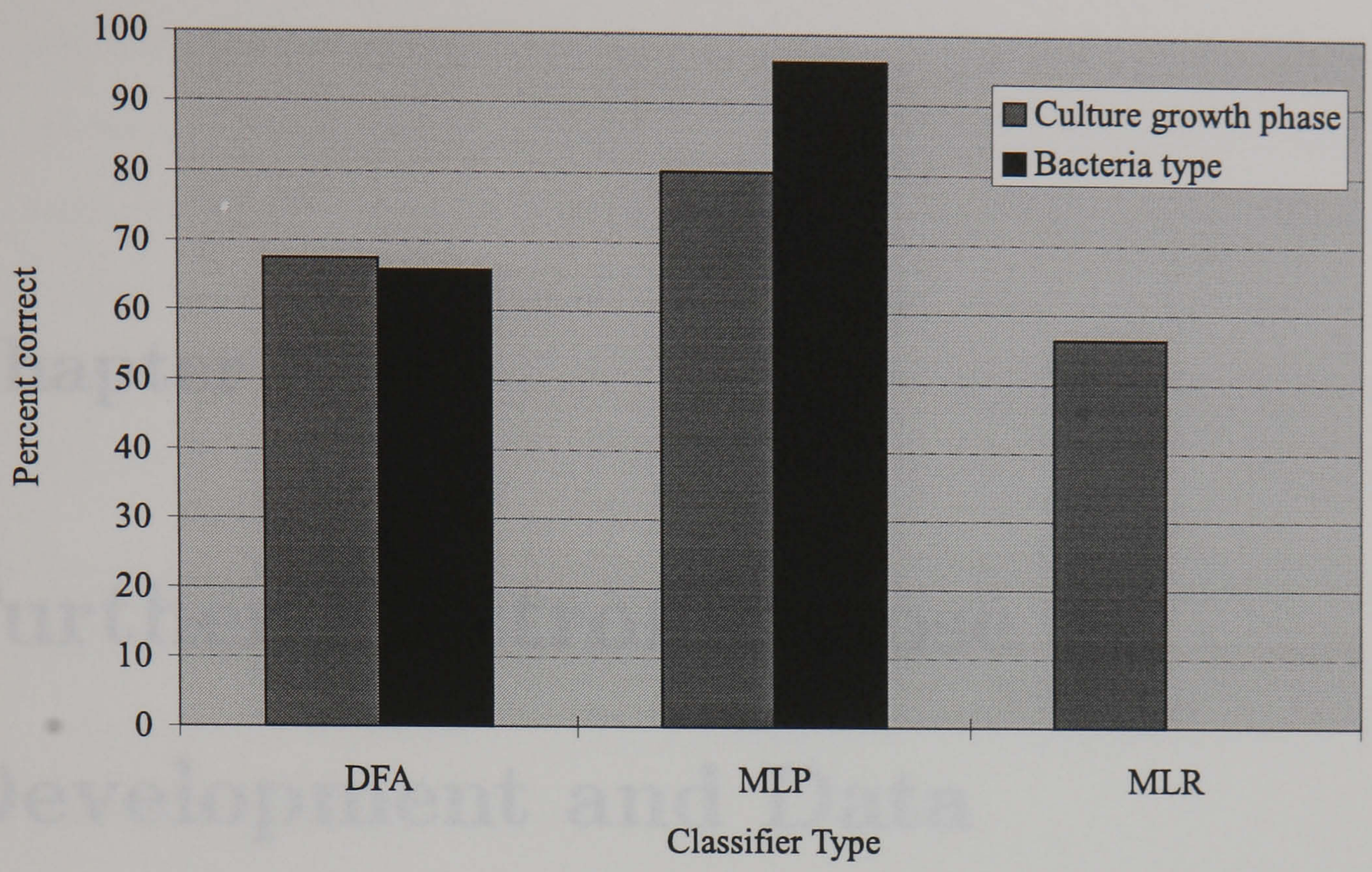


Figure 4.11: Bar chart showing the relative performance of the number of correct classifications for each classification technique for culture growth phase and bacteria type, using the best single performance in each case.

Finally in this chapter, the relationship between ambient temperature and baseline resistance was investigated further. It was shown, that if ambient air temperature could be controlled, then a source of unwanted variance (i.e. noise) could be removed and thus classification performance increased significantly.

Chapter 5

Further Electronic Nose Development and Data Collection Experiments

Having completed the first set of experiments and analysed the data, I decided to modify the Electronic Nose further and perform more data collection experiments. The instrument modifications took the form of a new sensor chamber within the Electronic Nose. The unit was moved back to the SRL in the Department of Engineering because of the need to test and re-calibrate the modified instrument.

It was felt that the plastic rectangular box used for the sensor chamber was not ideal because the dead-space was large, gas flow was not uniform across each sensor and the material could show a matrix effect. The fact that individual sensor responses were dependent on the position of the gas sensor within the gas chamber is clearly problematic for instrument reproduceability (this is discussed in section 5.2). Also it was observed from the gas sensor responses (see figure 3.8) that variations in gas and ambient temperatures were a major potential source of sensor noise (i.e. unwanted variance). For these reasons it was decided to modify the instrument to introduce the following features:

- Temperature control. After inspecting the problem, it was decided to set a spe-

cification for the temperature of the gas reacting with the gas sensors to ±0.1°C of the desired (target) temperature, the range of usable target temperatures were specified as +5°C to +25°C above ambient air temperature (which itself usually varied from 15°C to 25°C in out laboratory). Although the odour sources were kept at a near constant temperature, by use of the DRI-BLOC heater, the incoming gas to the sensor chamber fluctuated by as much as 10°C (see figure 3.8).

- Improved gas flow within the gas sensor chamber. The new chamber has been designed to provide a uniform flow of gas across each gas sensor. The design also allows the incorporation of a resistive heating element that forms part of a gas temperature control sub-system.

- Faster odour sampling speed. The dead-volume of the system should be maintained at as low level as is possible. The internal volume of the chamber is minimal in order to reduce gas mixing times (which increases overall response speed). Moreover the pipe-work and valves have a lower dead-volume than previously employed.

- Improved reliability. The original FOX 2000 unit had some reliability problems such as the pump failures detailed in chapter 3, our new unit had been re-designed to improve reliability and repeatability[1].

The modifications are described in two sections to aid the reader, although in reality the modifications made one sub-system. The sub-system is first described without specific detail about the physical design of the new main gas sensor chamber. Second, the actual design of the new main gas sensor chamber is discussed.

A philosophy adopted during the design phase was that of ease of fitting. It was considered advantageous to be able to modify the original FOX 2000 in a simple fashion by removing the old sensor chamber sub-system and replacing it with the new sub-

[1]This point may initially seem obvious, however, previous data collection required an operator to be constantly present (and so able to rectify problems). The new automated system only requires an operator at the start and the end of a data collection session. The system therefore operates alone for a considerable length, any problems that occur will not be detected until the end.

system. This would enable other instruments to be modified by this simple replacement operation.

5.1 Design of Gas Temperature Control Sub-System

Figure 5.1 shows the modified FOX 2000 incorporating the new sub-system. This can be compared to figure 3.6 which shows the original FOX 2000. The shaded area highlights the new sub-system components.

The design of the temperature control sub-system is based on twin chambers. A pre-heater (and heat exchanger) chamber is placed in series with the input to the main chamber. A twin chamber system is better than a single chamber system because, given the temperature specifications, it allows the use of two relatively low power heaters rather than one much larger one. Because the chambers were heated, they were constructed from metal rather than plastic (used for the original chamber), providing a suitably low thermal resistance from the heater element to the gas within the chamber. The metal chosen was aluminium because it has a smaller density than other metals, it is easy to machine, and was readily available[2]. The pre-heater chamber was small (58mm length, 32mm width and 11mm depth) and had a mean specific heat capacity (VALUE) that allowed large incoming gas temperature fluctuations to be compensated for (see section 5.3). The pre-heater chamber consists of two halves, in one half the input and output pipes were fixed and the passageway for the gas was machined. The passageway was a series of folded bends (this is shown in figure 5.1) that enabled a large surface area between the gas and the heated chamber block, so heating of the gas was improved. The main sensor chamber had a larger mean heat capacity (37mm depth, 74mm diameter, excluding lid) than the pre-heater chamber, it also had a larger thermal time constant (see section 5.3). Because the pre-heater chamber significantly reduced the fluctuation of temperature of gas input to the main chamber (See Table 5.1 and Figure 5.5), accurate control of the gas temperature within the main chamber was possible. A second sensor chamber, the pre-sensor chamber, was placed between the

[2]The aluminium was coated with a thin film of PTFE to reduce gas sorption.

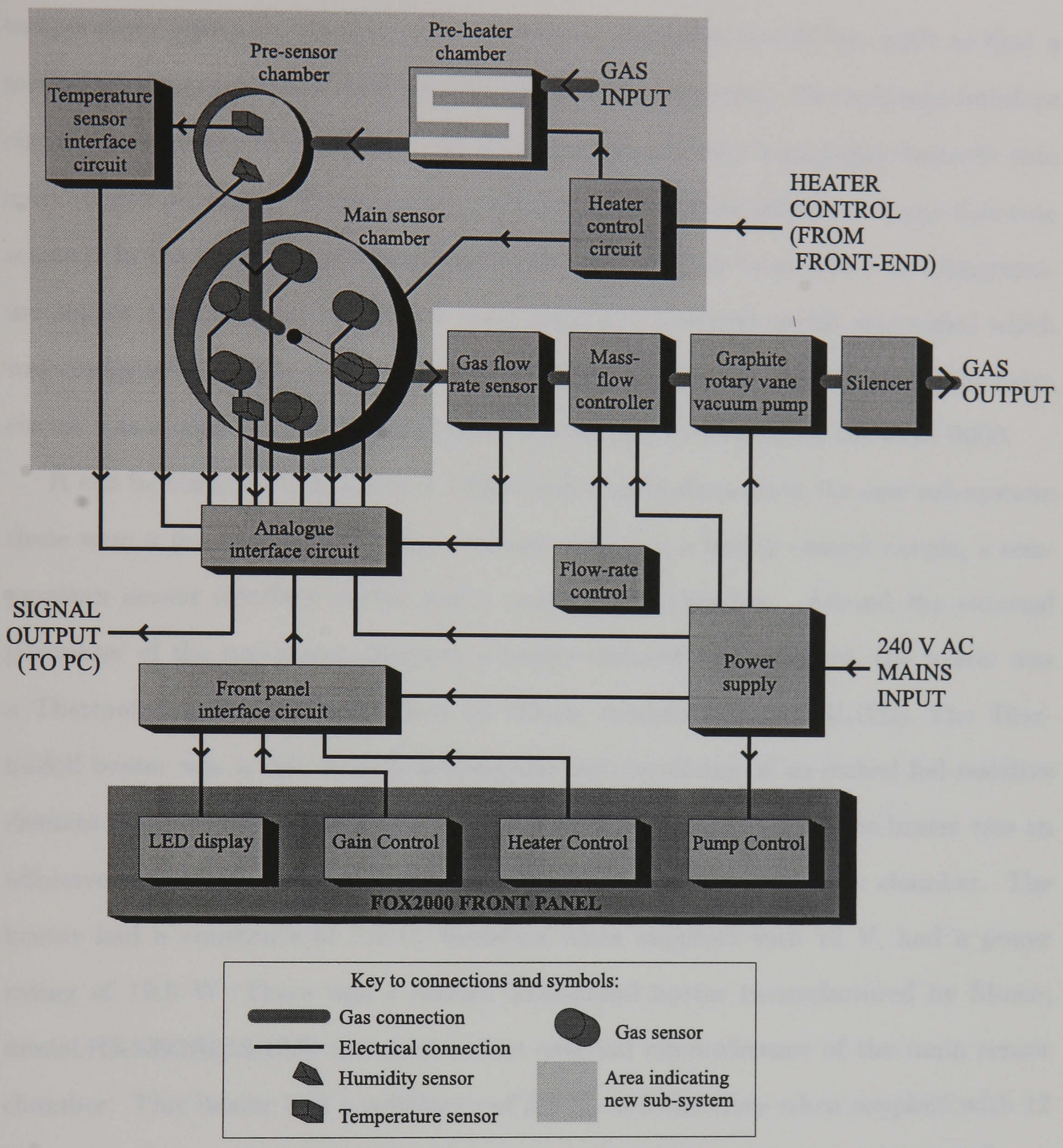


Figure 5.1: The Modified Electronic Nose showing the new main sensor chamber, pre-sensor chamber, pre-heater chamber, heater control circuit and the temperature sensor interface circuit.

main sensor chamber and the pre-heater chamber. The pre-sensor chamber contained a temperature sensor (LM35CZ) and a relative humidity sensor (MiniCap 2), the same types as used in the original sensor array, this allowed the temperature and humidity of the gas exiting from the pre-heater chamber to be measured. The main sensor chamber contained the 6 gas sensors with a second temperature sensor (of the same type as the

temperature sensor in the pre-sensor chamber). An extra circuit was built so that a second temperature sensor could be interfaced to the computer. The analogue interface circuit has capacity to interface to 12 devices, originally only 9 analogue channels were used (6 gas sensors + 1 temperature sensor + 1 humidity sensor + 1 gas flow-rate sensor). In the original instrument there was only circuitry to interface to 1 temperature sensor, the design of this circuit was copied and a second circuit was added which was connected to a spare analogue channel on the ADC in the LMP-16 card. The extra circuit was a copy of the original temperature sensor circuit within the FOX 2000.

It can be observed from figure 5.1 that were 5 main elements in the new sub-system; these were a pre-heater chamber, pre-sensor chamber, a heater control circuit, a temperature sensor interface circuit and a main sensor chamber. Around the external perimeter of the pre-heater chamber a heater element was attached, the heater was a Thermofoil heater (manufactured by Minco, model:HK5368R7.4L12E). The Thermofoil heater was a thin flexible heating element consisting of an etched foil resistive element laminated between layers of flexible insulation. One side of the heater was an adhesive strip, which was used to attach the heater to the pre-heater chamber. The heater had a resistance of 7.4 Ω, therefore when supplied with 12 V, had a power rating of 19.5 W. There was a second Thermofoil heater (manufactured by Minco, model:HK5393R8.9L12E) attached to the external circumference of the main sensor chamber. This heater had a resistance of 8.9 Ω, and therefore when supplied with 12 V, had a power rating of 16.2 W. Ultimate control of the heaters was via the PC and Figure 5.2 shows how the heaters were interfaced.

From Figure 5.2, it can be observed how the voltage supply to the heaters was controlled using a simple transistor switch design. The transistor used (2N3055) is an easily available high power, general purpose, *npn* transistor (maximum collector current rated at 10 A). The base current required to drive the base of the transistor was in the order of μA, as this was easily driven by the digital output of the LPM-16 I/O card, rated at up to 100 mA per line. The Front-end used 6 digital outputs for the 6 solenoid values, the LPM-16 I/O card had 8 digital output, so there were 2 unused. Via the Front-End Control Circuit, the base of the transistor switch was connected

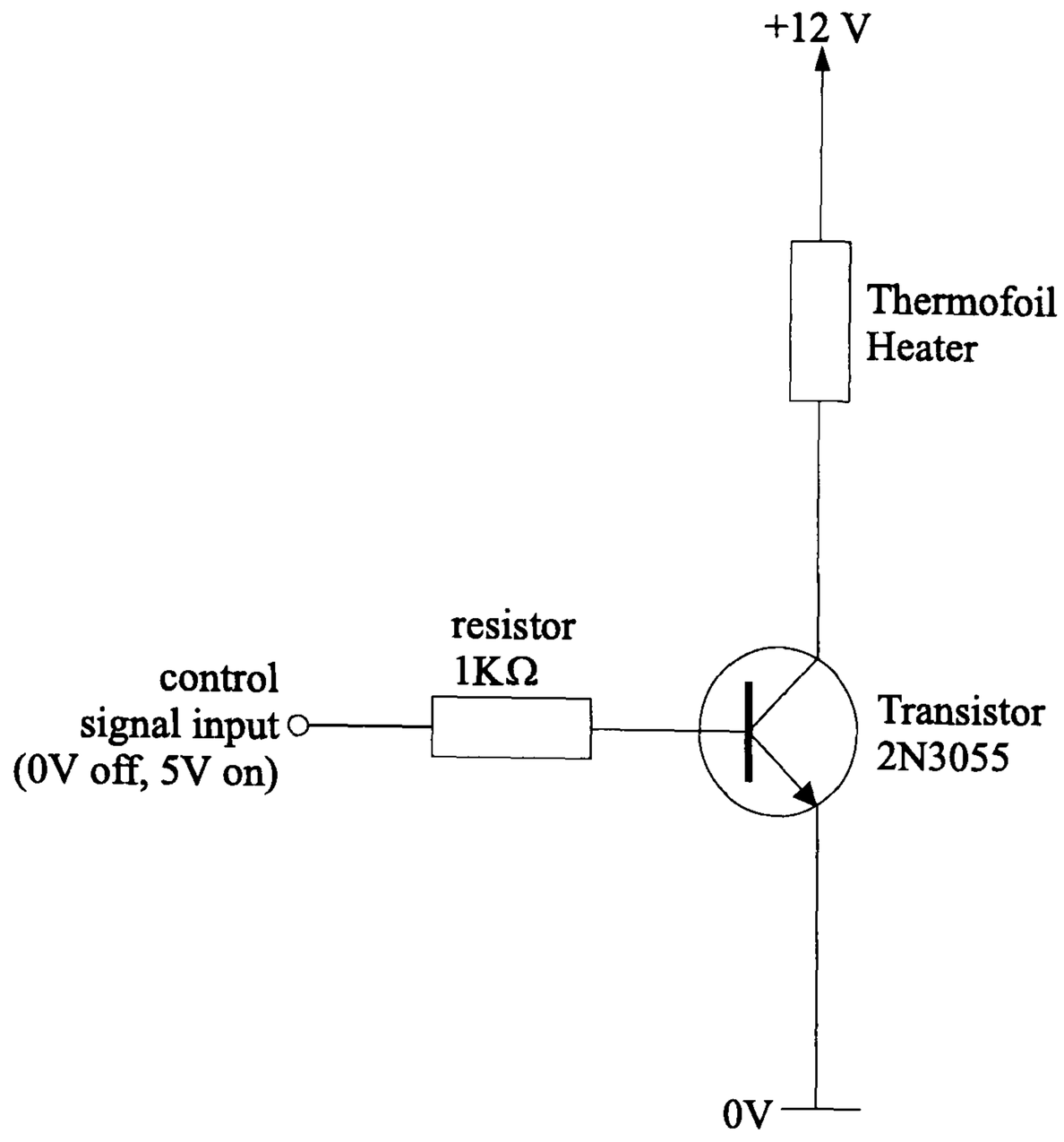


Figure 5.2: The circuit involved in controlling the heaters in the new sub-system, showing how the heaters were controlled from signals output from the PC.

to the unused digital output. When the PC output a high value, the corresponding transistor switch was on and so the its heater was turned on; when the PC output a low value, the corresponding transistor switch was off and so its heater was turned off. This method of heater control only allowed for the heaters to be either fully on or fully off, a proportional control would in theory have been better however in practice was not necessary[3]. Two external power supplies (basic 0-30 V bench-top models manufactured by Farnell, type E30/1) were used to power the heaters, when driven at 12 V, the pre-heater chamber heater element lead current was 1.82 A and the main sensor chamber heater element lead current was 1.35 A. The internal power supply of the FOX 2000 was not of sufficient capacity to drive the heaters.

[3]The LPM-16 I/O card had no analogue outputs and no free digital outputs, proportional control would not therefore have been possible from the computer using this I/O card. A separate, dedicated, circuit would have been possible; it was decided to try a simpler route first which turned out to meet the specifications.

5.1.1 Computerised Control of the Temperature Control System

The heaters were controlled from the PC. A program was written in the Labview language in order to allow intelligent control of the heaters. This program, called the Temperature Control Program, allowed the user to set a target temperature for both the pre-heater chamber and the main sensor chamber. This program is detailed in appendix A, a basic description is given here. It was decided to write a separate program instead of extending the Front-end Control Program because future projects may use the new sub-system but not the Front-end (or vice-versa). Separate programs allow separate implementation of the either the Front-end or the temperature control sub-system to be achieved easier. Development of the Temperature Control Program was problematic for several reasons:

- The specification of the PC computer was an Intel 80486DX33 CPU, which was run at a clock speed of 33 MHz. The available computing resources available to the Labview interpreter meant that running 2 Labview programs simultaneously pushed the limits of the computer.

- Because the Labview interpreter runs programs in a non-preemptive environment, it was possible for one Labview program to 'hog' the processor and so prevent the other from running. Essentially the programs would have to be 'compatible' with each other. Also the graphical environment was Microsoft Windows 3.1, which is also a non-preemptive environment so an independent program was able to halt the Labview interpreter.

- The Front-end Control Program changed the state of the digital output of the LPM-16 I/O card, if the Temperature Control Program did likewise, conflicts would occur. The LPM-16 digital output was controlled by writing a byte to a single location, hence it was impossible to change the state of one output line without affecting the other lines.

The first problem was solved by writing all programs in such a manner that they were as efficient as possible with processor resources. This meant, for example, that

floating point operations were kept at a minimum and graphical output was kept simple.

The second problem was solved by making sure no procedures in the Labview programs demanded exclusive processor usage for any significant length of time (more than a few milliseconds) and no other Windows programs were running simultaneously.

The third problem was the most difficult to overcome because both programs needed to be able to change the state of the LPM-16 I/O card digital outputs. The Front-end Control Program and the Temperature Control Program were adapted to communicate with each other using a Windows mechanism called DDE (dynamic data exchange). The Temperature control program acted as a server whereby the current status of the digital outputs were reported, the Front-end Control Program was the client, and read the status of the port and reported any change in output line status back to the server. Because of this no conflict occurred. The Temperature Control Program had to be started before the Front-end Control Program because the DDE server (the former) had to be operational before the DDE client (the latter) could start.

The actual method of temperature control was a simple 'bang-bang' controller[4], i.e. when the output from a temperature sensor rose above a pre-defined threshold the appropriate heater was switched on, and when the output fell below the threshold the appropriate heater was switched off. The value of the thresholds were determined by empirical study, where external thermometers were used to monitor gas temperature. This type of control is not ideal, for example a proportional controller would have performed well. However given the hardware and time restrictions, it was felt that the 'bang-bang' controller would suffice.

As well as setting the target temperature, the Temperature Control Program displayed a plot showing the output from the temperature sensors over the previous 10 minutes. Also the current status of the heaters was displayed (either on or off), the current mean and standard deviation for each temperature sensor was displayed, the current output from the temperature sensors was displayed and a histogram of the

[4]Originally pulse width modulation (PWM) was tried, however the implementation of a pulse width output software routine burdened the CPU too much and the PC internal time was not of a high enough frequency. The LPM-16 does not provide any timing circuitry suitable for implementing a PWM controller.

main sensor chamber temperature was displayed. The display, therefore, allowed at a glance the user to determine the current status of the temperature control system and its recent history.

After the software development was finished, there were two versions of the Front-end Control Program, the original version which did not use DDE and could be run in its own, and the modified version which used DDE and needed the Temperature Control Program to run.

Testing of the new software was performed during development using the Labview debugging tools. The new sub-system was connected to the LPM-16 using the connections described, external instrumentation was employed to monitor its behaviour whilst the software was tested. This tested that the software was able to run without crashing etc., however, testing of the performance was not possible until characterisation was performed.

5.2 Design of New Main Sensor Chamber

The design new main sensor chamber was a radical change from the design of the original sensor chamber. The diagrams in figure 5.3 show how the designs differed. In particular, the difference in the gas flow characteristics between the two designs can be observed. In the original design, the gas flow was turbulent and different responses would result from the same gas sensor being placed at different distances from the input. Those gas sensors furthest from the input would tend to have a faster response, although the gas appears at the input first, it accumulates faster near the output, this region becoming saturated first. Finally, the saturated region would extend to the gas sensors near the input and so speed up their response. Once the entire volume of the original chamber was saturated, an equilibrium would be reached where the concentration of odourants would be uniform. However, it is possible that temperature gradients could occur, the gas sensors contain heaters, when stabilisation occurred, the coolest gas would first occupy the region near the output, the gas sensors in this region would be cooled more than the others. Overall, before equilibrium, the odorant

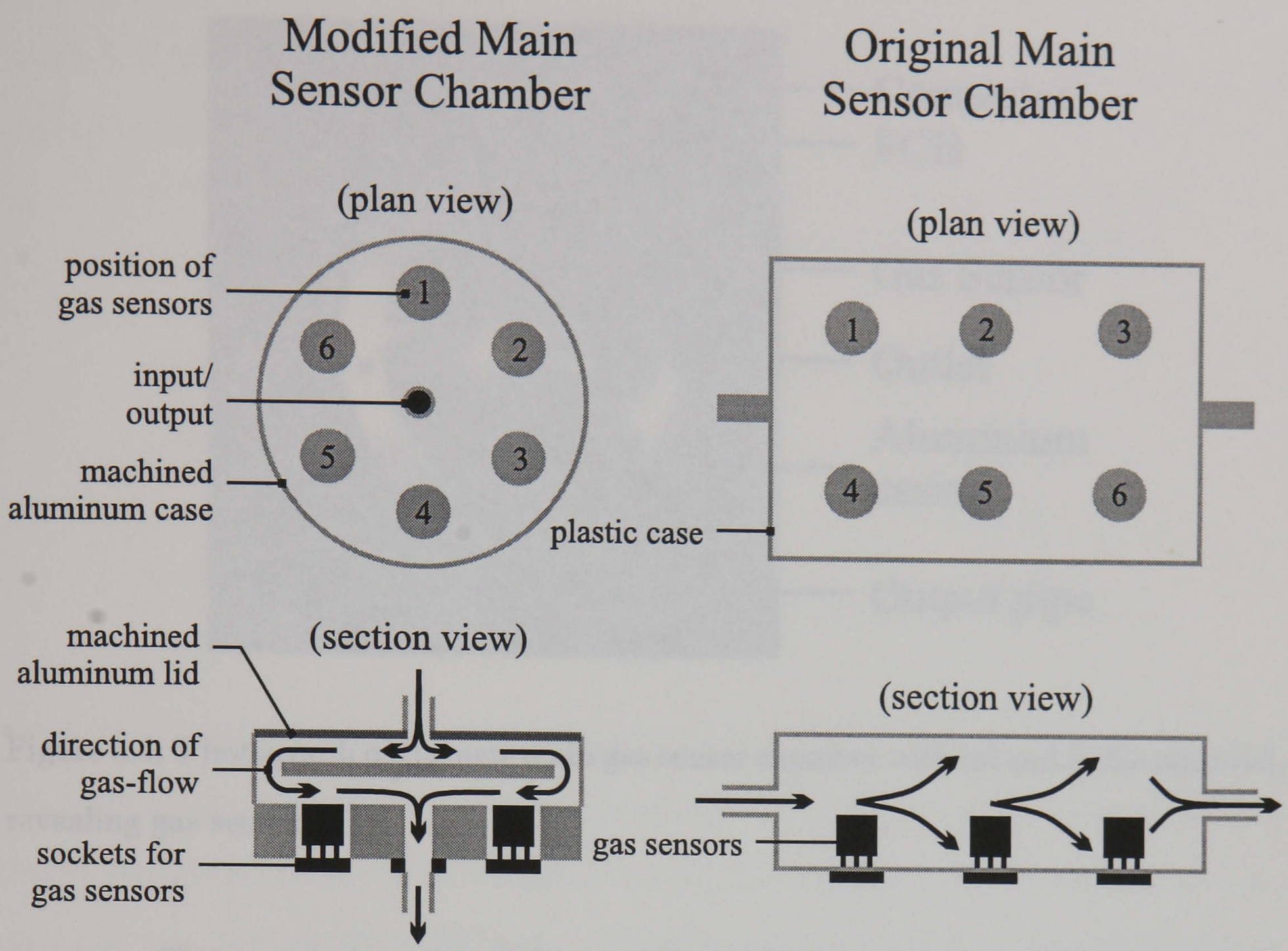


Figure 5.3: Diagram illustrating the new main sensor chamber fitted to the FOX 2000 compared with the original sensor chamber. The heater wrapped around the new main gas sensor chamber is not shown for clarity.

concentration differential would effect gas sensor response, and after equilibrium, the temperature differentials would effect gas sensor response.

The symmetry of the gas flow within the new main sensor chamber can be observed from figure 5.3. Upon entry, the gas was split into 6 equal streams. Each of the 6 gas sensors was exposed to a gas sub-stream. The sub-streams re-combined at the output. Therefore the concentration of odorants at each gas sensor changed equally and once equilibrium had been reached, the temperature differential between the gas sensors was negligible. A significant difference between the two designs was chamber volume, the original chamber had a volume of approximately 124 cm^3 and the new main gas chamber had a volume of approximately 14 cm^3. This is important because a smaller volume allows equilibrium to be attained faster, which in turn speeds overall gas sensor

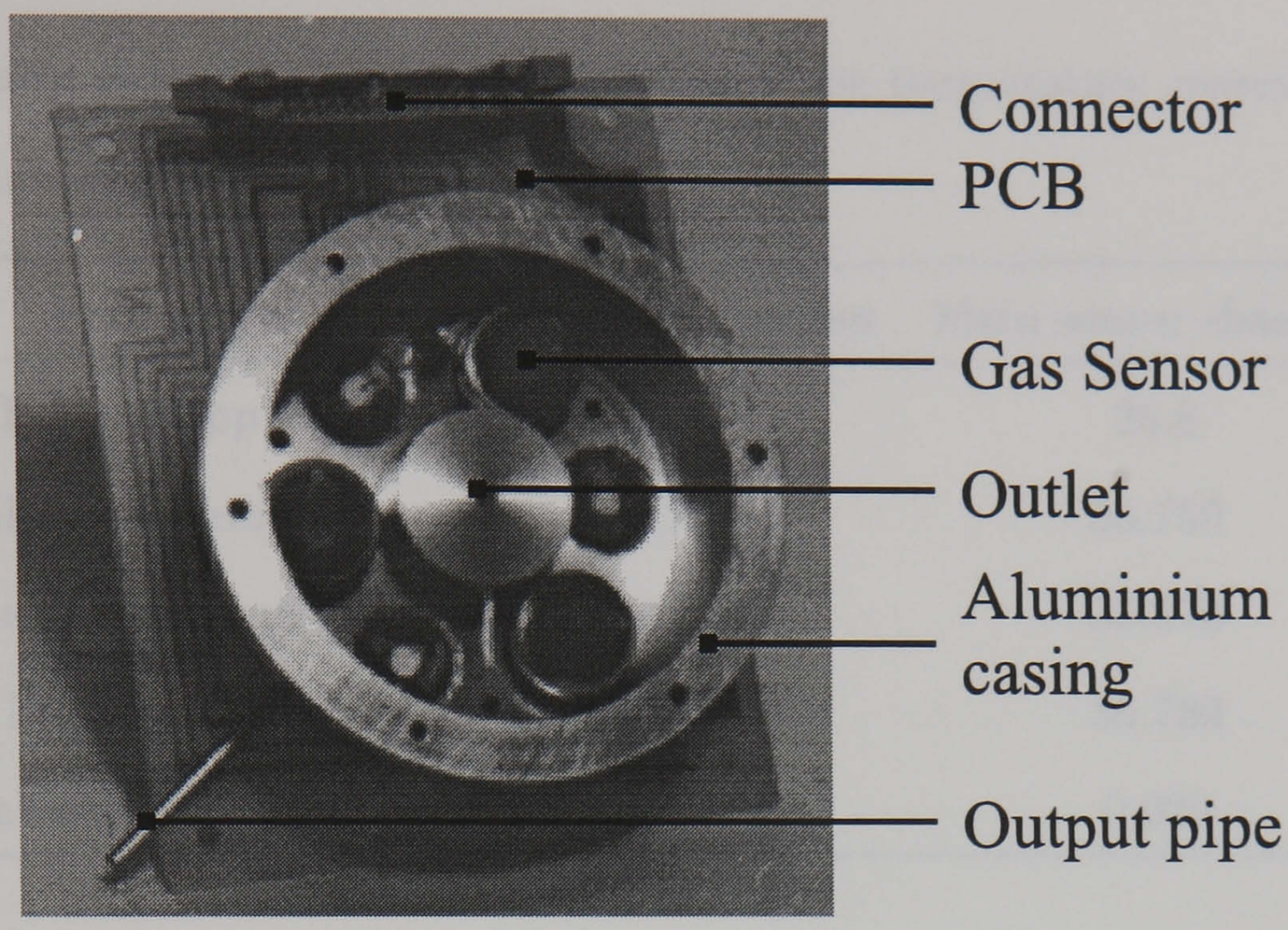


Figure 5.4: Photograph of the new main gas sensor chamber with lid and baffle removed, revealing gas sensors.

response. The new design allowed for easy dismantling in order to make changing the gas sensors less problematic. The lid was attached to the main casing with 6 equally spaced bolts around the circumference. Once the lid was removed, the internal baffles could be removed (this was held in place with a further 3 bolts) which allowed each gas sensor to be individually removed.

The temperature sensor was embedded approximately 20 mm into the aluminum casing. Standard heat sink compound was used to ensure a good heat conducting interface between the temperature sensor and the aluminum block. The temperature sensor therefore did not directly come into contact with the gas, but measured the temperature of the core of the aluminum block. The temperature differential between the core temperature of the aluminum block and the gas is set to be small.

5.3 System Testing and Characterisation

Before data gathering experiments were performed, it was necessary to characterise the new sub-system. The system was run in the 'standby' condition for 2 days prior to characterisation. The temperature control sub-system was powered up, target tem-

Table 5.1: Table summarising the performance of the temperature control sub-system during the characterisation test.

Parameter	Pre-heater chamber	Main sensor chamber
Target temp (°)	34.0	36.8
Minimum temp (°)	33.754	36.750
Maximum temp (°)	34.096	36.870
Mean temp (°)	33.945	36.789
Standard Deviation (°)	0.104	0.029

peratures of 34°C for the pre-heater chamber and 36.8°C for the main sensor chamber were set in the Temperature Control Program. This allowed the system to be in a stable condition before characterisation began. For the characterisation test, the system was run for a period of 23 hours with all vessels empty, the target temperature of the pre-heater chamber was set to 34°C and the target temperature of the main sensor chamber was set to 36.8° (the same as before). Table 5.1 summarises the statistical results from the characterisation experiment. It can be observed that the system meets the specifications (i.e. gas temperature in main sensor chamber ±0.1°C of target temperature). The larger standard deviation for the pre-heater chamber is larger than that for the main chamber, this shows that the temperature fluctuations for the pre-heater chamber were greater, as expected.

The plot in figure 5.5 shows the output from the temperature sensors for the duration of the test. The vertical axis for this plot is the same scale as that for the plot in figure 3.8 so that a comparison can be drawn. External monitoring equipment showed that the ambient temperature during this characterisation test varied from 20°C (during the night) to 26°C (during the day).

The performance was therefore tested at the temperatures of interest and at average ambient temperatures. The time constant for the pre-heater chamber was observed to be approximately 0.1°C s^{-1}, and for the main sensor chamber to be approximately 0.5°C min^{-1}, by observing the display of the Temperature Control Program. The new

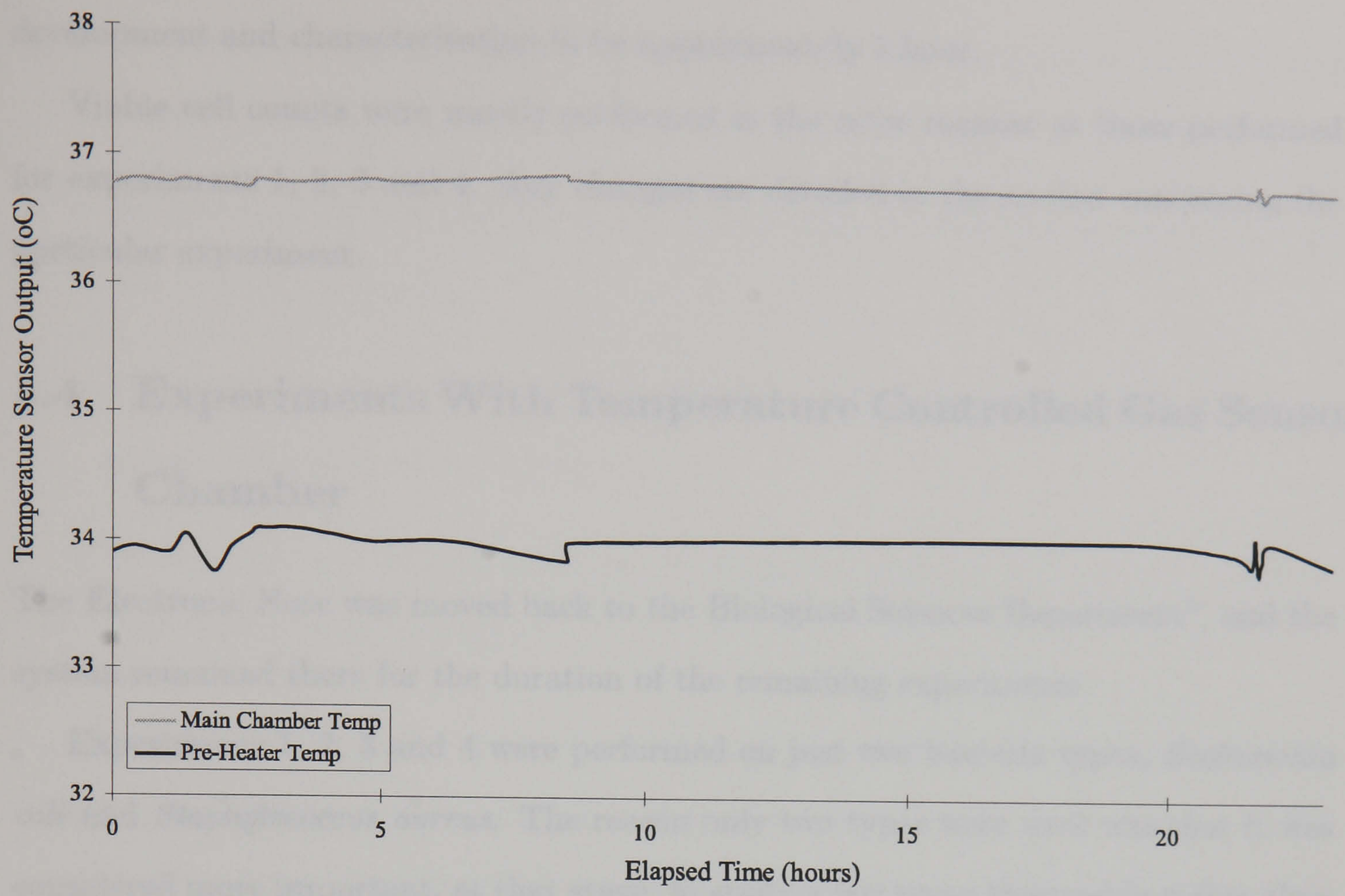


Figure 5.5: Plot showing the temperature of the gas in the pre-heater chamber and the main sensor chamber over a period of 24 hours.

modified electronic nose system was deemed to operate satisfactory and further data gathering experiments were then able to be performed.

5.3.1 Experimental Procedure Development

It was not necessary to repeat the development experimental procedures as a whole, with the operation of the new sub-system in mind, the original procedures were adopted with some minor changes.

The major change in experimental procedure was connected the with operation of the computer software. Before a data gathering experiment could be performed, the Electronic Nose had to be in a stable state (the 'standby' state), however, as well as the gas sensors being stable the temperature of the main sensor chamber also had to be stable. The 'stand-by' state was re-defined to include the Temperature Control Program running with the desired target temperatures being set. The length of time for temperature stabilisation to occur was estimated from observations made during

development and characterisation to be approximately 1 hour.

Viable cell counts were mostly performed in the same manner as those performed for experiments 1, 2, 3 and 4. Any changes are detailed in the section concerning the particular experiment.

5.4 Experiments With Temperature Controlled Gas Sensor Chamber

The Electronic Nose was moved back to the Biological Sciences Department[5], and the system remained there for the duration of the remaining experiments.

Experiments 1, 2, 3 and 4 were performed on just two bacteria types, *Escherichia coli* and *Staphylococcus aureus*. The reason only two types were used was that it was considered more important, at that stage, to study a few types thoroughly rather than a large number in less detail. Analysis of the data from these experiments allowed some insight to be gained as to the potential problems, such as sensor drift. In order to test the performance of odour classification further, it was decided to perform data gathering experiments on more bacteria types and also of mixtures of 2 different bacteria types. The analysis of odour data from a mixture of bacteria types represents a more difficult challenge, and is also more realistic of the growth environment in a patient. After consultation with members of staff at the Biological Sciences Department, it was decided to perform data gathering experiments on the following odour samples:

- *Escherichia coli*, this is the benchmark gram -ve bacteria. It also served as a control because it was used in experiments 1 and 2.

- *Staphylococcus aureus*, this is a common gram +ve bacteria and served as control because it was used in experiments 3 and 4.

- *Pseudomonas aeruginosa*, this is a common gram -ve bacteria that grows easily in NB and also is a possible pathogen.

[5]This time the experiments were conducted in an unused laboratory, which provided a more stable environment. Experiments 1,2,3 and 4 were conducted in the instrumentation laboratory, where there was considerable activity of both personnel and other equipment (thus the environment was less stable).

- *Streptococcus pyogenes* this is the most common bacterial pathogen in the upper respiratory system.

- *Escherichia coli* and *Staphylococcus aureus* mixture, this mixed a gram-ve type with a gram +ve type.

- *Pseudomonas aeruginosa* and *Staphylococcus aureus* mixture, this mixed a gram-ve type with a gram +ve type.

- *Escherichia coli* and *Pseudomonas aeruginosa* mixture, this mixed a gram-ve type with a gram +ve type.

Other considerations for choice of bacteria types was availability and handling. The bacteria types employed are common and the laboratory can prepare cultures within a short time. More exotic bacteria types have to be specially ordered. Also handling was an important consideration. The laboratories used were classed as hazard level 2, this means that the microorganisms that are allowed to be handled are up to and including hazard level 2. In microbiology, microorganisms are classed from hazard level 1 to hazard level 4. If a person is contaminated, hazard level 1 microorganisms are those that pose no threat to health; level 2 are those that are potentially a mild threat to health; level 3 are those that pose a serious threat to health but can be treated; level 4 are those that are deadly and have no treatment (for example *Escherichia coli* 0157). For safety reasons, the experiments were only permitted to be carried out in a level 2 environment. Many ENT pathogens, for example *Corynebacterium diphtheriae* or *Haemophilus influenzae* were too dangerous to use[6].

It can be noticed that ***Streptococcus pyogenes*** was not used in a mixture. This bacteria type did not grow well in a NB growth medium, when mixed with one of the other types it would have been wiped out by the other type in the first 1 or 2 hours.

Two data gathering experiments were performed employing each single bacteria type odour source, and 1 was performed employing each mixture, a total of 11 experiments. The experimental setup was identical to that employed for the first 4 experiments

[6]Trained staff might be able to perform experiments on level 3 organisms, new experimental procedures would be necessary, especially for equipment cleaning

apart from the odour source used as the reference. The original odour source was pure nutrient broth, this was found to be potentially problematic (discovered performing the work detailed in chapter 4), therefore in the remaining experiments the reference odour was air (i.e. vessel #1 was empty).

The target temperature of the pre-heater chamber was set to 34.0°C and the temperature of the main sensor chamber was set to 36.8°C. The temperature differential between the pre-heater chamber and the main sensor chamber was necessary in order to allow accurate control of the temperature of the main sensor chamber. Since this chamber only had a heating element, temperature reduction was achieved by the cooling effect of the gas, which was initially 2,8°C cooler. The gas entering the pre-heater chamber was cool enough to ensure accurate control (this temperature approximately ranged from 20°C to 30°C).

5.4.1 Experiments Performed on

The first two data gathering experiments were performed on *Escherichia coli*. The odour sources were 25 ml NB in vessel #1 and aqueous cultures of 25 ml NB inoculated with 0.25 ml *Escherichia coli* 'master' culture in vessels #2 and #3. These experiments were denoted as experiment 5 and 6, respectively. During experiment 5, the ambient temperature ranged from 18°C to 21°C and the relative humidity ranged from 39% to 40%. During experiment 6, the ambient temperature ranged from 18°C to 19°C and the relative humidity ranged from 37% to 42%. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters. Figure 5.6 and figure 5.7 show the results of the viable cell counts performed during experiments 5 and 6, respectively.

As expected the growth curves for these two experiments were similar to those for experiment 1 and 2. In all these experiments the maximum population count is in the same order of magnitude. In experiment 5 the a lag phase was detected, in experiment 6 the lag phase was too short to be measured.

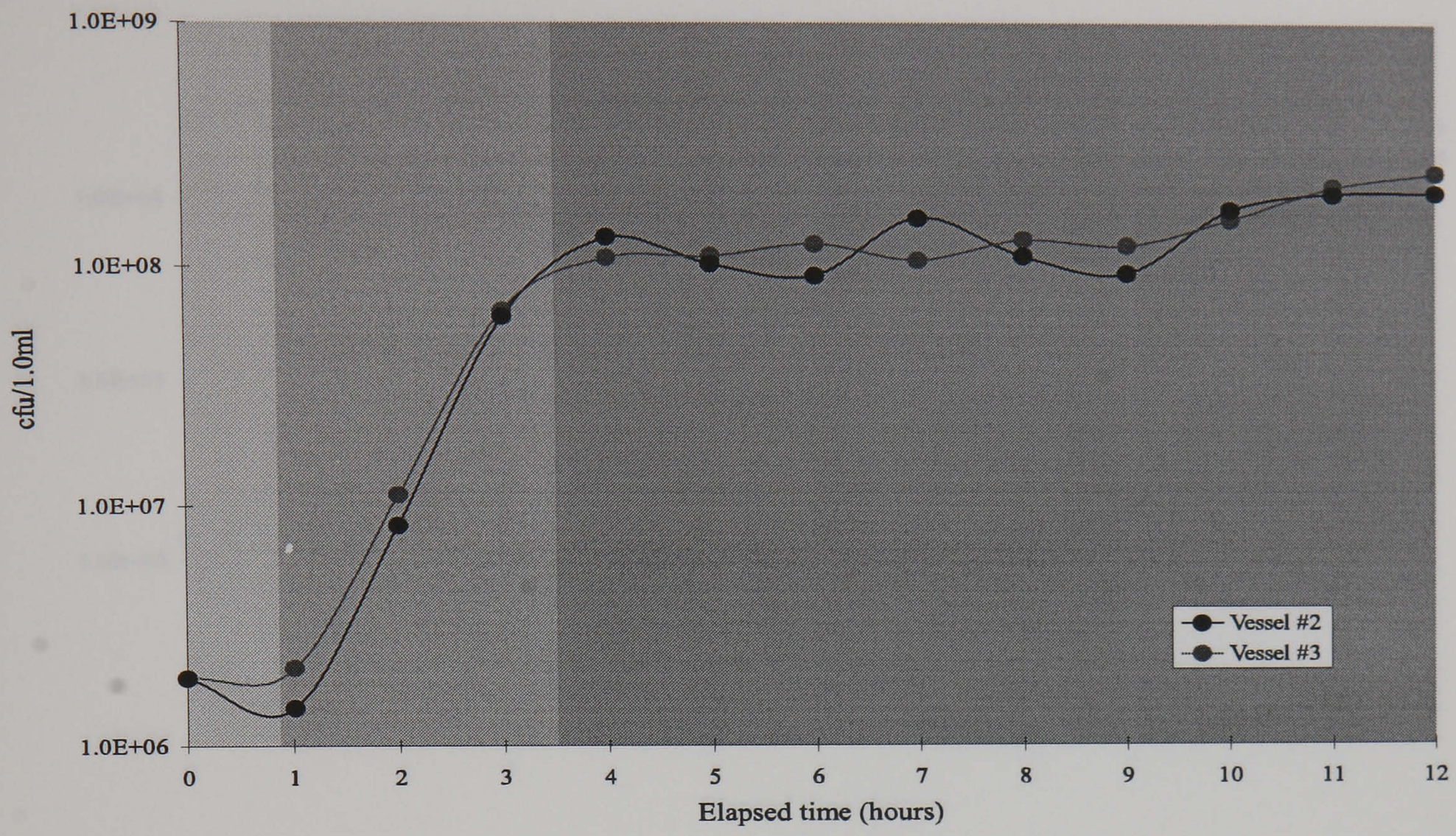


Figure 5.6: Plot of the number of colony forming units (cfu) in 0.1 ml of inoculum for *Escherichia coli* experiment 5, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

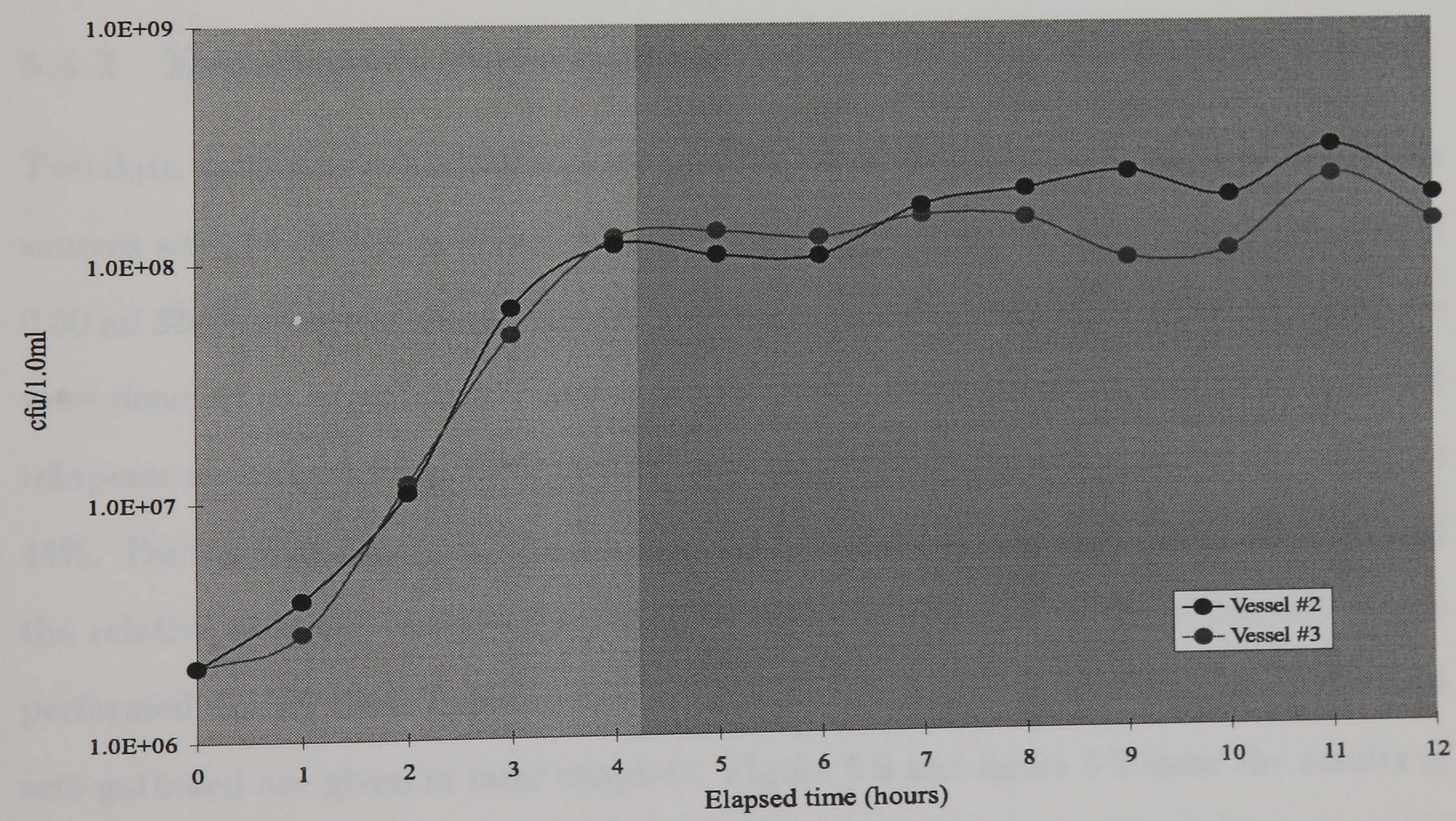


Figure 5.7: Plot of the number of colony forming units (cfu) in 0.1 ml of inoculum for *Escherichia coli* experiment 6, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

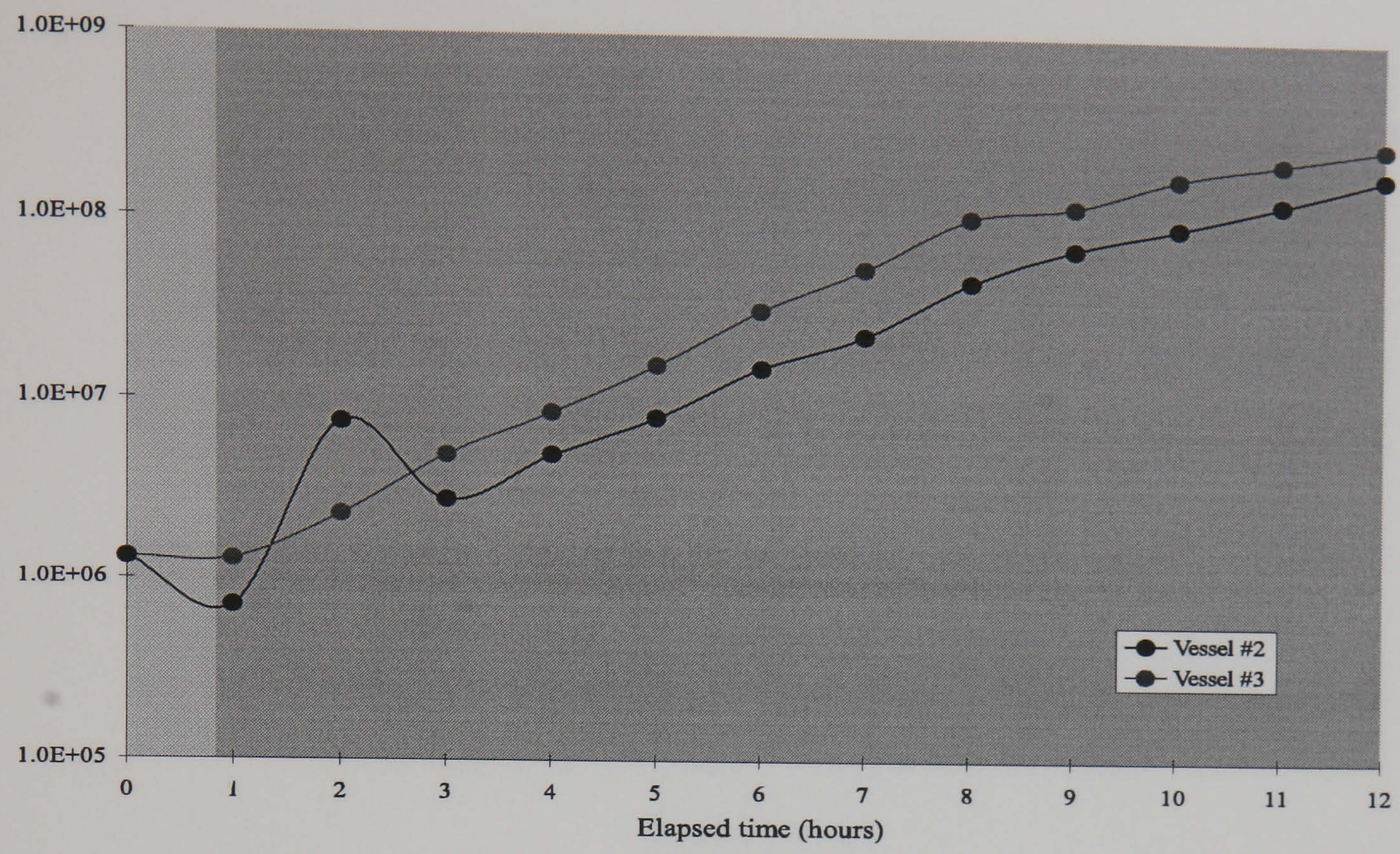


Figure 5.8: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Staphylococcus aureus* experiment 7, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

5.4.2 Experiments Performed on

Two data gathering experiments were performed on *Staphylococcus aureus*. The odour sources were 25 ml NB in vessel #1 and aqueous cultures of 25 ml NB inoculated with 0.50 ml *Staphylococcus aureus* 'master' culture in vessels #2 and #3. These experiments were denoted as experiment 7 and 8, respectively. During experiment 7, the ambient temperature ranged from 19°C to 21°C and the relative humidity ranged from 39% to 44%. During experiment 8, the ambient temperature ranged from 16°C to 22°C and the relative humidity ranged from 31% to 39%. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters. Figure 5.8 and figure 5.9 show the results of the viable cell counts performed during experiments 7 and 8, respectively.

The maximum culture population for these two experiments tended to be slightly higher than those for experiments 3 and 4, the reason for this was not precisely traced, however it is possible that there was some variation in the quality of the growth medium.

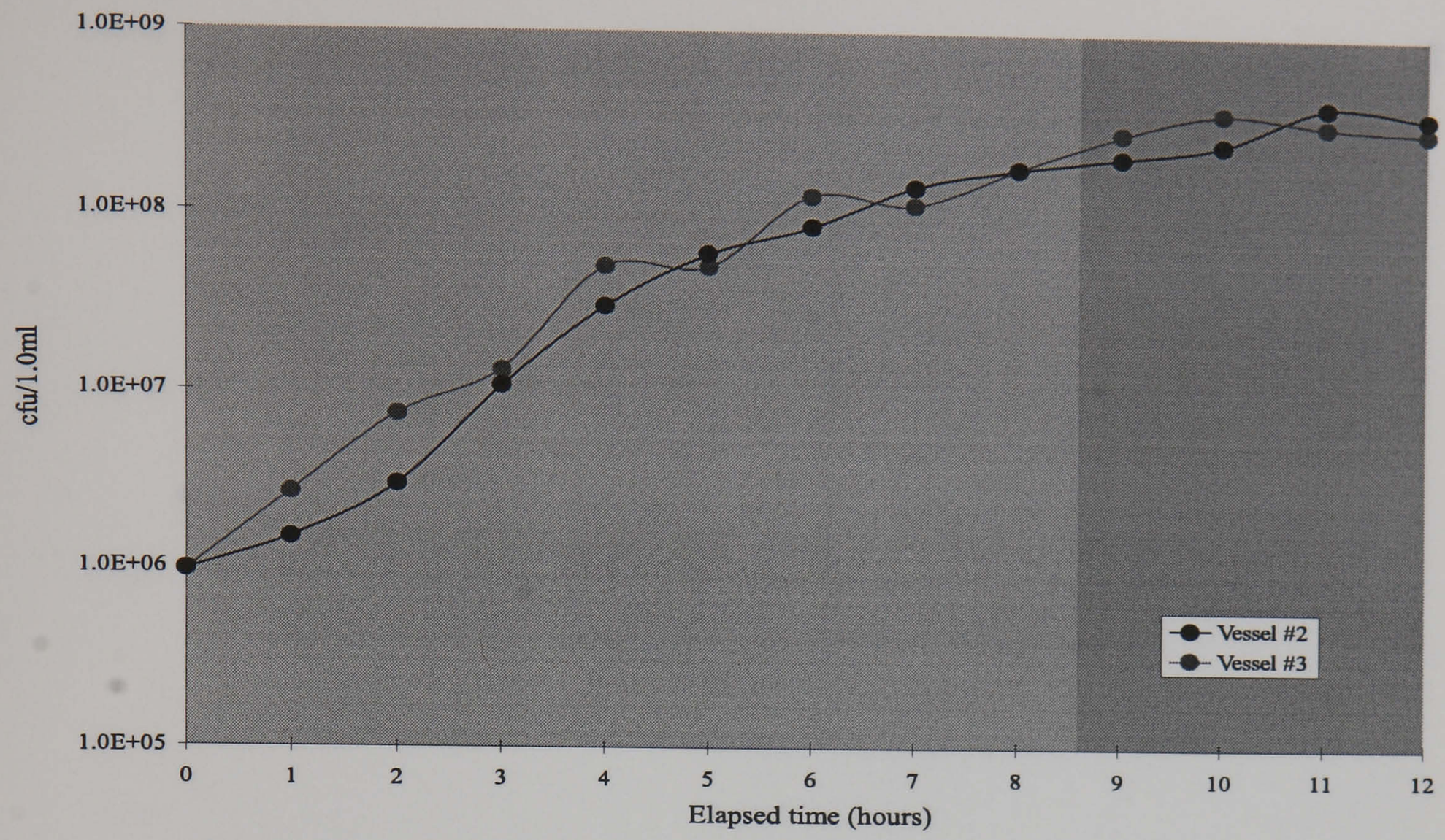


Figure 5.9: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for ***Staphylococcus aureus*** experiment 8, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

The viable cell counts for experiment 7 show no stationary phase, the populations continued to grow steadily. The reason for this was unknown, the rate of growth can be observed to decline with time, if further measurements had been made it is likely the stationary phase would have been detected.

5.4.3 Experiments Performed on

Two data gathering experiments were performed on *Pseudomonas aeruginosa.* The odour sources were 25 ml NB in vessel #1 and aqueous cultures of 25 ml NB inoculated with 0.25 ml *Pseudomonas aeruginosa* 'master' culture in vessels #2 and #3. These experiments were denoted as experiment 9 and 10, respectively. During experiment 9, the ambient temperature ranged from 18°C to 19°C and the relative humidity ranged from 36% to 39%. During experiment 10, the ambient temperature ranged from 18°C to 20°C and the relative humidity ranged from 34% to 39%. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of

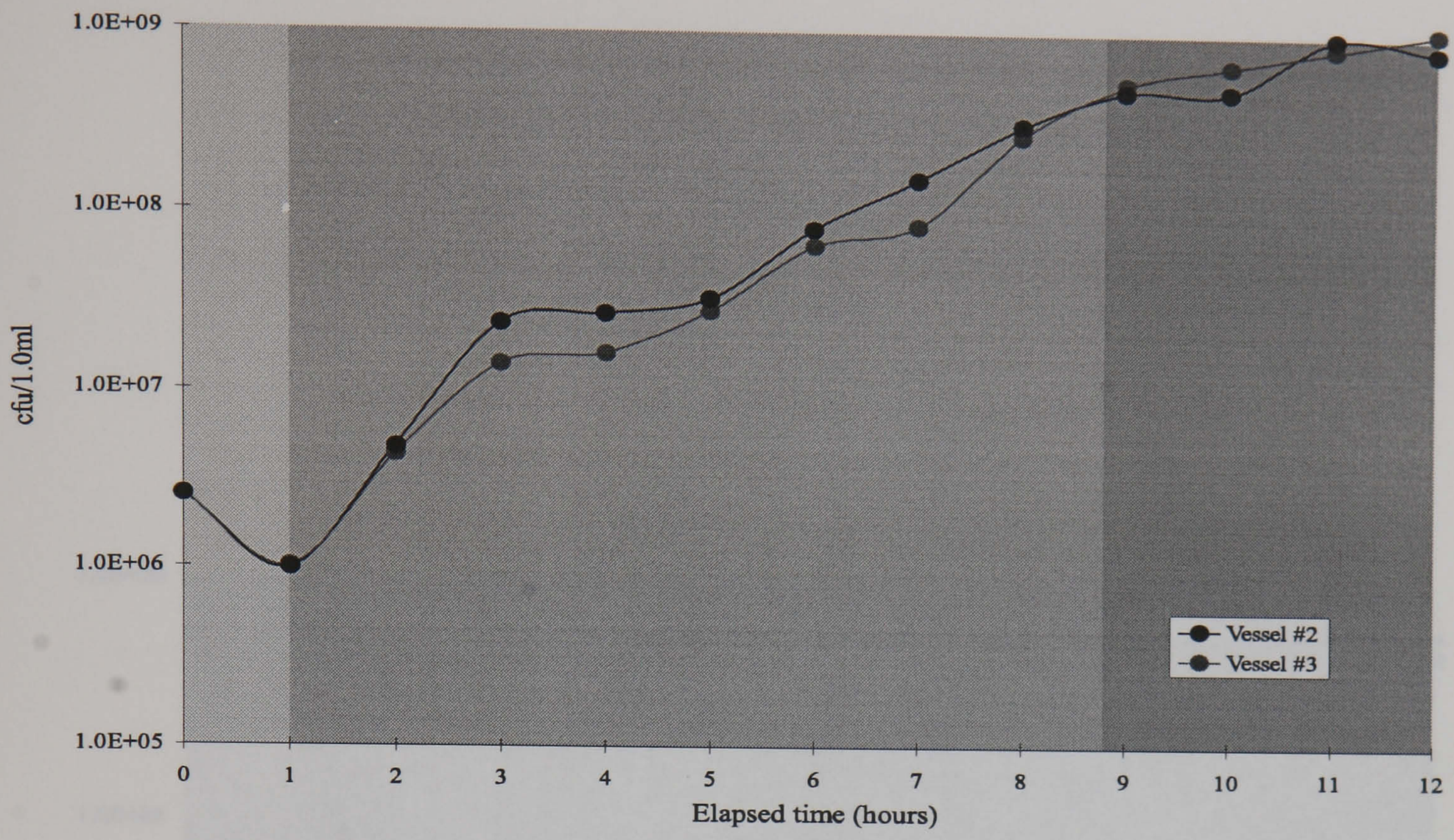


Figure 5.10: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Pseudomonas aeruginosa* experiment 9, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

the data sets gathered are given in later chapters. Figure 5.10 and figure 5.11 show the results of the viable cell counts performed during experiments 9 and 10, respectively.

Pseudomonas aeruginosa grows well in a NB growth medium (as it does in most general purpose growth media), like *Escherichia coli*, it is motile which means that it tends to grow well in aqueous media. The growth plots reflect the ability of this bacteria type to grow fast. Also *Pseudomonas aeruginosa* is a bacteria type that has a particularly characteristic odour which was often smelt and identified by micro-biology staff.

5.4.4 Experiments Performed on

Two data gathering experiments were performed on *Streptococcus pyogenes*. The odour sources were 25 ml NB in vessel #1 and aqueous cultures of 25 ml NB inoculated with 0.50 ml *Streptococcus pyogenes* 'master' culture in vessels #2 and #3. These experiments were denoted as experiment 11 and 12, respectively. During experiment

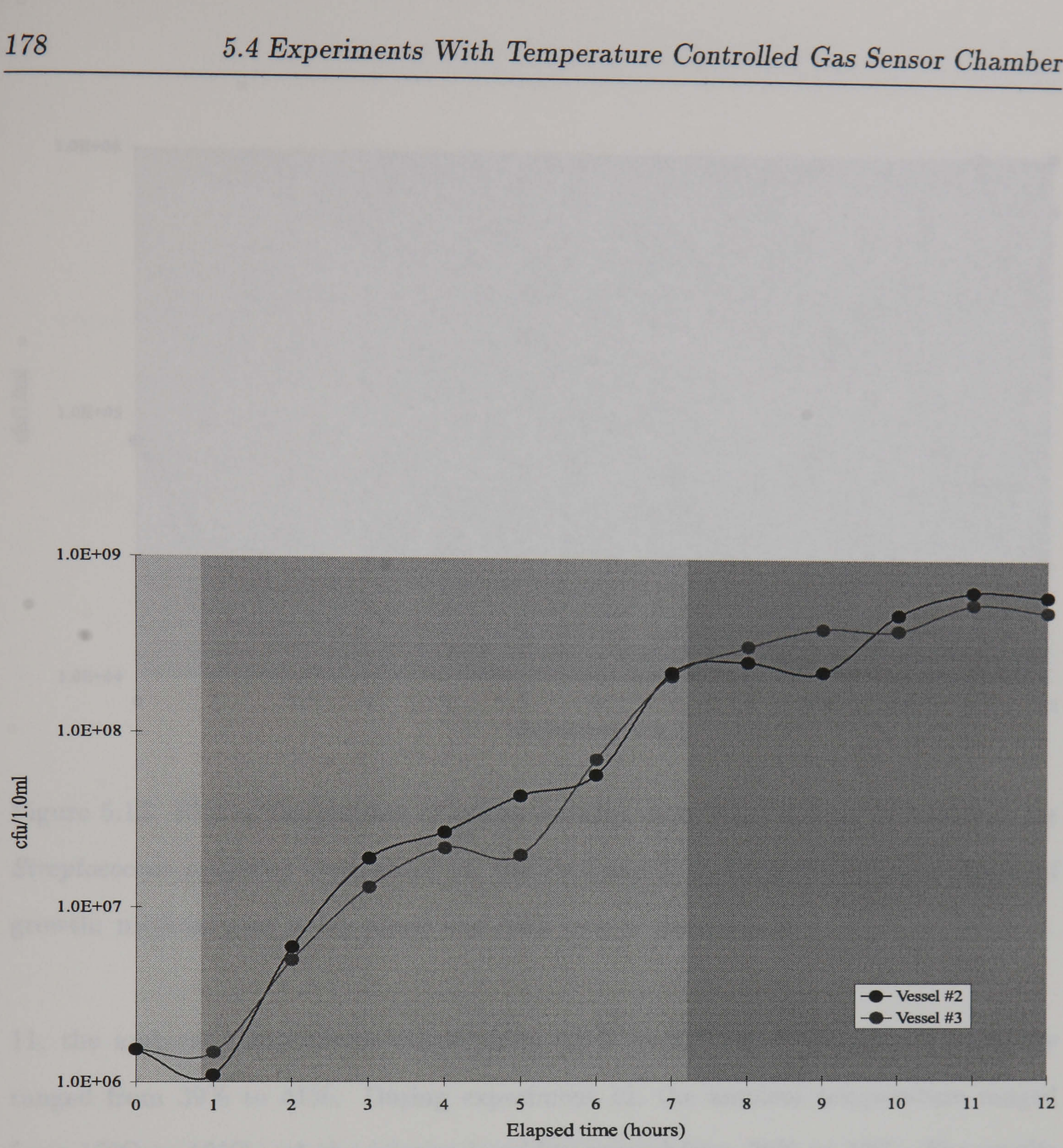


Figure 5.11: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Pseudomonas aeruginosa* experiment 10, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

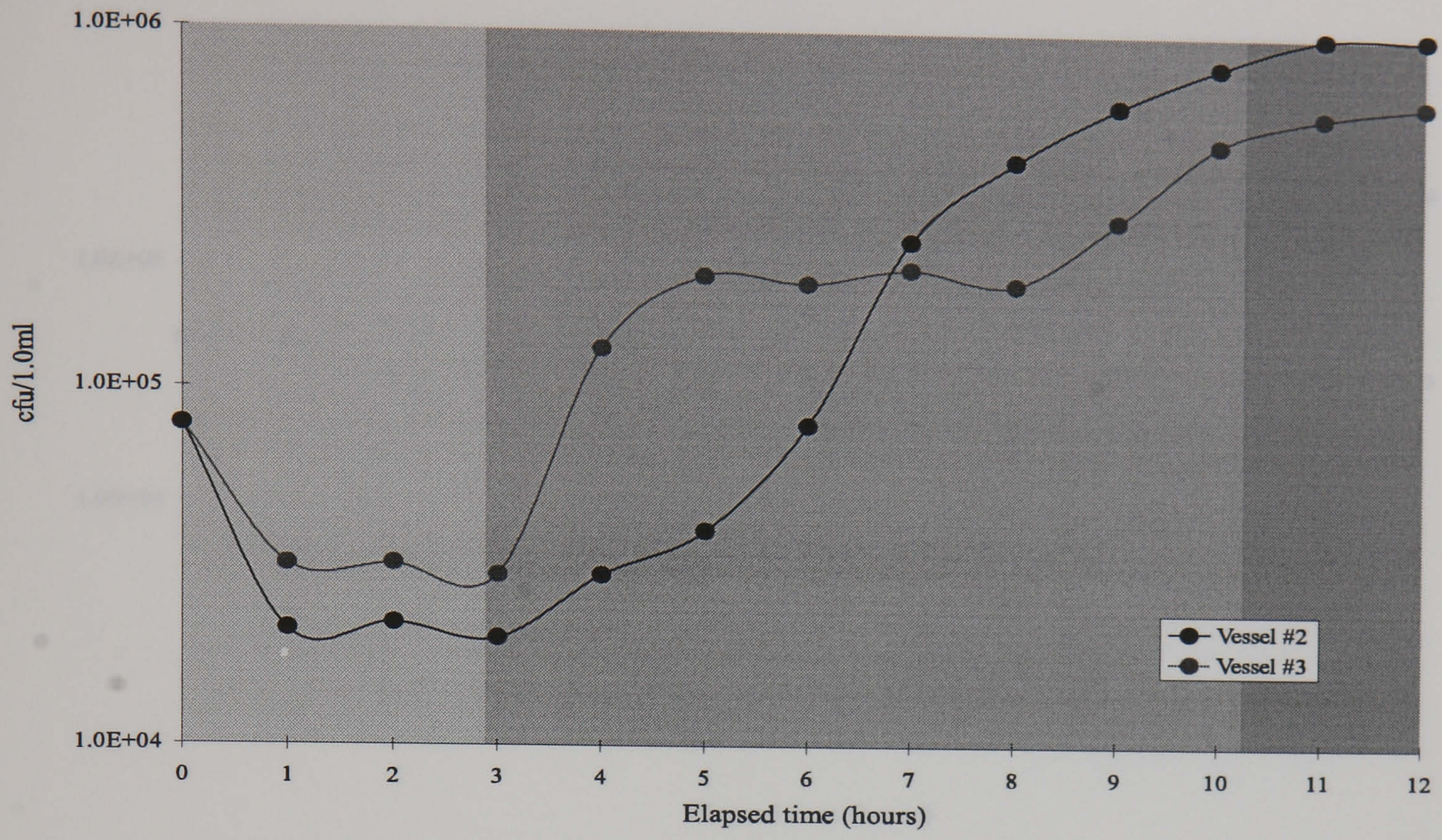


Figure 5.12: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Streptococcus pyogenes* experiment 11, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

11, the ambient temperature ranged from 18°C to 21°C and the relative humidity ranged from 39% to 41%. During experiment 12, the ambient temperature ranged from 16°C to 19°C and the relative humidity ranged from 39% to 43%. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters. Figure 5.12 and figure 5.13 show the results of the viable cell counts performed during experiments 11 and 12, respectively.

The culture populations in these two experiments did not reach high levels, they were several orders of magnitude less than the other bacteria types. The populations in experiment 12 show particularly slow growth. *Streptococcus pyogenes* does not grow well in a NB growth medium, an anaerobic environment is also preferential. Also, being nonmotile, an aqueous growth medium was the best. It would have been possible to grow the cultures in better suited conditions, but it was felt that continuity in the experiment was important and that measuring the odour from a different bacteria type

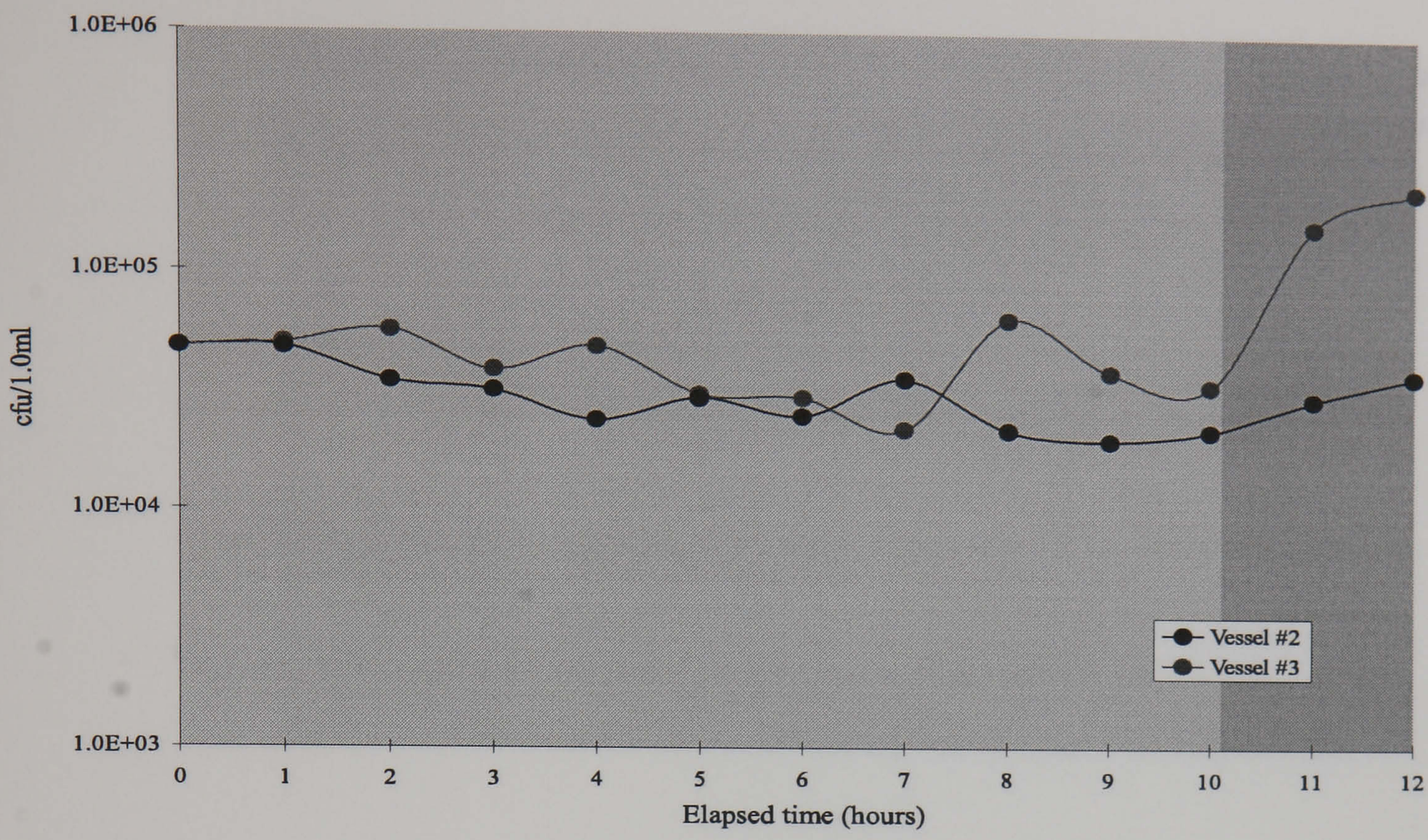


Figure 5.13: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Streptococcus pyogenes* experiment 12, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

and growth media type would have caused complications. It was decided that although the populations were small, they were large enough to generate odour.

The growth medium used in the Petri dishes used for the viable cell counts was not NA but blood agar. The reason for this is that *Streptococcus pyogenes* does not grow well on NA, and the colonies that would have formed on such a medium would have been slow growing and very small. Blood agar is simply sterile defibrinated blood (usually ox or sheep blood) mixed with agar base. Colonies of *Streptococcus pyogenes* have a hemolytic effect on the growth medium, i.e. the area surrounding the colony becomes transparent. Also *Streptococcus pyogenes* grow better on blood agar, thus the colonies were well formed and easily identifiable.

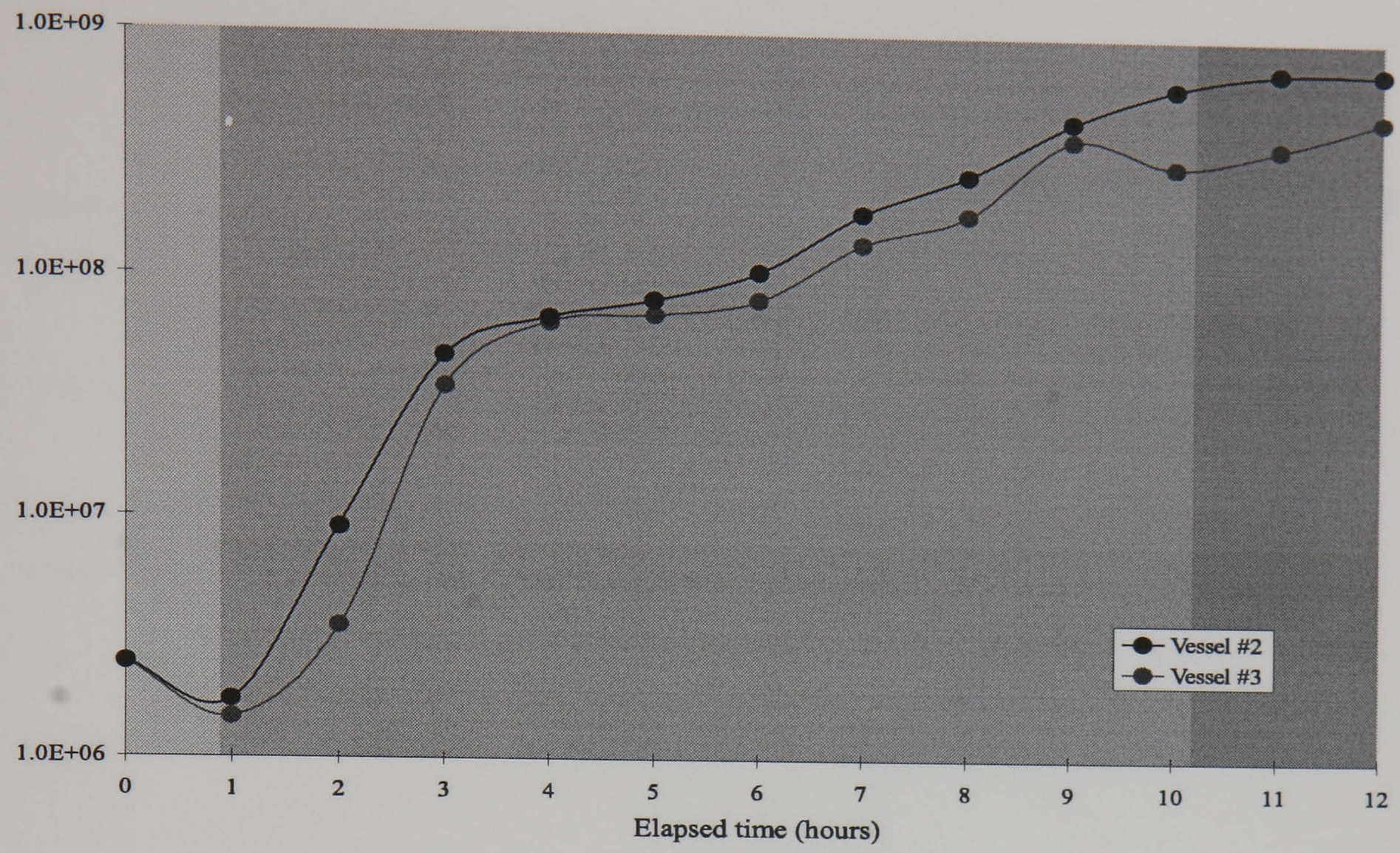


Figure 5.14: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Escherichia coli* experiment 13, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

5.4.5 Experiment Performed on and mixture

One data gathering experiment was performed on *Escherichia coli* and *Staphylococcus aureus* mixed. The odour sources were 25 ml NB in vessel #1 and aqueous cultures of 25 ml NB inoculated with 0.25 ml *Escherichia coli* and 0.50 ml *Staphylococcus aureus* 'master' culture in vessels #2 and #3. This experiment was denoted as experiment 13. During experiment 13, the ambient temperature ranged from 17°C to 19°C and the relative humidity ranged from 39% to 42%. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters. Figure 5.14 show the results of the viable cell counts for *Escherichia coli*, and figure 5.15 show the results of the viable cell counts for *Staphylococcus aureus*, performed during experiment 13.

As expected *Escherichia coli* showed a consistently higher population size than *Staphylococcus aureus*, however, the latter still managed to grow despite the former

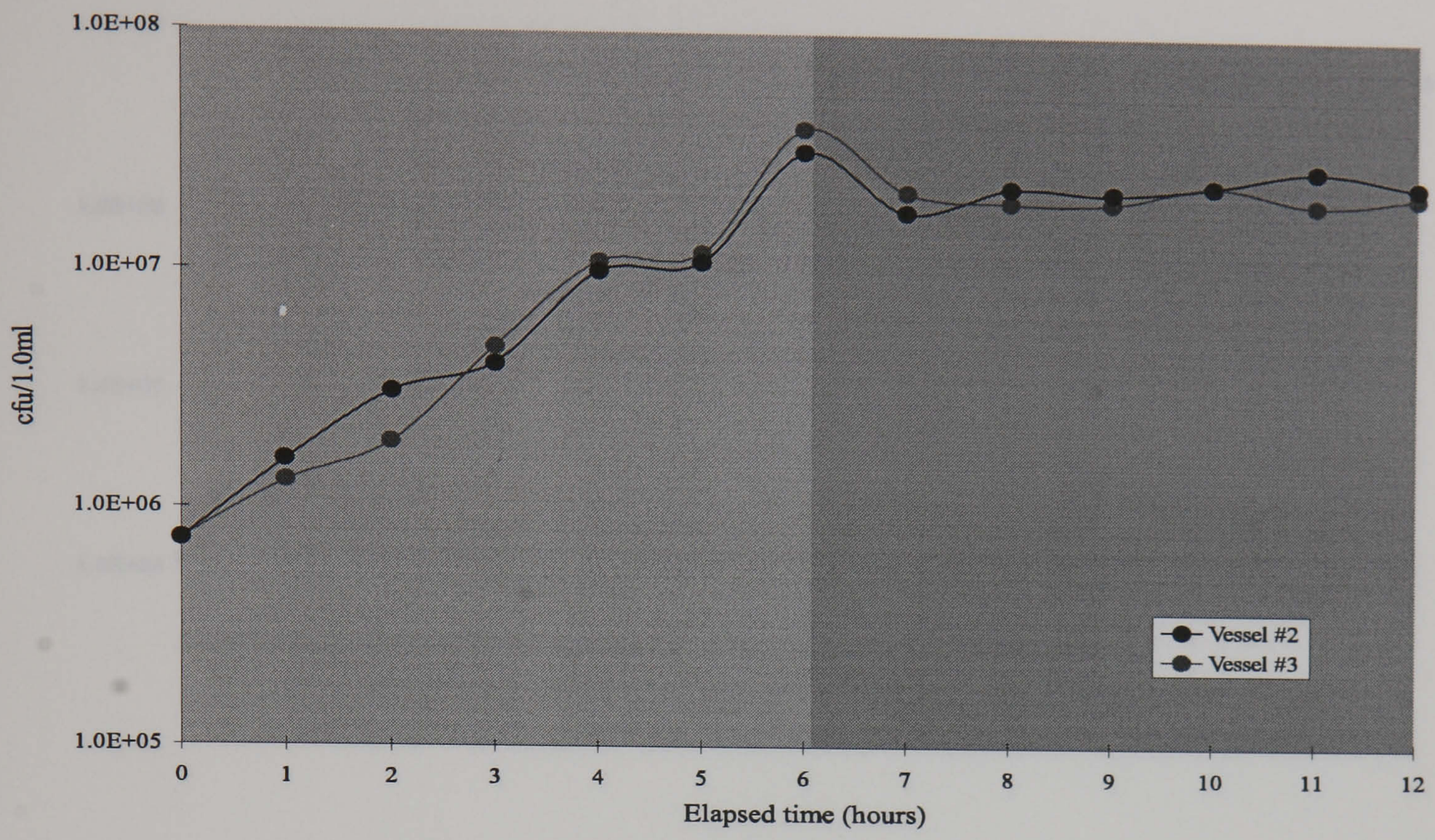


Figure 5.15: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for ***Staphylococcus aureus*** experiment 13, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

using up the available nutrients.

The growth medium used in the Petri dishes was NA, colonies of ***Escherichia coli*** appear large, off-white in colour with no clear defined edges, colonies of ***Staphylococcus aureus*** appear white, small and well defined. Thus it was possible to detect the different types of colony with sufficient accuracy.

5.4.6 Experiment Performed on and mixture

One data gathering experiment was performed on ***Pseudomonas aeruginosa*** and ***Staphylococcus aureus*** mixed. The odour sources were 25 ml NB in vessel #1 and aqueous cultures of 25 ml NB inoculated with 0.25 ml ***Pseudomonas aeruginosa*** and 0.50 ml ***Staphylococcus aureus*** 'master' culture in vessels #2 and #3. This experiment was denoted as experiment 14. During experiment 14, the ambient temperature ranged from 17°C to 20°C and the relative humidity ranged from 37% to 41%. The results

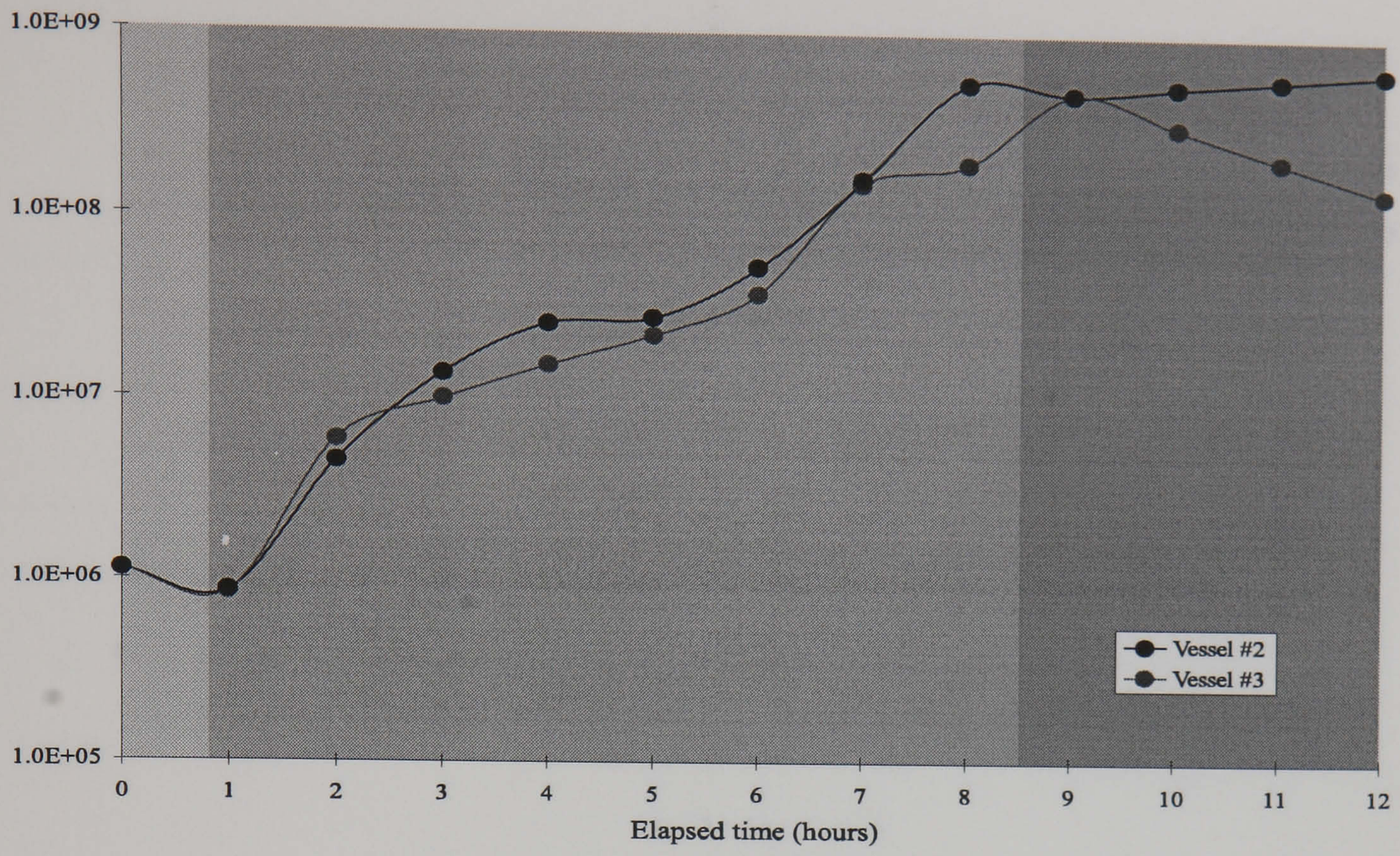


Figure 5.16: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Pseudomonas aeruginosa* experiment 14, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters. Figure 5.16 show the results of the viable cell counts for *Pseudomonas aeruginosa*, and figure 5.17 show the results of the viable cell counts for *Staphylococcus aureus*, performed during experiment 14.

The population counts for *Pseudomonas aeruginosa* were again very high, the population in vessel #3 did start to decline, whether this was a sign of entry into the death phase or whether it was a 'hiccup' was unknown. As expected, *Staphylococcus aureus*, had populations an order of magnitude less than the former type, like experiment 13, it survived despite the presence of a dominant organism.

The growth medium used in the Petri dishes was NA, colonies of *Pseudomonas aeruginosa* appear large, yellowish with no clear defined edges, colonies of *Staphylococcus aureus* appear white, small and well defined. Thus it was possible to detect the different types of colony with sufficient accuracy.

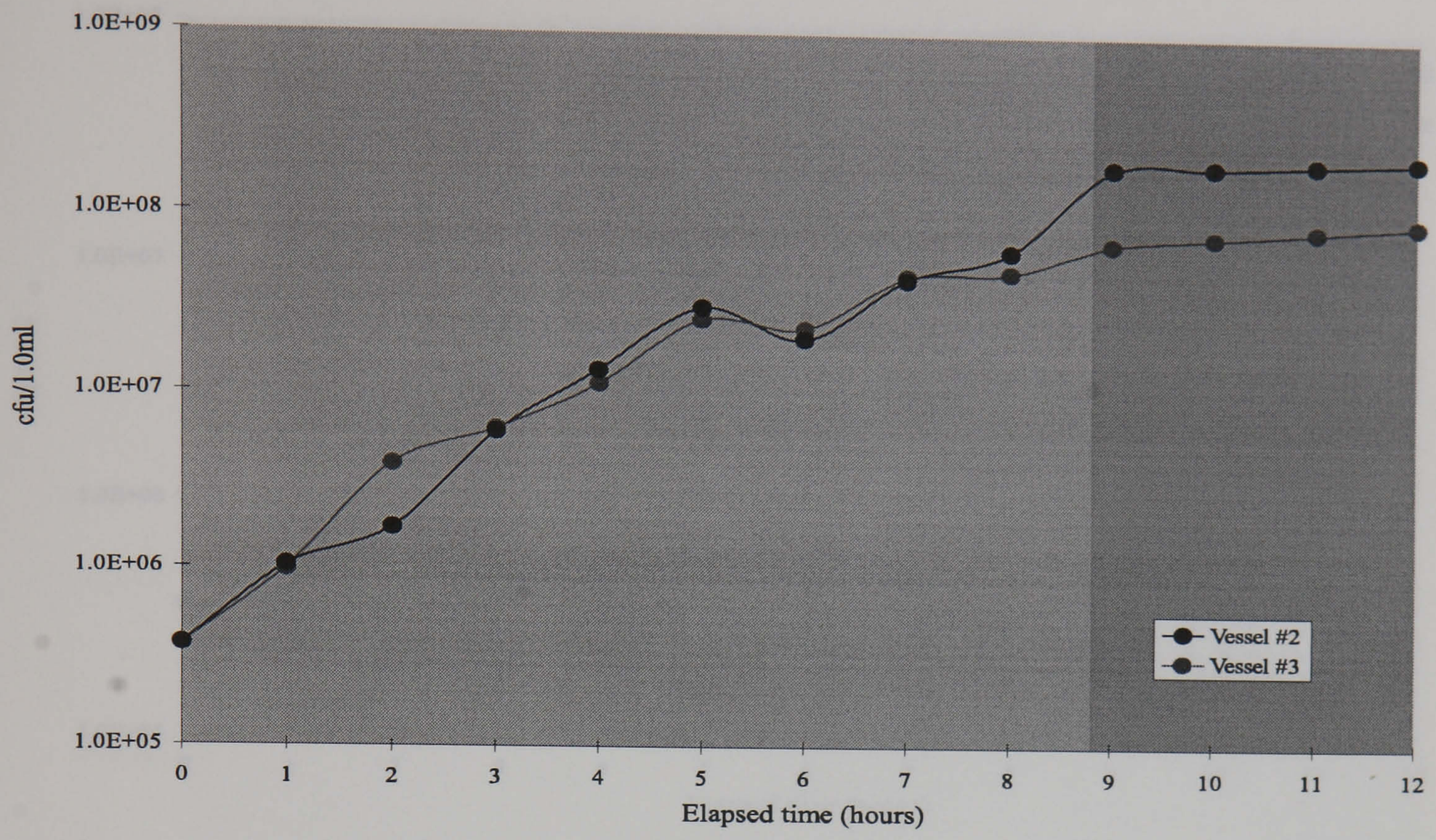


Figure 5.17: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for ***Staphylococcus aureus*** experiment 14, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

5.4.7 Experiments Performed on and mixture

One data gathering experiment was performed on *Escherichia coli* and *Pseudomonas aeruginosa* mixed. The odour sources were 25 ml NB in vessel #1 and aqueous cultures of 25 ml NB inoculated with 0.25 ml *Escherichia coli* and 0.25 ml *Pseudomonas aeruginosa* 'master' culture in vessels #2 and #3. This experiment was denoted as experiment 15. During experiment 15, the ambient temperature ranged from 18°C to 21°C and the relative humidity ranged from 38% to 42%. The results of the viable cell counts performed during these experiments are given in appendix B, the analysis of the data sets gathered are given in later chapters. Figure 5.18 show the results of the viable cell counts for *Escherichia coli*, and figure 5.19 show the results of the viable cell counts for *Pseudomonas aeruginosa*, performed during experiment 15.

Both bacteria types showed similar culture population sizes. *Pseudomonas aeruginosa* has smaller population counts than those measured in previous experiments,

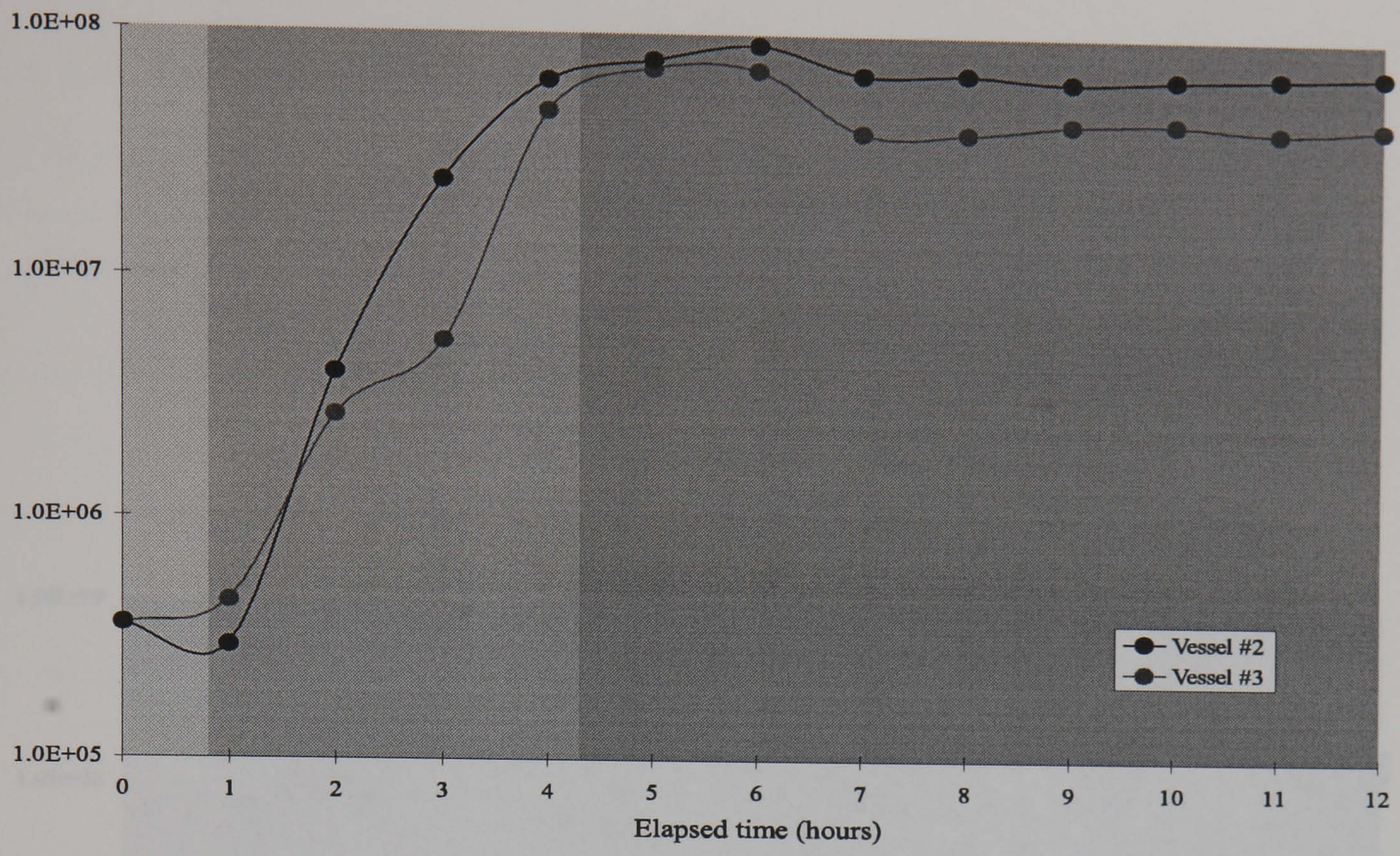


Figure 5.18: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Escherichia coli* experiment 15, vessels 2 and 3; showing the different phases of growth: medium grey = log phase and dark grey = static phase.

this was probably due to *Escherichia coli* using up the available nutrients. No one bacteria type dominated the other to any large degree.

The growth medium used for the viable cell counts was not NA but MacConkey Agar. The Petri dishes were prepared in a similar manner to the Petri dishes with NB. Colonies of *Escherichia coli* show up pink and colonies of *Pseudomonas aeruginosa* show up off-white colour. On NA, the colonies would have appeared too similar for viable cell counts to be performed with any accuracy. Also these bacteria types grow equally well on this growth medium, so colonies of one type did not mask out the colonies of the other. MacConkey Agar is popularly used to identify lactose-fermenting colonies, such as *Escherichia coli* from other types.

5.5 Summary

Following on from the analysis described in the previous chapter, a number of issues were highlighted. Firstly, there was unwanted variance in the data caused by fluctu-

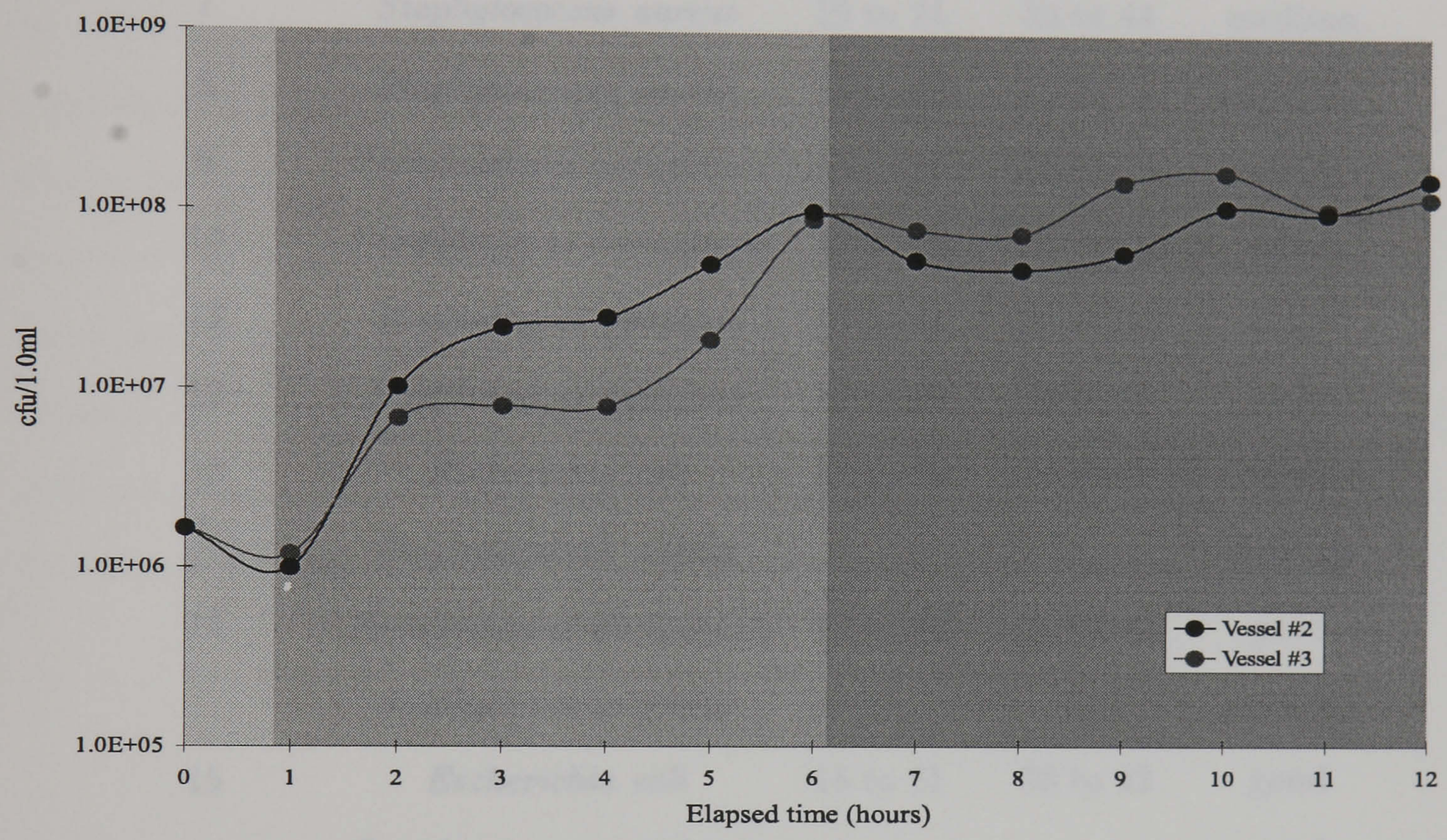


Figure 5.19: Plot of the number of colony forming units (cfu) in 1 ml of inoculum for *Pseudomonas aeruginosa* experiment 15, vessels 2 and 3; showing the different phases of growth: light grey = lag phase, medium grey = log phase and dark grey = static phase.

Table 5.2: Table summarising the data gathering experiments performed using the temperature controlled sensor chamber.

Experiment no.	Bacteria types	External temp °C	External humidity %	Growth curve
5	*Escherichia coli*	18 to 21	39 to 40	good
6	*Escherichia coli*	18 to 19	37 to 42	good
7	*Staphylococcus aureus*	19 to 21	39 to 44	medium
8	*Staphylococcus aureus*	16 to 22	31 to 39	medium
9	*Pseudomonas aeruginosa*	18 to 19	36 to 39	good
10	*Pseudomonas aeruginosa*	18 to 20	34 to 39	good
11	*Streptococcus pyogenes*	18 to 21	36 to 41	poor
12	*Streptococcus pyogenes*	16 to 19	39 to 41	poor
13	*Escherichia coli*	17 to 19	39 to 42	good
	Staphylococcus aureus			medium
14	*Pseudomonas aeruginosa*	17 to 20	37 to 41	good
	Staphylococcus aureus			medium
15	*Escherichia coli*	18 to 21	38 to 42	good
	Pseudomonas aeruginosa			good

ations in ambient temperature and other environmental causes (such as poor gas flow characteristics within the sensor chamber). Secondly, that data on more different bacteria types was needed. Finally, that it would be desirable to collect data from mixtures of bacteria.

In response to the issues highlighted, the electronic nose was modified further. A new gas sensor chamber was designed and implemented that incorporated improved gas flow characteristics, this chamber contained the gas sensors and a temperature sensor. A sub-system that controlled the ambient air temperature was also designed and implemented. This sub-system consisted of two heating elements, one incorporated into the main gas sensor chamber, and the other incorporated into a pre-heater chamber.

A second sensor chamber existed between the pre-heater chamber and the main gas chamber, this chamber contained a second temperature sensor and a relative humidity sensor. The temperature was controlled using computer software which work alongside the existing software which controlled the solenoid valves. The ambient temperature was controlled to an accuracy of $\pm 0.1°C$.

Using the newly modified electronic nose, eight data gathering experiments were performed on cultures of a single bacteria type and three experiments were performed on cultures of a mixture of two different bacteria types. Table 5.2 summarises the experiments performed. Overall the experiments were satisfactory (i.e. there were no serious breakdowns and the planned experiments were done), however ***Streptococcus pyogenes*** did prove to be difficult to grow. The equipment proved reliable with no break downs, and no samples became infected. Thus the data gathered should be well-behaved. At this point there was a considerable library of bacteria odour data, a total of 15 data sets from the 15 data collection experiments, each data set contained 43200 data points, this was more data than had ever been collected before for one application.

Chapter 6

Data Analysis Using Novel Techniques For Odour Classification

Initial data analysis was previously performed (see chapter 4 using data from experiments performed using the original Electronic Nose (experiments 1 to 4). This chapter, using the knowledge gained from the initial exploration, details further data analysis using data from experiments performed using the modified Electronic Nose (experiments 5 to 15). The data analysis described in this chapter builds on that previously performed. Because significantly more data were gathered, more difficult classification tasks were attempted, i.e. classification of more bacteria types and identification of single types from binary mixtures.

6.1 Classification of a Single Bacteria Type

Previous classification used data-sets originating from experiments 1 to 4, which only contained information for 2 different bacteria types. The data-sets from experiments 5 to 12 contained information for 4 different bacteria types. The ability to classify 1 of 4 instead of 1 of 2 bacteria types was investigated. If MLPs were to be employed in a commercial electronic nose product that would be used in the treatment of infection

it would be advantageous to be able to discriminate correctly between as wide a range of bacteria types as possible. Also, if satisfactory results can be achieved then this would indicate MLPs as being capable of performing the necessary pattern recognition functions. Classification using an increased number of types presents a more challenging test of MLPs as classifiers.

The ANN MLP classifiers previously employed to classify 1 of 2 bacteria types were modified and re-applied to the classification of 1 of 4 bacteria types. The 4 bacteria types were ***Escherichia coli*** (collected from experiments 5 and 6), ***Staphylococcus aureus*** (collected from experiments 7 and 8), ***Pseudomonas aeruginosa*** (collected from experiments 9 and 10) and ***Streptococcus pyogenes*** (collected from experiments 11 and 12). All gas sensor feature models and normalisation algorithms were employed in turn (see the feature-set types described in table 4.1 and table 4.2). Thus a comparison of the relative performance of gas sensor feature models and normalisation algorithms was possible. For each feature-set type there were 2 feature-sets, the first (feature-set #1) contained data from experiments 5, 7, 9 and 11, and the second (feature-set #2) contained data from experiments 6, 8, 10 and 12. Thus each feature-set contained data which spanned all 4 bacteria types. Each feature-set a total of 720 feature vectors (4×180 feature vectors). It was previously shown (see tables 4.3, 4.4, 4.10 and 4.11) that reversing the training and testing feature-sets did not significantly effect classification performance, so it was decided to train the MLPs with feature-set #1 and test with feature-set #2. Splitting the training and testing data according to experiment meant that any classification would have to overcome any inter-experiment variance, just as in the eventual application. Essentially the methods used with the MLPs were identical in all practical respects to that described in chapter 4, the only change in network topology was an increase in the number of output nodes from 2 to 4. The same early stopping training algorithm was again employed. Ten MLPs were trained and tested for each feature-set type, the best performing one is subsequently quoted. The training parameters (for BP with momentum) were the same as those used previously, i.e. $\mu = 0.001$, $\alpha = 5.0$, $c = 0.1$ (flat-spot elimination constant added to derivations) and $d_{max} = 0.1$ (maximum tolerance of error per output). Table C.9 and table C.10 in

Table 6.1: Average and standard deviation of performance of bacteria type classification by means of MLPs for different gas sensor feature models.

Model	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
df	2146.62	90.53	12.50	9.98	9.13	4.54	12.50	6.73
rl	2212.84	37.25	13.61	8.25	29.10	10.02	57.29	15.35
fd	2064.23	105.26	22.26	8.38	22.95	12.64	54.69	20.27
af	2118.20	318.80	36.28	19.44	26.32	10.20	30.24	26.33
mn	1940.58	624.69	41.67	27.38	31.67	6.15	26.67	26.43
fr	2013.36	40.75	21.91	6.91	25.24	11.42	52.85	18.26
md	2103.20	185.18	22.15	15.19	27.99	13.33	52.92	28.17
mf	2012.50	96.34	34.13	9.46	26.70	11.45	39.17	9.04
ff	1952.99	72.08	31.95	21.35	25.45	10.83	42.61	31.97

appendix C summarize the results, the former table shows the results for the original feature-set types, and the later table shows the results for the new feature-set types. For interpreation of the network output, the previous '402040' rule was used, therefore a number of input vectors were unclassified. Table 6.1 and table 6.2 show, using simple descriptive statistics, the performance of different gas sensor feature models.

From table 6.1 it can be observed that MLP performance was not as good as that described in chapter 4 for the other data-set. Here the average error, SSE over the entire testing data-set, was in the order of 1900 to 2200 (as compared to 300 to 700 previously, see table 4.5). However, in this case there were twice the number of vectors in each feature-set and there were twice the number of outputs, therefore the average SSE for each output for each vector is a fairier measure. The average squared error for each output node for each input vector (individual error) was in the order of 0.66 (for 1900 SSE) to 0.76 (for 2200 SSE) compared to the previous values in the order of 0.42 (for 300 SSE) to 0.97 (for 700 SSE). The major effect of increasing the number of classes from 2 to 4 was not to increase the error for each output (for each pattern) but

Table 6.2: Minimum and maximum of performance of bacteria type classification by means of MLPs for different gas sensor feature models.

Model	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
	min	max	min	max	min	max	min	max
df	2107.89	2276.12	3.06	24.56	4.44	14.86	65.00	90.14
rl	2132.04	2212.84	4.17	23.75	19.58	38.89	39.72	74.44
fd	1944.94	2190.22	13.33	31.25	4.72	32.78	35.97	81.94
af	1827.07	2467.92	15.00	53.19	17.50	36.81	10.00	67.50
mn	1155.00	2530.37	16.11	73.61	25.56	37.50	0.83	56.67
fr	1964.32	2064.10	15.00	29.03	11.94	35.83	35.14	73.06
md	1934.30	2268.45	2.50	39.44	9.44	38.75	21.81	88.06
mf	1931.35	2136.70	26.25	47.50	12.22	36.94	29.17	50.83
ff	1850.41	2013.45	14.31	60.58	13.89	38.17	1.25	71.81

to restrict its range. This may be due partly to the fact that individual erronous input vectors within a larger feature-set (i.e. 4 classes) may influence the overall performance less than would otherwise occur in a smaller feature-set (i.e. 2 classes), i.e errors were averaged out. In a larger feature-set there may be not only more erronous vectors but more sources of error, thus total eradication of error is less probable and partial eradication more probable than for a smaller feature-set. Given that individual errors were not significantly different, the overall performance[1] for classification of 4 bacteria types appears to be worse than that for 2 bacteria types (from table 4.5). The high values for σ indicate a high degree of variance in classification performance for different normalisation (given the same gas sensor feature model). The minimum and maximum measures listed in table 6.2 indicate the best number of correct class achieved was 73.61% with a lowest SSE of 1155.

Looking at the relative performances of feature models for classification of 2 bac-

[1] Overall performance can be summarised as the number of correct classifications compared to the number of incorrect and unknown classifications.

Table 6.3: Average and standard deviation of performance of bacteria type classification by means of MLPs for different normalisation algorithms. Key to notation: n = none, s = sensor normalisation, a = auto-scaling and v = array (vector) normalisation.

Norm.	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Type	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
n	1909.70	307.70	37.33	21.45	29.32	13.08	33.35	28.45
s	2132.42	141.65	23.70	9.06	29.55	11.25	46.74	19.03
a	2224.10	189.93	17.33	13.25	17.65	9.00	61.84	23.89
v	1969.02	133.21	26.76	16.82	23.27	6.75	51.44	21.16

Table 6.4: Minimum and maximum of performance of bacteria type classification by means of MLPs for different normalisation algorithms. Key to notation: n = none, s = sensor normalisation, a = auto-scaling and v = array (vector) normalisation.

Norm.	SSE		Correct (%)		Incorrect (%)		Unknown (%)	
Types	min	max	min	max	min	max	min	max
n	1155.00	2162.88	10.83	73.61	4.72	38.89	0.83	81.94
s	1957.93	2343.88	5.97	35.83	4.44	37.50	34.03	89.58
a	1964.32	2530.37	2.50	47.50	6.81	33.33	13.47	90.14
v	1733.07	2212.84	4.17	55.14	14.86	36.39	8.47	74.44

teria types, the Absolute Final Output (af) model performs comparatively better here. The Difference (df) model, which previously performed well, now performs comparatively worse. It is possible that different gas sensor feature models perform better for discriminating particular bacteria types.

Table 6.3 and table 6.4 show, using simple descriptive statistics, the performance of different normalisation algorithms. From these tables significant differences in performance can be observed. Compared to the ranking described for classification of 2 bacteria types (see chapter 4), there were a numbers of differences. Sensor normalisation, which

was previously the best algorithm, is now third. The best classification performance was achieved with no normalisation at all. Both sensor normalisation and auto-scaling apply a different linear transformation to each component of the feature vector. Thus information which was encoded as the angles between vectors, was changed. If the feature vectors for different classes had small angles between them, the ability of the classifier to differentiate between them could be diminished by such normalisation algorithms. Vector normalisation equalises vector lengths, if vector length was related to not only odour concentration but odour quality, then again the ability for a classifier to differentiate between similar vectors of different classes could be diminished. With no normalisation, the MLP internally scales the vector components, this may be the optimum method. The potential cost of letting the MLP learn the best scaling of the input vectors in that the learning phase would be more problematic, does not seem to have been significant. The motivation for developing feature-extraction and normalisation was to make life easier when teaching a classifier. These results indicate that if there is sufficient, good quality data, it is best to implement minimal pre-processing and let the classifier do the work.

The best performance for a particular feature-set type, was achieved with the Minimum Output gas sensor feature model with no normalisation (*mnn*). This setup achieved 73.61% correct classifications, 25.56% incorrect classifications, 0.83% unknown classification and the SSE over the entire testing feature-set was 1155.

Table 6.5 shows the confusion matrix for the output of the best MLP. From this it can be seen that there were significant differences in performance for each class. ***Escherichia coli*** was correctly classified with 108 patterns (60%) accuracy, ***Staphylococcus aureus*** was correctly classified with 87 patterns (48.33%) accuracy, ***Pseudomonas aeruginosa*** was correctly classified with 155 patterns (86.11%) accuracy, and ***Streptococcus pyogenes*** was classified with 180 patterns (100%) accuracy. Also from this matrix, a measure of confidence can be derived. When the ***Escherichia coli*** class was output, it was correct with 72.97% (108 out of 148 patterns) accuracy. When the class ***Staphylococcus aureus*** was output, it was correct with 96.67% (87 out of 90 patterns) accuracy. When the ***Pseudomonas aeruginosa*** class was output, it was correct with 81.58% (155

Table 6.5: Confusion matrix for best single result for bacteria type classification, using minimum output feature model and no normalisation, using experiments 5, 7, 9 and 11 for training and experiments 6, 8, 10 and 12 for testing.

Actual Output	Target Output			
	E. coli	*Staph. aureus*	*Ps. aeruginosa*	*Strept. pyogenes*
E. coli	108	21	19	0
Staph. aureus	0	87	3	0
Ps. aeruginosa	0	35	155	0
Strept. aureus	72	32	2	180
Unknown	0	5	1	0

out of 190 patterns) accuracy. When the ***Streptococcus pyogenes*** class was output, it was correct with 62.94% (180 out of 286 patterns) accuracy. Therefore, although ***Streptococcus pyogenes*** was correctly identified 100% of the time, when it was output by the MLP it was only correct 62.94% of the time. The reverse was true for ***Staphylococcus aureus***. Confidence is important: when class A is output, how reliable is the answer? Some indication of 'likeness' can also be observed from the matrix, the MLP mistook ***Escherichia coli*** for ***Streptococcus pyogenes*** in 40.00% (72 patterns) of cases. Also there was significant confusion between ***Staphylococcus aureus*** and the other bacteria types.

The difference in performance between classification of 2 bacteria types and of 4 bacteria types may be due to the increase in classifier complexity, therefore it is necessary to look at the significance of the number of output classes upon performance. If one wanted to achieve the highest number of correct classifications without designing a classifer, then all vectors would be assigned as belonging to the same class; or a random assignment. Therefore, for 2 classes, the number of correct classifications would be 50% (the number of incorrect classifications would be 50%), and for 4 classes the number of correct classifications would be 25% (the number of incorrect classifications would be 75%). The larger the difference between correct and incorrect classifiction, the better the classifier. What was desired was classification performance that was significantly

better than could be achieved by constant or random classification. For 2 bacteria types, the best difference was 93.89% (the 1.67% of unclassified patterns was considered not to be significant). For 4 bacteria types, the best difference was 48.05%. Both of these are 100% better than would be expected for constant or random classification. So given the increase in complexity of the problem, the decrease in performance is no greater than that which would be expected for identical classifier designs.

6.2 Dynamic Gas Sensor Feature Models

All classification so far described in this thesis has employed static feature models. These feature models derive features from parameters which are measurements of static quantities in gas sensor responses to odours (see section 4.1.1 in chapter 4). It was briefly mentioned in the review (see chapter 2) that previous work had been published detailing the use of dynamic features [54]. Included in the diagram which illustrated a 'smell', Figure 4.1, were several parameters which relate to timing (temporal) information. These were used here to derive dynamic features.

Taking a resitive single gas sensor, its associated output voltage at time t after being exposed to a step-input function (for odour concentration) can be expressed as $V(t)$. Conductance, $G(t)$ is therefore $kV(t)^{-1}$ (where k is a constant). The transient response of the conductance of a tin oxide gas sensor [115] has been shown to approximate an exponential rule[2]:

$$G(t) = G_0 + \Delta G(1 - exp(\frac{-t}{\tau})) \tag{6.1}$$

Where G_0 is the initial conductance (also called baseline conductance) at time $t = 0$, and τ is the time constant for the gas sensor. As $t \to \infty$, the conductance, G tends towards G_f, which is the final, (steady-state) value and $\Delta G = G_f - G_0$. Thus the rate of change of output with respect to time can be defined:

$$\frac{dG}{dt} = \frac{\Delta G}{\tau} exp(\frac{-t}{\tau}) \tag{6.2}$$

[2] A bi-exponential rule is even better

The derivative $\frac{dG}{dt}$, is related to both the type and concentration of the odour through ΔG as in the static model. From the 'raw' data file, ΔG and τ were extracted for each sensor response for each 'smell' cycle. This transient information was used to construct a new feature, the Transient model (*tr*):

$$x = \left.\frac{dG}{dt}\right|_{t=\tau} \tag{6.3}$$

Making $t = \tau$, the first derivative simplifies to:

$$\frac{dG}{dt} = \frac{\Delta G}{\tau} exp(-1) \simeq \frac{\Delta G}{2.7\tau} \tag{6.4}$$

Thus the derivative contains not only the static response, ΔG, but also the reaction kinetics through τ. It is known that the value of τ depends upon the analyte type and concentration too. The advantage of using the derivative is that baseline variance is removed, like biological systems. From this new model, and employing all types of normalisation, new feature-sets were formed. These feature-sets were applied to the classification of 1 of 4 bacteria types.

6.2.1 Multiple Layer Perceptron

The best feature model so far used was the minimum output feature model (*mn*) (see table 6.5). If the transient feature model (*tr*) contains discriminatory information that had previously been excluded in the static feature models, its inclusion should improve classifier performance. New feature-sets were formed from a combination of the *mn* and *tr* feature-sets, these were identified using *tm* notation. Two sets of experiments were performed, the first trained and tested MLPs using the *tm* feature-set (i.e. combined static and dynamic features), and the second using the *tr* feature-set (i.e. only dynamic features). The purpose of performing the second experiment was to provide a control by which the effect of dynamic features could be gauged, for example, would better performance be achieved by using dynamic features alone? In these experiments, as before, same training and testing methods were used. For each feature-set type there were 2 feature-sets, the first (feature-set #1) contained data from experiments 5, 7, 9 and 11, and the second (feature-set #2) contained data from experiments 6, 8, 10

Table 6.6: Performance of bacteria type classification by means of MLPs using combined Minimum Output and Transient feature models.

Feature-set	SSE	Correct (%)	Incorrect (%)	Unknown (%)
tmn	911.92	82.22	14.44	3.33
tms	2193.68	13.19	11.39	75.42
tma	2367.50	13.06	11.25	75.69
tmv	1703.59	35.58	18.06	46.39

and 12. The MLPs were trained using feature-set #1 and tested using feature-set #2. By adopting the same training and testing methods, useful comparisons can be made between the performance of different MLPS, i.e. any difference would not be due to experimental methods but the MLPs themselves.

Combined Static and Dynamic Features

For this experiment, MLP topology was 18 inputs feeding into 20 hidden nodes, feeding into 4 output nodes. All four normalisation methods were investigated. Table 6.6 lists the results obtained for different normalisation methods. The best single result was with no normalisation which achieved 82.22% correct classifications, 14.44% incorrect classifications and 3.33% unknown classifications (and with a SSE of 911.92 over the entire test feature-set). For 4 bacteria types, the previous best single performance was achieved using the *mnn* feature-set with 73.61% correct classifications. The combined feature-set improved this figure by 8.61%.

The confusion matrix shown in Figure 6.7 shows the best individual MLP performance. ***Escherichia coli*** was classified with 96.11% accuracy (173 patterns), ***Staphylococcus aureus*** was classified with 77.78% accuracy (140 patterns), ***Pseudomonas aeruginosa*** was classified with 59.44% accuracy (107 patterns) and ***Streptococcus pyogenes*** was classified with 95.56% accuracy (172 patterns). Confidence was highest for ***Pseudomonas aeruginosa*** at 98.17%, followed by ***Escherichia coli*** at 86.93%, ***Streptococcus pyogenes*** at 81.90% and ***Staphylococcus aureus*** at 78.21%. Therefore ***Escherichia***

Table 6.7: Confusion matrix for bacteria type classification, using combined Minimum Output and Transient feature models and no normalisation, using experiments 5, 7, 9 and 11 for training and experiments 6, 8, 10 and 12 For testing.

Actual Output	Target Output			
	E. coli	*Staph. aureus*	*Ps. aeruginosa*	*Strept. pyogenes*
E. coli	173	4	16	6
Staph. aureus	0	140	39	0
Ps. aeruginosa	2	0	107	0
Strept. pyogenes	4	32	2	172
Unknown	1	4	16	3

coli was most easily classified with a high accuracy and confidence.

The combined feature-set has 3 features for each gas sensor, giving a total of 18 inputs (for 6 gas sensors), and previous feature-set types yielded 1 or 2 features per gas sensor (giving 6 or 12 inputs). The increase in the number of inputs increases the number of weights in the network. The error surface of a network with a large number of weights is more likely to have a large number of local minima than an error surface of a network with a similar number of weights. In theory, the MLPs so far employed with 6 or 12 inputs (230 or 356 weights[3]) should be at an advantage compared to the MLP which used the combined feature-set which had 18 inputs (482 weights). Yet the results so far obtained indicate that the more complex MLPs have performed the best, *mn* feature-set and *tm* feature-set.

Dynamic Features

For this experiment, MLP topology was 6 inputs feeding into 20 hidden units, feeding into 4 output units. Again, all four different normalisation methods were investigated. Table 6.8 lists the results obtained for different normalisation methods.

The results were significantly worse than those for the previous experiment, the best

[3]The number of weights is for 20 hidden and 4 output nodes and includes thresholds.

Table 6.8: Performance of bacteria type classification by means of MLPs using combined Minimum Output and Transient feature models.

Feature-set	SSE	Correct (%)	Incorrect (%)	Unknown (%)
tmn	2169.39	18.06	28.97	53.47
tms	2141.95	24.72	32.50	42.78
tma	2191.03	24.58	11.94	63.47
tmv	2093.99	22.08	35.69	42.22

percentage of correct classification for this experiment was 24.72%, compared to 82.22% for the previous experiment. The SSE for this experiment was consistently high, around 2100, this indicates that the networks did not converge well, this is also indicated by the large percentage of patterns that were unknown (from 42.22% to 63.47%). Using static features, the best performance achieved was 73.61% (see section 6.1). Dynamic information alone resulted in worse performance than static, and dynamic combined with static.

It can be concluded from this that discriminatory information is contained in both static and dynamic features (possibly more in static features than dynamic) and that improved classification can be achieved by a combination of the two rather than each on their own.

6.2.2 Discriminant Function Analysis

Linear Discriminant Function Analysis (DFA) has been applied (see section 4.5 in chapter 4) to classify bacteria type and to predict growth phase. Compared to MLPs its performance was found to be inferior, e.g. 67.50% correct classification of bacteria type compared to 96.11% for MLPs. DFA is an established statistical classification method that serves here as a benchmark. The relative performance of MLPs can be better judged when comparing their performance to that of a technique like DFA. The method of implementation here was identical to that used previously. The feature-sets used were that feature-set pair which was reported as being the best for MLPs

Table 6.9: Classification function coefficients calculated using discriminant function analysis for bacteria type (using *tmn* feature-sets).

	Bacteria Type			
Coefficient	*E. coli*	*Staph. aureus*	*Ps. aeruginosa*	*Strept. pyogenes*
c_0	-391.18	-606.09	-678.63	-502.75
c_1	-48.34	-40.08	-43.11	-53.37
c_2	-30.98	-31.61	-33.28	-32.53
c_3	-17.69	-6.14	-8.83	-9.99
c_4	-2.41	-18.19	-4.02	-23.70
c_5	39.81	19.88	12.58	12.59
c_6	-4.17	-7.48	-5.16	0.36
c_7	-26.02	-48.55	-59.95	-28.35
c_8	-26.66	-35.68	-50.04	-25.47
c_9	1.27	17.40	14.95	15.78
c_{10}	100.75	139.84	134.46	133.45
c_{11}	103.18	142.29	145.81	127.96
c_{12}	39.68	45.19	48.29	31.97
c_{13}	31.01	31.54	48.56	18.25
c_{14}	-0.33	-7.52	-13.30	-1.02
c_{15}	-5.25	-19.92	-19.66	12.98
c_{16}	10.51	15.83	13.11	25.36
c_{17}	-28.62	-24.43	-4.76	-16.26
c_{18}	-27.62	-30.90	-30.68	-23.75

so far, i.e. Combined Minimum Output and Transient gas sensor feature models with no normalisation (*tmn*). The discriminant functions (DFs) and classification functions (CFs) were calculated using data from experiments 5, 7, 9 and 11 and were tested using data from experiments 6, 8, 10 and 12. Table 6.9 shows the results of DFA, the coefficients of the CFs are listed.

From the table, coefficients c_{10} and c_{11} are dominant. These coefficients correspond to gas sensor 4. Previously, when bacteria type was classified using DFA, gas sensors 2 and 4 were found to yield dominant coefficients. Here, gas sensor 2 (coefficients c_4, c_5 and c_6) does not appear to be dominant, there is commonality with respect to gas sensor 4. This may indicate that gas sensor 4 (FIS P.A.2, polar compounds) is the most significant.

Since Wilk's Lambda is a measurement of variance not involved in classification, in this case the value measured for Wilk's Lambda was 0.001, thus unwanted variance was negligible. Table 6.10 shows the corresponding Mahalanobis and Fisher's distance measurements resulting from the DFA. From this table an appreciation of the similarity of classes can be gained, similar classes have small distances between each other, and dissimilar classes have larger distances. The largest distance is that between ***Pseudomonas aeruginosa*** and ***Escherichia coli*** (D^2=1001.82 and F=210.38), the smallest distance is that between ***Staphylococcus aureus*** and ***Pseudomonas aeruginosa.*** This was reflected in the results, ***Pseudomonas aeruginosa*** was not misclassified as ***Escherichia coli***, and vice versa for all test vectors. But, ***Staphylococcus aureus*** was misclassified as ***Pseudomonas aeruginosa*** (and vice versa) in 93 cases (out of 374 total misclassifications). This can be compared with the output from the MLP using the confusion matrix in Figure 6.7, the trend with regard to these two class pairs was repeated, with 18 misclassified vectors for the former par of classes, and 39 for the later.

Reclassification of the training feature set was 99.58% correct (717 correct out of 720 total), testing using the test feature-set yielded a classification performance of 48.06% correct (346 correct out of 720 total). This result was considerably worse than classification using a MLP (82.22%). It is also worse than that for the previous classification of 2 bacteria types, which was 65.83% (this is to be expected because of the increase of complexity).

Figure 6.1 shows a plot of the test feature-set against the first two ranked discriminant functions (DFs). The target class membership is indicated by plot colour (see figure caption). ***Streptococcus pyogenes*** showed the best clustering, ***Escherichia coli*** and ***Staphylococcus aureus*** show clusterings that have significant overlap. The first two

Table 6.10: Mahalanobis, D^2, and Fisher's F distances between class centroids calculated using discriminant function analysis for bacteria type (using *tmn* feature-sets).

		Escherichia coli	*Staphylococcus aureus*	*Pseudomonas aeruginosa*	*Streptococcus pyogenes*
Escherichia coli	D^2	0	191.76	210.38	107.34
	F	0	913.17	1001.82	511.17
Staphylococcus aureus	D^2	191.76	0	31.97	55.13
	F	931.17	0	152.27	262.54
Pseudomonas aeruginosa	D^2	210.38	31.97	0	55.13
	F	1001.82	152.27	0	515.82
Streptococcus pyogenes	D^2	107.34	55.13	108.31	0
	F	511.17	262.54	515.82	0

DFs described 96% of the variance of the data (the third DF described 4% of the variance of the data). Misclassification between *Escherichia coli* and *Staphylococcus aureus* occurred in 66 cases (out of 374 total misclassifications), this relatively low number indicates that the other DFs reduced the overall cluster overlap. The plot in Figure 4.9 is still however a good indication of the quality of the DFs calculated.

6.3 Culture Growth Phase Compensation Using Fuzzy Sets

It has been shown in chapter 4 that culture growth phase influences the odours released from bacteria cultures. Also the possibility of implementing a two stage classifier was discussed, where a separate classifier identified the culture growth phase, the output from this stage is then input to the classifier in order to aid bacteria type classification. Before this can happen, growth phase needs to be predicted in a more accurate manner in order to reduce error within the classifier system. The application of fuzzy set theory provided an opportunity to achieve this.

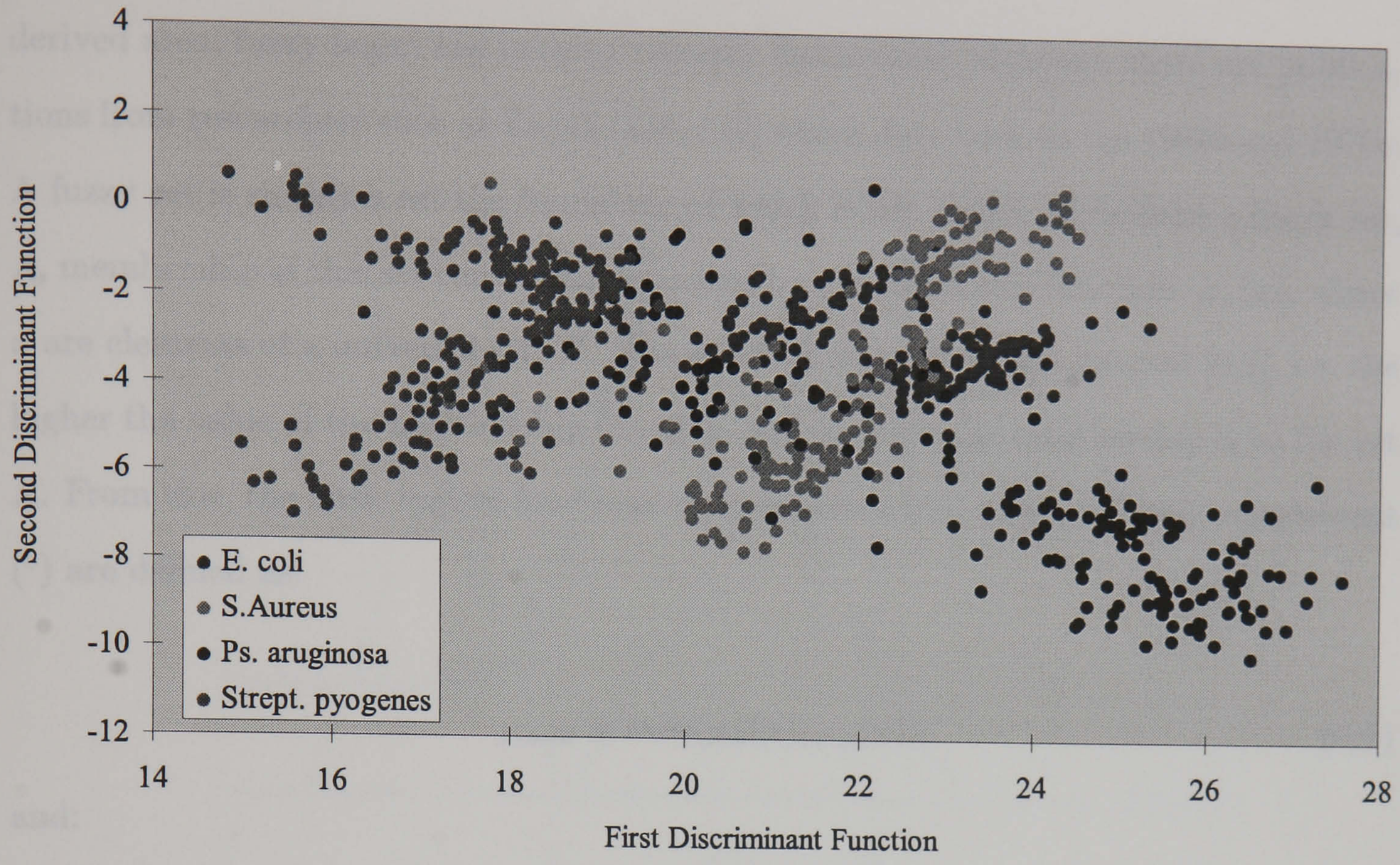


Figure 6.1: Results of discriminant function analysis of bacteria type (using *tmn* feature-sets).

6.3.1 Fuzzy Set Theory

The boundaries between the different growth phases are not 'hard' but 'soft'. What is meant by this is that the transition from one growth phase to the next is not instantaneous but gradual. Previously in chapter 4, growth phase was predicted based upon the assumption that at a given time, the bacteria culture was exclusively in one of the growth phases. It was found that the majority of errors occurred in the boundary regions of growth phase, this was due to the implementation of 'hard' boundaries. If the boundaries were implemented as 'soft', which is closer to the reality of culture growth, better predication should result. There is no single point in time when all the cells within a culture spontaneously changed their behaviour, in actuality, the transition has taken place when the majority of cells in the culture behaved in a similar manner. Fuzzy set theory provides an technique for encoding these 'soft' boundaries.

Fuzzy set theory originated from control engineering in which fuzzy controllers use fuzzy set theory and have been applied to many problems. Fuzzy set theory (and its

derived area, fuzzy logic) has become recently fashionable, although there are publications from researchers such as Zadeh [116, 117] which date back to the 1960s and 1970s. A fuzzy set is simply a set the boundary of which is not sharp. If we have a fuzzy set, A, membership of this set can be characterised by a membership function, $\mu_A(x)$, where x are elements of a universal set, X. The set X is mapped to the interval [0,1], i.e. the higher the value of the membership function, the stronger the membership is to the set A. From this, the basic logical functions of intersection ($\cap$), union ($\cup$) and complement (^{c}) are defined as:

$$\mu_{A\cap B} = min(\mu_A(x), \mu_B(x)) \tag{6.5}$$

and:

$$\mu_{A\cup B} = max(\mu_A(x), \mu_B(x)) \tag{6.6}$$

and:

$$\mu_{A^c} = 1 - \mu_A(x) \tag{6.7}$$

In our case we have three sets describing growth phase[4], A for the lag phase, B for the log phase, and C for the static (stationary) phase. The elements, x, correspond to the elapsed time during which the culture in question has been growing. If 'classical' set theory were employed, for a given value of x, the culture would be deemed to be either fully in one or another class (assuming no overlap occurs between sets), i.e. A, B or C. Therefore thresholds exist in the domain of x where the culture growth phase changes, for example from A to B, this is what is meant by the term 'hard' boundary. If these sets are fuzzy, then the determination of growth phase becomes different. Figure 6.2 shows pictorially how fuzzy sets were applied to a typical culture growth curve.

The different membership functions in this diagram are indicated by different colours (see caption for key). The membership functions are basically triangular, the first and

[4]The fourth growth phase, i.e. death phase, is not considered here because no culture was used for long enough to enter this phase.

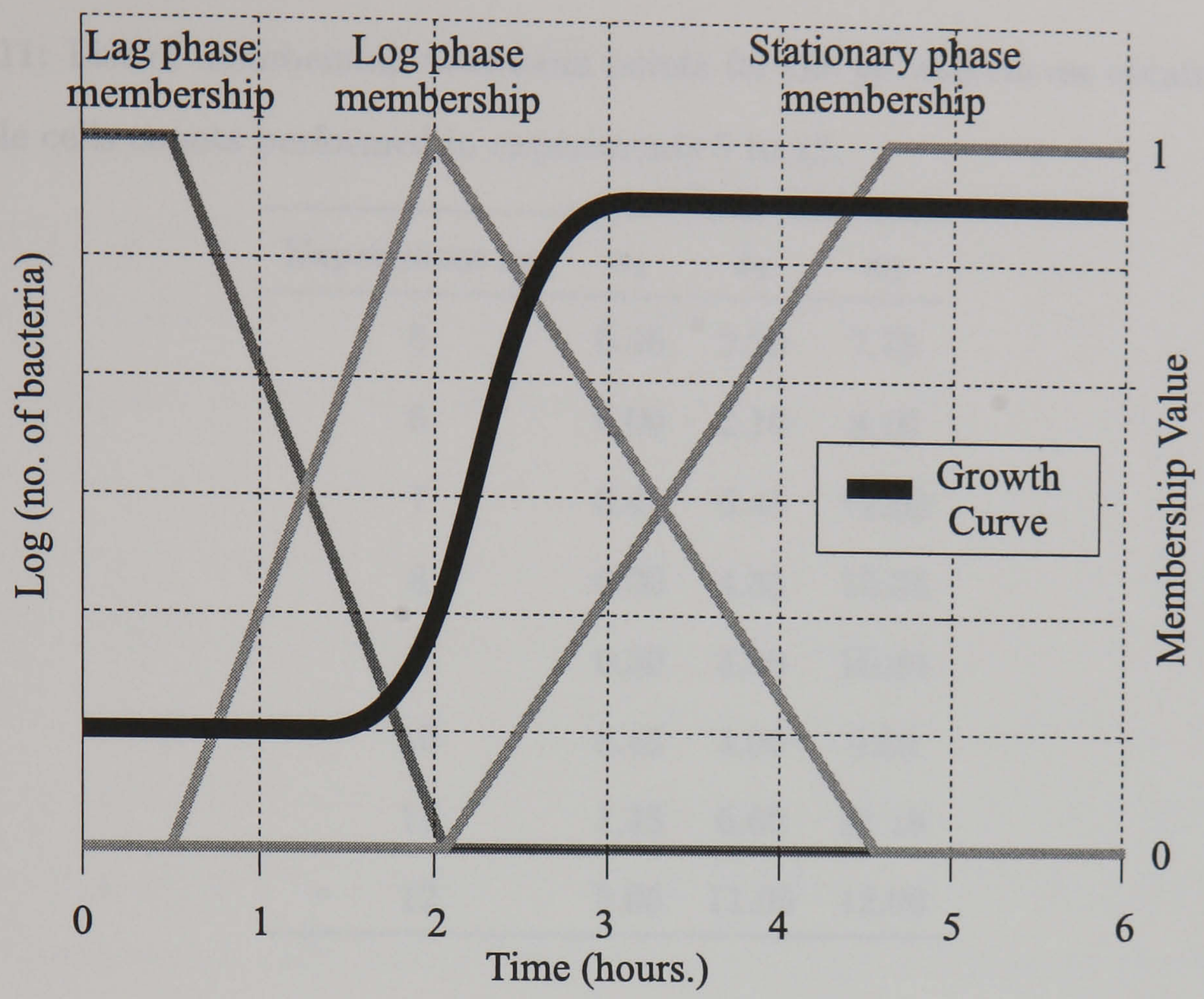


Figure 6.2: Fuzzy growth phase membership functions related to a typical culture growth curve. Key: Blue - lag phase membership ($\mu_A(x)$), Pink - log phase membership ($\mu_B(x)$), Green - static phase membership ($\mu_C(x)$).

last functions extend to the minimum and maximum values of x, respectively; the sum of value of all memberships is unity for all values of x. These functions are the simplest that can be applied and are popular in many applications. It can also be noticed that the point of equal membership between two adjacent functions is the the original 'hard' boundary, and the maximum is approximatley midway in the respective growth phase. For each growth curve, three values of x were specified, x_1, x_2 and x_3, which corresponded the the points of maxmimum membership of the three membership functions, respectively. Maximum membership was at the midpoint in each phase. Thus for any value of x, $\mu_A(x)$ can be defined as:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x < x_1 \\ 1 - \frac{x - x_1}{x_2 - x_1} & \text{if } x_1 \leq x < x_2 \\ 0 & \text{if } x_2 \leq x \end{cases} \tag{6.8}$$

Table 6.11: List of membership maximum points for the growth curves obtained from the viable cells counts performed in experiments 5 to 12.

Experiment no.	x_1	x_2	x_3
5	0.45	2.20	7.75
6	0.00	2.10	8.10
7	0.45	6.45	12.00
8	0.00	4.35	10.35
9	0.50	4.90	10.40
10	0.45	4.05	9.60
11	1.45	6.60	11.15
12	5.05	11.05	12.00

The membership function $\mu_B(x)$ can be similarly defined as:

$$\mu_B(x) = \begin{cases} \frac{x-x_1}{x_2-x_1} & \text{if } x_1 \leq x < x_2 \\ 1 - \frac{x-x_2}{x_3-x_2} & \text{if } x_2 \leq x < x_3 \\ 0 & \text{if } x < x_1 \text{ or } x_3 \leq x \end{cases} \tag{6.9}$$

And the membership function $\mu_C(x)$ can be defined as:

$$\mu_C(x) = \begin{cases} 0 & \text{if } x < x_2 \\ \frac{x-x_2}{x_3-x_2} & \text{if } x_2 \leq x < x_3 \\ 1 & \text{if } x_3 \leq x \end{cases} \tag{6.10}$$

Table 6.11 lists the values of x_1, x_2 and x_3 for the growth curves obtained from experiments 5 to 12. These points were determined by inspection of the growth plots which are shown in chapter 5.

For any single value of x, allocation to a growth phase is now described by the three variables: $\mu_A(x)$, $\mu_B(x)$ and $\mu_C(x)$.

6.3.2 Fuzzification Using Multiple Layer Perceptrons

The question to 'ask' the classifier is not which growth phase is the culture currently in, but which growth phase describes the majority of cells in the culture. From this idea, a set of MLPs were constructed, trained and tested. The gas sensor feature model and normalisation algorithms that resulted in the best performance previously (i.e. Absolute Final Output Feature Model and Auto-Scaling) were re-applied to the new data to form new feature-sets. The training feature-set and testing feature-set were formed from different experiments as described in section 6.1. The methods of training and testing were the also the same as those applied previously to MLPs (i.e. the same training algorithm, the same learning parameters and the same validation techniques). The network topology was 12 inputs feeding into 20 hidden nodes feeding into 3 output nodes. Also the feature-sets were re-sampled in order to equalise class membership, this technique was also applied previously in chapter 4. Figure 6.3 shows the output from a MLP to a set of vectors in the testing feature-set which correspond to experiment 8 (*Staphylococcus aureus*), vessel #3[5]. The plots are shown in order to illustrate the behaviour of a MLP which was trained on fuzzy outputs as opposed to 'hard' outputs.

Over the entire testing feature-set (720 vectors), the SSE was 214.97. Therefore for each vector the average SSE was 0.3, so the average error was 0.32 per output. This compares to minimum SSE of 423.38 (360 vectors), or 0.63 per output, that was previously achieved in chapter 4. It can be seen from comparing the target outputs and the actual outputs in Figure 6.3, that the MLP was successful. Because the output was fuzzy membership, an algorithm such as the '402040' methods which has been used so far, cannot be applied in this case. This MLP is not envisaged as operating on its own, but as part of a larger classifier system. Its performance can therefore be measured by the relative performance of the larger classifier system.

[5]The entire output cannot be shown in this form because the high number of data points would make the plots unreadable

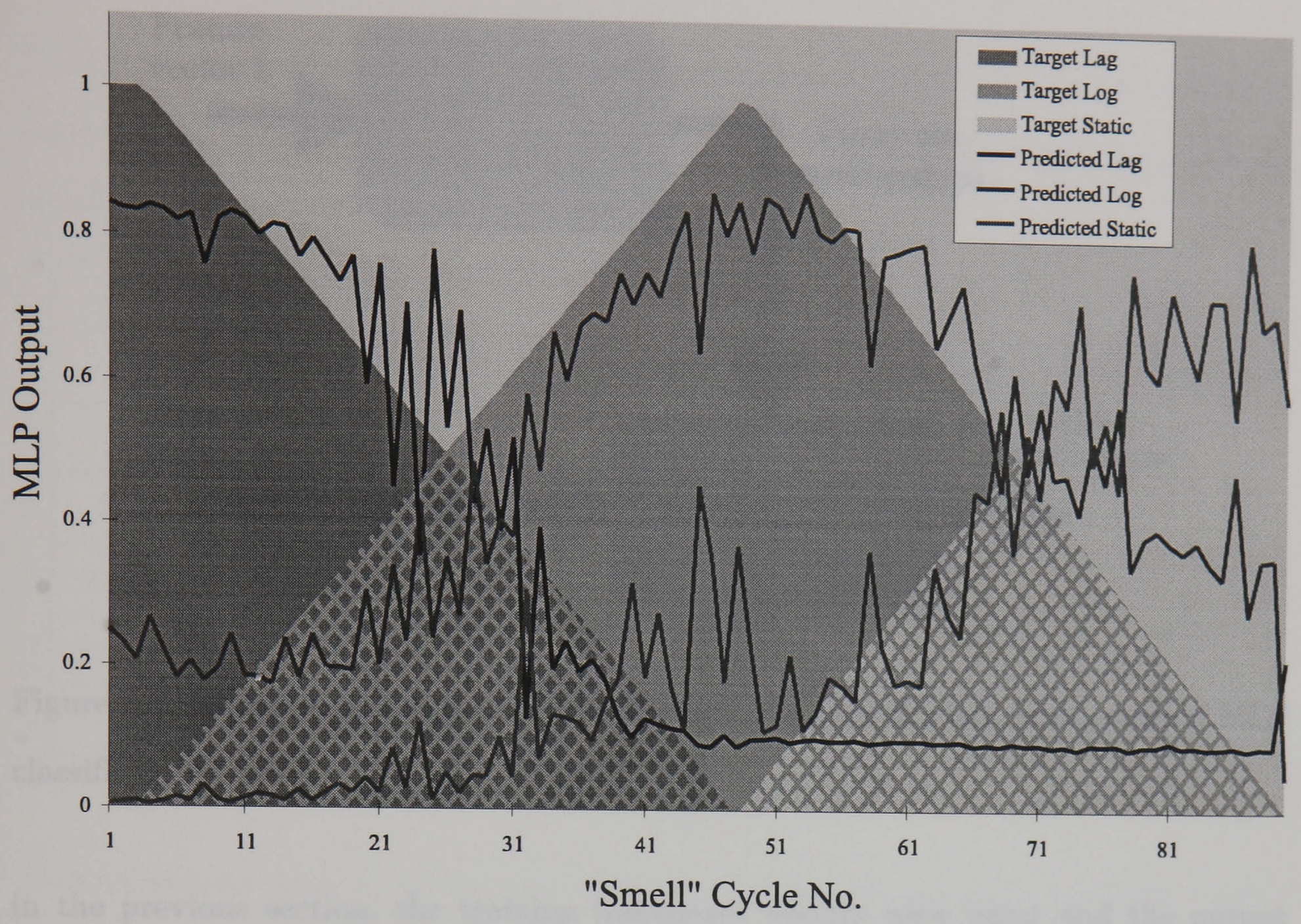


Figure 6.3: An example of the results of predicting growth phase membership showing both the actual output (lines) and target output (shaded areas) of a MLP for experiment 8, vessel #3.

6.3.3 Implementation of Growth Phase Compensation

The MLP design described in the previous section was combined with the MLP design described in section 6.2.1. Figure 6.4 illustrates how the compensated MLP was constructed. There are two sets of inputs because optimum performance of phase prediction and type classification occurred using different gas sensor feature models and normalisation algorithms[6].

The method of training the network was to train the phase predictor and then train the type classifier separately[7]. Firstly, the phase detector was trained, this is described

[6] If this wasn't the case it would have been better to employ a single network which internally learnt phase compensation.

[7] Both MLPs could have been trained and tested simultaneously but the implementation of this in the simulation software would have been very time-consuming. The implementation of separate MLPs allowed more rapid development.

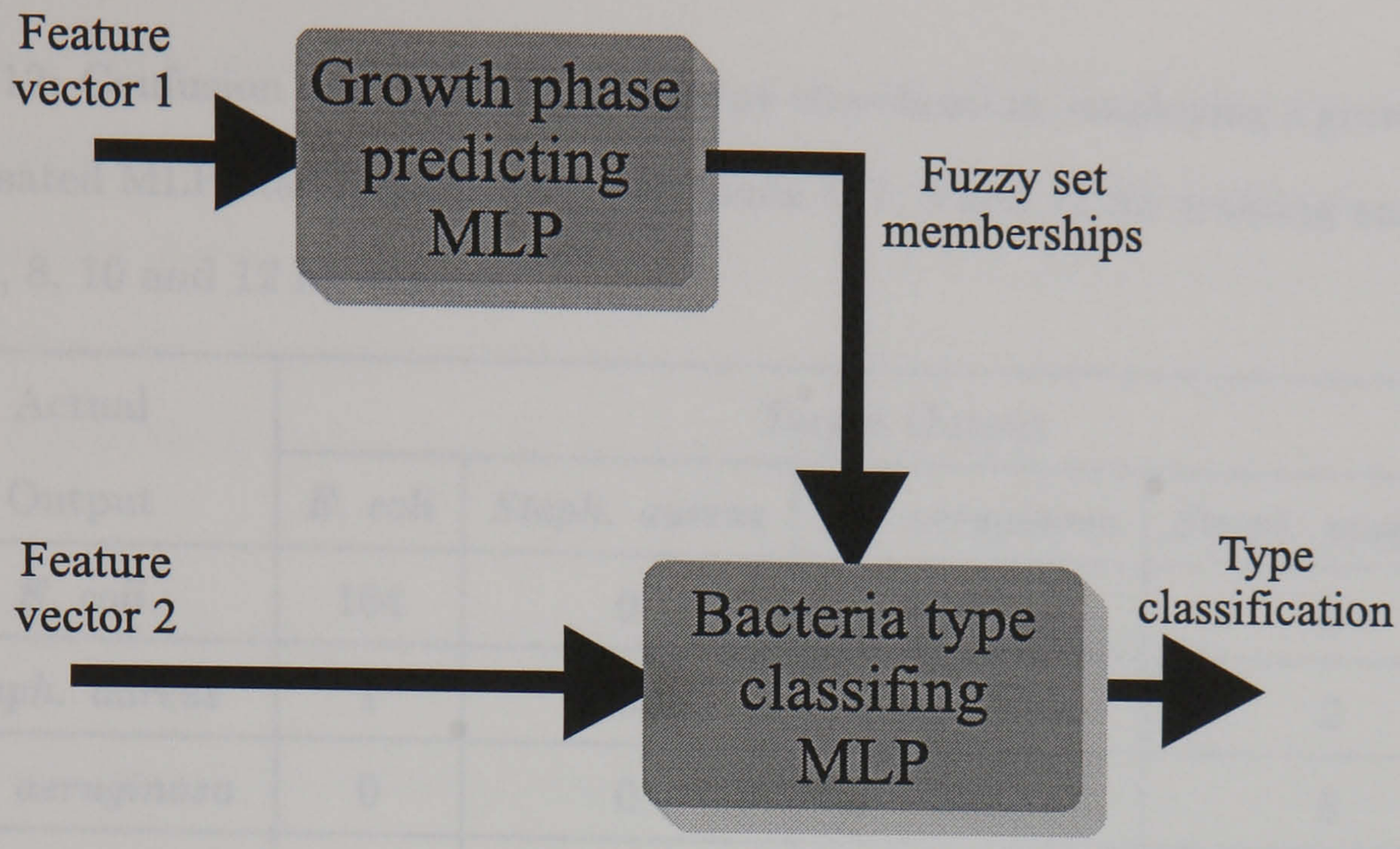


Figure 6.4: Illustration showing the construction of a growth phase compensated MLP classifier.

in the previous section, the training feature-set vectors were input and the output vectors recorded. The recorded output was then combined with the training feature-set (for the type classifier) to produce a new training feature-set which contained all the inputs to the type classifier. Similarly, the testing feature-set for the phase detector was input to the phase detector and the output recorded. This recorded output was combined with the testing feature-set (for the type classifier) to produce a new testing feature-set than contained all the inputs to the type classifier. Once the new training and testing feature sets were created, the type classifier was trained. The training, testing and validation methods employed were the same as those used previously. The network topology was 21 inputs feeding 20 hidden units feeding 4 output units. The performance obtained by re-classifying unknown patterns from the testing feature-set was 92.78% correct (668 patterns), 5.69% incorrect (41 patterns), 1.53% unknown (11 patterns) and a SSE of the entire testing feature-set of 257.34. The classification results are summarised in confusion matrix in Table 6.12.

The performance of this MLP design was better than the previous best of 82.33% correct classification which was achieved using combined static and dynamic features. The SSE measurement also showed a considerable drop, from 911.92 to 257.34, this

Table 6.12: Confusion matrix for bacteria type classification, employing a growth phase compensated MLP classifier, using experiments 5, 7, 9 and 11 for training and experiments 6, 8, 10 and 12 for testing.

Actual Output	Target Output			
	E. coli	*Staph. aureus*	*Ps. aeruginosa*	*Strept. pyogenes*
E. coli	164	0	0	0
Staph. aureus	1	163	1	2
Ps. aeruginosa	0	0	177	5
Strept. pyogenes	15	17	0	164
Unknown	0	0	2	9

gives an average error of 0.3 for each output for each pattern (compared to 0.56 previously). This highlights that the MLP reached a lower global minimum during training, i.e. this MLP was better able to construct an internal representation which satisfactorily fitted the data. Comparing their respective confusion matrices (i.e. Figure 6.12 and Figure 4.16) it can be noted that while the performance of the MLP overall has increased, the performance for individual classes has both increased and decreased. The number of correct classifications for ***Escherichia coli*** had dropped from 173 to 164, and those for ***Streptococcus pyogenes*** had dropped from 172 to 164. However, these reductions were less than the increases in the number of correct classifications for ***Staphylococcus aureus***, which increased from 140 to 163, and those for ***Pseudomonas aeruginosa***, which increased from 107 to 177. There was a considerable drop in the number of unknown patterns, from 24 to 11, this was a result of the lower minimum reached during training (confirmed by the lower SSE measurement). The incorrectly classified patterns were probably outliers, the better global minimum converged upon during training meant that the majority of patterns were classified at the expense of a small number of outlying patterns. This explains why the number of correct classifications for some types actually reduced. It is virtually impossible to accommodate all outlying patterns without over-fitting the data and reducing generalisation performance.

Confidence levels were 100% for *Escherichia coli*, 97.60% for *Staphylococcus aureus*, 97.25% for *Pseudomonas aeruginosa*, and 83.67% for *Streptococcus pyogenes*. The low confidence for *Streptococcus pyogenes* indicates that the largest number of incorrect classifications was *Staphylococcus aureus* misclassified as *Streptococcus pyogenes*. This may be due to their similar nature, i.e. both gram +ve and non-motile.

6.4 Classification of Multiple Bacteria Types

There were three data collection experiments performed where mixtures of bacteria types were used. Experiment 13 employed *Escherichia coli* and *Staphylococcus aureus* mixture, experiment 14 employed *Pseudomonas aeruginosa* and *Staphylococcus aureus* mixture, and experiment 15 employed *Escherichia coli* and *Pseudomonas aeruginosa* mixture. Proportions of these mixture changes dynamically during the experiment with some bacteria types growing faster than others. Therefore the growth phase compensation that was described in the previous section cannot be applied here because each bacteria type may be in a different growth phase at any one time. From this, the static and dynamic feature models discussed in section 6.2.1 were used (also no normalisation was used). Each data gathering experiment yielded 180 vectors (90 per sample vessel), from all three experiments there were a total of 540 vectors. These vectors were split-up to provide two sets, a training vector set and a vector testing set. The training vector set contained all the vectors from vessel #2 (i.e. the first sample), the testing vector set contained all the vectors from vessel #3 (i.e. the second sample). Thus each vector set contained 270 vectors. Once the training and testing vector sets were formed, they were subject to feature extraction using the Minimum Output gas sensor feature model and the Transient gas sensor feature model, therefore combined data sets were formed containing the output from two feature models. Thus a training feature-set and a testing feature-set were formed. There were 3 features for each gas sensor, making a total of 18 inputs. Since there were only three different bacteria types used in these experiments, there were 3 outputs (one for each type). Ideally, for each input pattern, two outputs should be activated. The MLP topology was 18 inputs

Table 6.13: Table summarising the results of bacteria type classification from a mixture of two different types for each output (i.e. class), using experiments 13, 14 and 15. sample 1 (Vessel #2) for training and sample 2 (vessel #3) for testing.

Type	Correct (%)	Incorrect (%)	Unknown (%)
Escherichia coli	161 (59.60%)	76 (28.15%)	33 (12.22%)
Staphylococcus aureus	260 (96.30%)	4 (1.48%)	6 (2.22%)
Pseudomonas aeruginosa	171 (63.33%)	25 (9.26%)	74 (27.40%)

feeding 20 hidden nodes feeding 3 output nodes. The training and testing techniques used were the same as those used previously.

Because more than one output is intended to be activated at any one time (2 of 3 instead of 1 of 3), the results were interpreted for each output, i.e. each bacteria type. From this, a modified version of the '402040' rule (see section 4.2.1 in chapter 4) was used for interpretation of the outputs. If the output from a node was in the upper band and the target output was high, then the output was deemed as correct. If the output was between the upper and lower bands, then the output was deemed as unknown. Otherwise, the output was deemed incorrect. Therefore for each output there was a % correct, % incorrect and a % unknown measure. Table 6.13 summarises the results obtained. The SSE measured over the entire testing feature-set was 155.12.

A meaningful confusion matrix cannot be constructed because incorrect classifications occurred when more than one other class was the target output. However useful information can be gained from Table 6.13. In terms of percentages, ***Escherichia coli*** was correctly identified in 59.6% of all cases, ***Staphylococcus aureus*** was correct in 96.30% of all cases and ***Pseudomonas aeruginosa*** was correct in 63.33% of all cases.

For the ***Escherichia coli*** and ***Staphylococcus aureus*** mixture (experiment 13), both types were simultaneously, correctly identified in 51.11% of all cases. More specifically for these 90 patterns, ***Escherichia coli*** was identified output for 65 patterns (72.22%) and ***Staphylococcus aureus*** was correctly identified for 89 patterns (98.89%). Most of the errors for ***Escherichia coli*** occurred in the 11 patterns (i.e. during the first 88

minutes of data collection). This result was unexpected because ***Escherichia coli*** was the dominant bacteria in this mixture (see Figures 5.14 and 5.15). Also ***Escherichia coli*** produces stronger odours than ***Staphylococcus aureus***, given equal concentrations. It is possible that there was an odorant from ***Staphylococcus aureus*** which was not present in the odour from ***Escherichia coli***, but which the sensor array was sensitive to. This is possible because the two types of bacteria are very different (for example different Gram types) and metabolise nutrients very differently.

For the ***Pseudomonas aeruginosa*** and ***Staphylococcus aureus*** mixture (experiment 14), both types were simultaneously identified in 28.89% of all cases. More specifically, ***Staphylococcus aureus*** was correctly identified for all 90 patterns (100%), but ***Pseudomonas aeruginosa*** was correctly identified for only 16 patterns (17.78%). For 66 patterns, only ***Staphylococcus aureus*** was indicated as being present. This is unexpected because of all the bacteria types used, ***Pseudomonas aeruginosa*** produces the strongest odours.

For the ***Escherichia coli*** and ***Pseudomonas aeruginosa*** mixture (experiment 15), both types were simultaneously identified in 91.11% of all cases. ***Escherichia coli*** was identified in 87 patterns (96.67%) and ***Pseudomonas aeruginosa*** was identified in 89 patterns (98.89%). The weights in the MLP for the paths to the ***Escherichia coli*** and ***Pseudomonas aeruginosa*** were very similar, basically one output was a duplicate of the other, thus errors occurred when only one of these types was present in the mixture.

From all performance measurements so far described, the behaviour of the MLP was almost opposite to what was expected, i.e. that the presence of ***Staphylococcus aureus*** masked out the other two types and caused errors for those types. In reality, ***Staphylococcus aureus*** produces the weakest odour and therefore should have been masked out by the presence of the other two types. The greatest amount of similarity was between ***Escherichia coli*** and ***Pseudomonas aeruginosa*** (i.e. both Gram -ve and motile). Although lower in concentration, the MLP was able to more easily distinguish ***Staphylococcus aureus*** from the other types because of its different nature. These results support the theory that odours at different concentrations are not as big a problem as odour similarity for identification of single types from a mixture. Because

Staphylococcus aureus gave off a very different odour from the other two, it came to dominate the MLP, rather than the other bacteria types whose odour concentration was greater.

6.5 Summary

The classification of bacteria type using the best classifier from chapter 4 was only partially successful, with only 73.61% of all patterns in the testing feature-set being correctly classified (compared to 96.11% for two bacteria types). Up to that point, only static gas sensor feature models had been employed. New feature-sets were formed using a dynamic feature model, this employed features relating to the rate of change of resistance with time of the gas sensor. Feature-sets which contained only dynamic features and feature-sets which contained a mixture of static and dynamic features were investigated. The classification performance using purely dynamic features was poor, at only 24.72% of all patterns being correctly classified. This figure rose to 82.22% for a mixture of dynamic and static features. Linear Discriminant Function Analysis was used to benchmark the MLP with dynamic and static features, this indicated that the MLP was superior as only 48.06% of all patterns were correctly classified.

Previous investigation (chapter 4) highlighted a relationship between culture growth phase and odour, since growth phase could be predicted from bacteria odour (a MLP achieved 80.28% accuracy). A MLP was trained to predict the fuzzy membership of an input pattern to one of three growth phases. The output from this MLP was fed into a second MLP in order to provide growth phase compensation of variance within the data. The result of compensation for growth phase was an increase in classification performance to 92.78%. The bar chart in Figure 6.5 summarises the comparative performance of the different input features and classifiers.

The identification of a single bacteria type from a mixture was attempted using a MLP. Data from three different mixtures was collected, ***Escherichia coli*** with ***Staphylococcus aureus***, ***Pseudomonas aeruginosa*** with ***Staphylococcus aureus***, and ***Escherichia coli*** with ***Pseudomonas aeruginosa***. The results were encouraging because the bac-

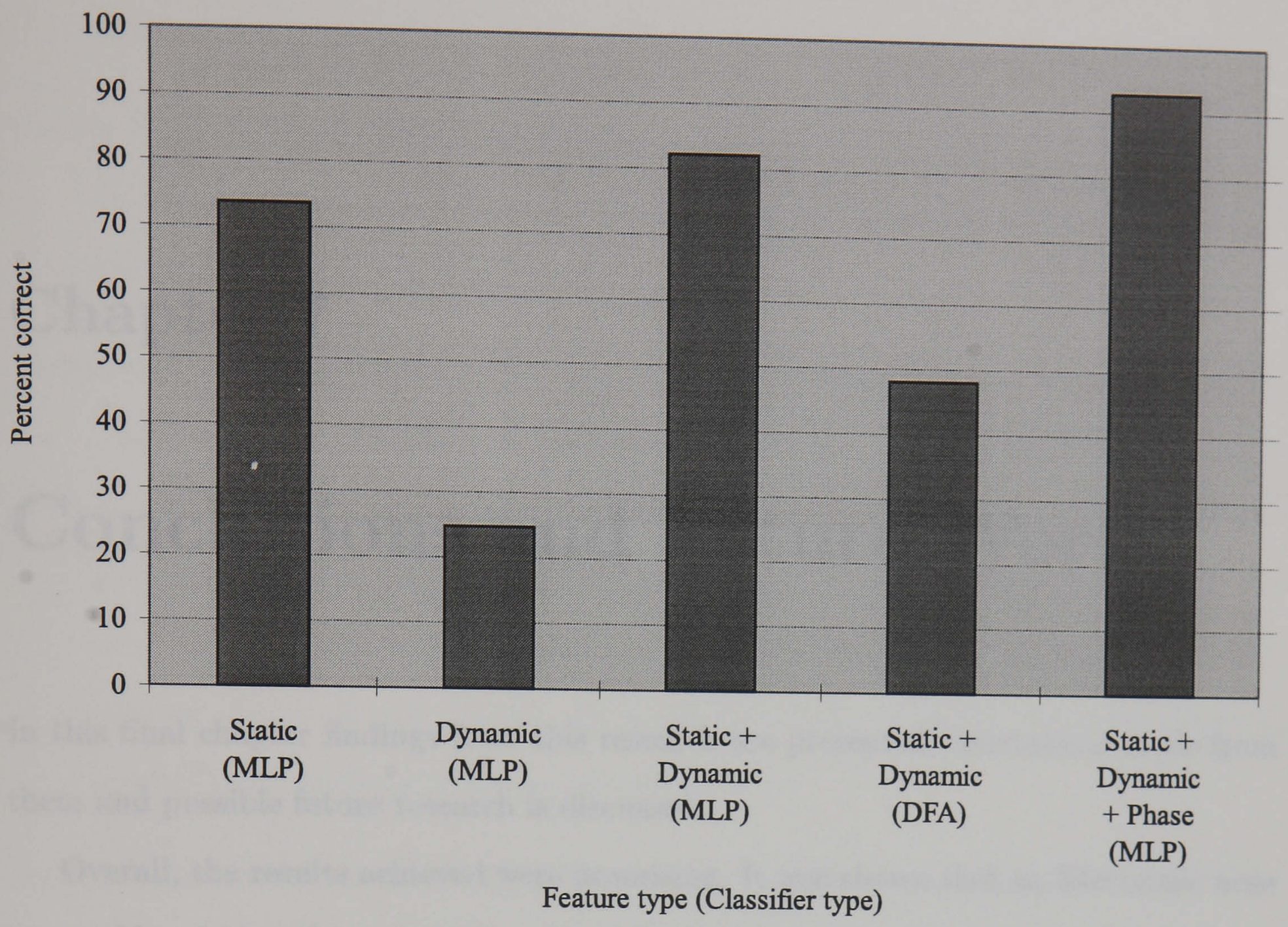


Figure 6.5: Bar chart showing the relative performance of the number of correct classifications of the four different bacteria types for each set of input features and classification technique.

terium type with the weakest odour, *Staphylococcus aureus*, was identified with the highest accuracy, 96.30%. If odours with relatively low concentrations had been problematic then further investigation into odour concentration dependence would have been necessary. Identification for *Escherichia coli* and *Pseudomonas aeruginosa* was less accurate at 59.60% and 63.33% respectively. Because of the similarity between the odours for these two types, there was error when the MLP distinguished between the two. The fact that weak odours can be distinguished is important because it allows easier application of electronic nose technology, thus these findings are important and promising.

Chapter 7

Conclusions and Future Work

In this final chapter findings from this research are presented, conclusions made from them and possible future research is discussed.

Overall, the results achieved were promising. It was shown that an Electronic nose is capable of identifying bacteria types from their odour with the best accuracy of 92.78% being achieved using a multiple-layer perceptron incorporating growth phase compensation for classifying one from four types. Improvement upon initial results was achieved in two ways: the re-design of the instrumentation, and the development of new pattern recognition techniques. The following list states the major conclusions of this thesis:

Automated Odour Delivery Using an automated delivery system, large amounts of data were collected, a total of 2700 'smells' were performed over 15 data collection experiments. The large amounts of data allowed a more accurate estimation of classifier performance (rather than inferior cross-validation).

Rapid Odour Sampling Because the 'dead-volume' up to the sensor chamber was significantly reduced, there was a more 'step-like' response to odours. From this investigation into the dynamic behaviour of gas sensors was more meaningful.

Gas Temperature Control The gas sensor type employed (metal oxide semiconductor) is sensitive to ambient temperature. Gas temperature control was increase to $\pm 0.1°C$ and reduced unwanted variation in the signals from the gas

sensors.

Viable Cell Counts Performing simultaneous, independent counts of colonies enabled odour quality to be correlated against the size and state (growth phase) of the culture.

Feature Extraction Because the 'raw' data were unsuitable for input directly to classifiers, feature-extraction was necessary to perform dimensionality reduction. The effect of different feature models was considerable, however the best feature-extraction models were those that performed minimal processing. Although the more basic feature models output more features and therefore increased the complexity of the classifier, their effect was beneficial compared to more complex models upon eventual classifier performance.

Normalisation The aim of normalisation was to re-scale the feature vectors into ranges more suitable for the classifiers. However, it was found that normalisation at best marginally increased performance and at worst prevented the classifier from performing with any appreciable accuracy. Classifiers are capable of performing internal re-scaling of vectors, with no significant undesired effect upon training.

Classification of Bacteria Type A starting point of classifying bacteria type from the odour of a culture containing a single type was adopted. The best performance achieved employing a multiple-layer perceptron. For two bacteria types, classification accuracy was 96.11%, for four bacteria types using the same classifier design, the accuracy dropped to 73.61%.

Prediction of Culture Growth Phase Because there was more variation in the features from bacteria type than for growth phase, prediction of growth phase was more problematic. The best accuracy achieved, which was using a multiple layer perceptron, was 80.28%. The initial phase was the biggest source of error, as expected.

Advanced Gas Sensor Chamber Design Improving the gas flow and reducing the

volume of the gas sensor chamber improved the dynamic behaviour of the gas sensors. This helped to further reduced unwanted variation within the sensor signals.

Classification using Compensated Neural Net It was found that different features were optimum for bacteria type classification and growth phase prediction. Separating out these two tasks into two different neural nets and then feeding one (growth phase) into the other (type) enhanced performance. The analysis of odour from samples, such as bacteria, that are not stable requires information about the current state of the sample to be processed effectively.

Identification of a Single Type From a Mix A more complex task to set the Electronic Nose, and following on from classification of a single bacteria type, was the identification of bacteria types present in a mixture. It was found that odour similarity rather than concentration was the key factor.

Sensor Drift Metal oxide sensors suffer from baseline drift. The causes of such drift are long term changes in the reactive element. Each data gathering experiment lasted 12 hours, and during this time the base line output from the gas sensors drifted by as much as 1.5 volts (out of a possible output range of 0 to 10 volts). There is no method of preventing sensor drift, it can only be compensated for. In this research, drift compensation occurred within the pattern recognition stage (i.e. many pre-processing models subtracted the baseline and therefore removed it, and the neural nets learnt to ignore drift).

System Calibration In order to achieve good results, calibration of the system is important. Calibration involved exposing the sensors to air for a period of 12 hours with the air temperature set at 36.8°C. During this time the sensors stabilise at their baseline value. The instrumentation was adjusted[1] so that the baseline output was 8 volts. Calibration occurred before each data gathering experiment.

[1]The FOX 2000 has a calibration control for each gas sensor, calibration was therefore straightforward

Some important research has been described in this thesis that will provide the basis for further development. It was never envisaged that a fully working prototype Electronic Nose product would be produced after this research, but rather that some fundamental questions be answered. Can an Electronic Nose classify bacteria types? The answer is yes.

There is scope for future work. Data gathering experiments could be performed using swabs of infected matter from real patients. The Electronic Nose could 'sniff' the breath of patients. Many diseases, such as diabetes, can result in a particular type of odour present in the breath. Wounds, such as burns, could be 'sniffed' and the onset of any infection could be detected.

Although the instrumentation was improved considerably during this research, further improvement might be advantageous:

- The 'bang-bang' temperature controller could be replaced with a proportional one, thus potentially improving performance.
- The brass piping could be replaced with piping constructed from a less reactive substance, for example stainless steel.
- The automated delivery system could be modified further to accommodate a larger number of sample vessels. A larger number of sample vessels would allow simultaneous data gathering from a larger number of bacteria types.
- The gas flow characteristics of the gas sensor chamber could be modeled and possible improvements in gas flow within the chamber implemented.
- Polymer based gas sensors might prove useful, they could also be used in an array containing more than one as sensor type (i.e. metal oxide and polymer types).

As well a instrumentation improvements, there is scope for further pattern recognition research:

- Non-linear DFA could be used. The linear DFA so far used may perform less well than the non-linear varieties.

- More novel neural nets, such as self-learning Kohonen Self Organising Maps (SOMs) or Adaptive Resonance Theory (ART) nets [98], could be tested.

- Another branch of pattern recognition that may be relevant is expert systems. These systems are well established and may yield good results.

Although much was done, there is still a great deal more that can be done. Electronic Nose Technology still has some way to go in the research and development stage before it becomes a common rather than a rare appliance, but from this work and other similar work, the future looks bright for clinical applications.

Appendix A

Virtual Instrumentation Programs

There were three programs in total:

LPM-16 Output Program The LPM-16 Output Program allowed the digital outputs on the National Instruments LPM-16 I/O card to be manually controlled.

Front-End Control Program The Front-End Control Program had two main functions; firstly to control the solenoid valves within the Front-End, and secondly to record sensor signals from the FOX 2000.

Temperature Control Program The Temperature Control Program controlled the temperature of the gas sensor chamber within the FOX 2000.

The following sections show a screen snapshot and program detail for each program.

A.1 LPM-16 Output Program

The LPM-16 output program was used to manually control the digital output lines from the LPM-16 I/O card. The purpose of the program was for trouble shooting. The program was run on its own within the Labview environment in order to avoid hardware clashes (i.e. two programs trying to exclusively control the same hardware). Each switch corresponded to a pair of digital outputs. For the first three switches

(from the left), each switch controlled a pair of digital outputs which in turn controlled a single channel in the automated sampling sub-system. The fourth switch initially had no function, after the implementation of the temperature controlled sensor chamber, it controlled the output to the heaters. Figure A.1 is a screen-shot of the program.

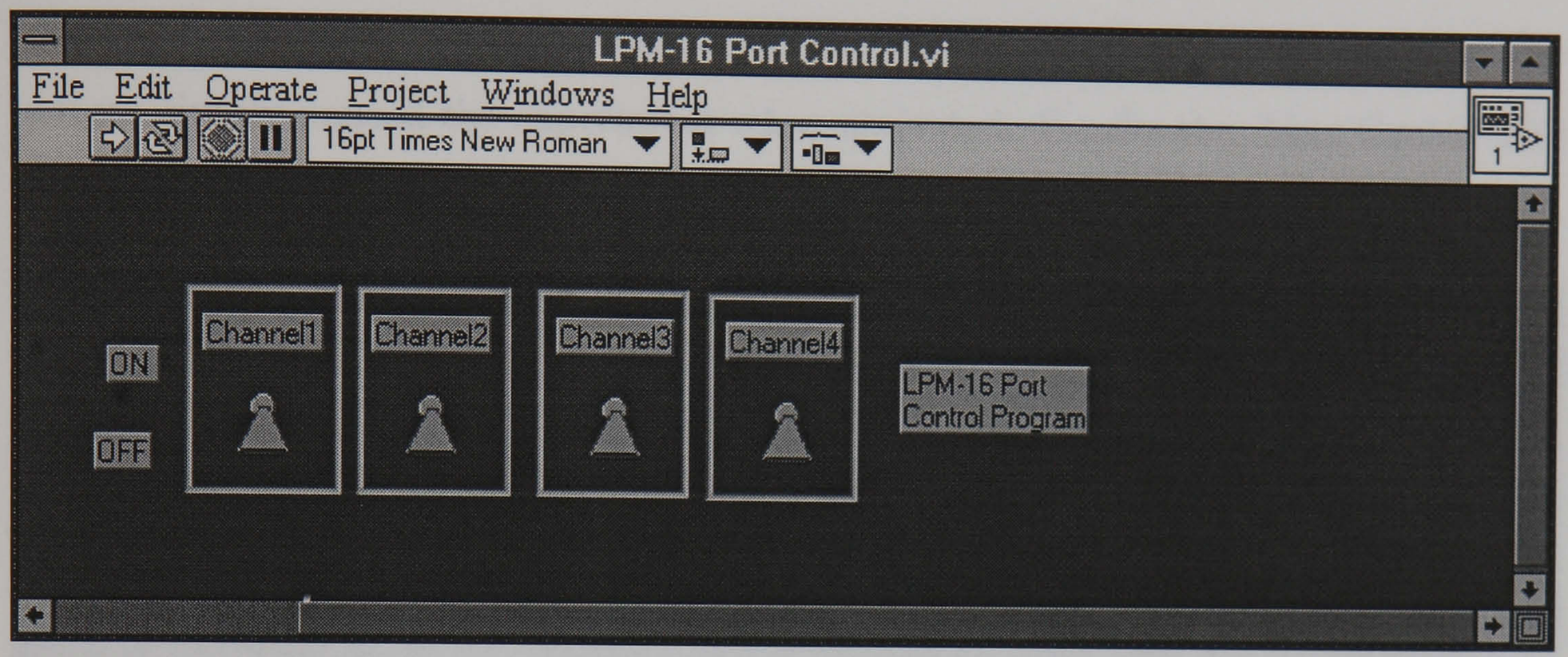


Figure A.1: Snapshot of the LPM-16 Output Program showing typical operation.

A.2 Front-End Control Program

The Front-End control program was the main program. Using this program, the automated sampling sub-system was controlled and the output from the sensors was sampled and stored. Figure A.2 is a screen-shot of the program, references to details in this picture are used to describe its functionality. The 'SAVE CONFIG' button was active when the experiment was stopped or paused, its activation allowed configuration parameters to be saved to a file. Upon startup, a configuration file could be loaded in order to reduce repeated setting-up. Configuration parameters were valve timings, cycle duration and maximum number of cycles. The 'Elapsed Time' display displayed the elapsed time since the beginning of the last completed cycle. The 'Cycle Time' edit boxes allowed the duration of the cycle to be entered or edited. The 'No Cycles' display displayed the number of complete cycles completed since the start of the experiment. The 'Exit Program' button, when activated, simply exited the program. The 'Data File' edit box allowed the current edit file name to be entered or edited. The user is

prompted to enter a data file name upon program startup. The 'Config File' edit box similarly allowed modification of the config file name. In the bottom right, a graph shows the sensor output for the last 200 minutes, with a key along the left hand side. In the bottom left are the edit boxes for setting on and off times for each channel, with an indicator for each channel along the left hand side. The 'Run Mode' switch at the bottom simply paused the experiment when activated. Finally the 'Max Cycles' edit box allowed the maximum number of cycles to set or edited.

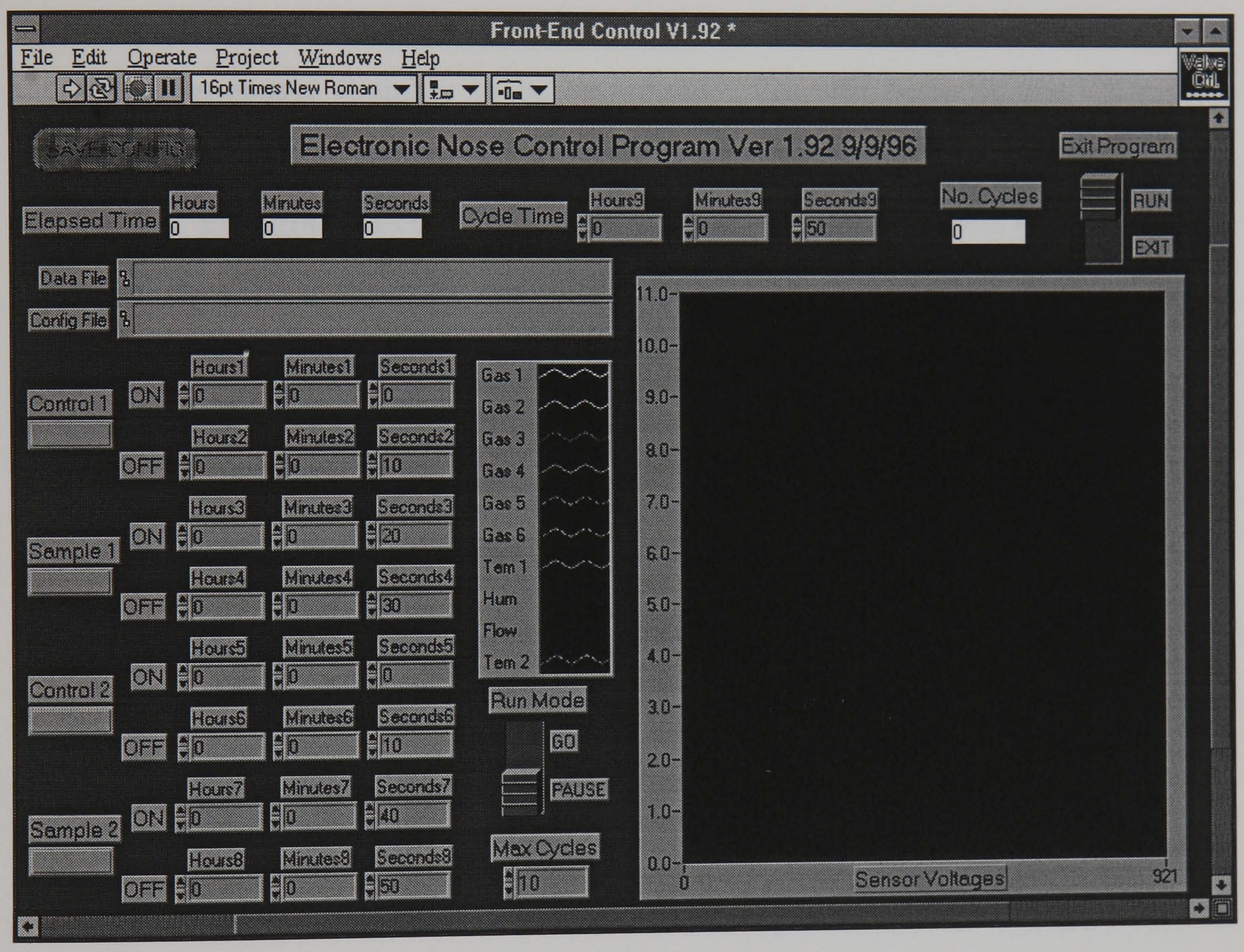


Figure A.2: Snapshot of the Front-End Control Program showing typical operation.

A.3 Temperature Control Program

The temperature control program was used in conjunction with the Front-End Control Program, it controlled the heaters within the Electronic Nose in order to maintain a constant temperature. Figure A.3 is a snapshot of this program. The screen was divided

into seven areas, each area will be briefly described. The 'System Controls' controlled the overall running of the program. The 'Run Mode' switch when activated halted the program, the 'Analysis' switch, when activated allowed analysis of the output from the temperature sensors (described later). The 'Main Heater On' and 'Pre Heater On' were simply indicators that turned red when the respective heater was being powered. The 'Offset' and 'Ratio' edit boxes allowed fine tuning of the internal conversion of temperature sensor output voltage into a numeric value of degrees centigrade. The controls in the 'Target Temps' allows the target temperature for the pre-heater chamber and the main sensor chamber to be set/edited. The edit boxes in the 'Histogram Parameters' section defined the minimum and maximum for the x-axis in the histogram plot (in the 'histogram' section, bottom right section). The 'Statistics (Main Temp)' section contained indicators for the values of the mean and standard deviation of the gas sensor chamber (main) temperature. The 'Temperatures' section contained two indicators for the current temperature of the pre-heater chamber and the main sensor chamber. Finally, the 'Temperature History' section contained a plot showing the previous temperature of the pre-heater chamber and the main chamber for the last two hours.

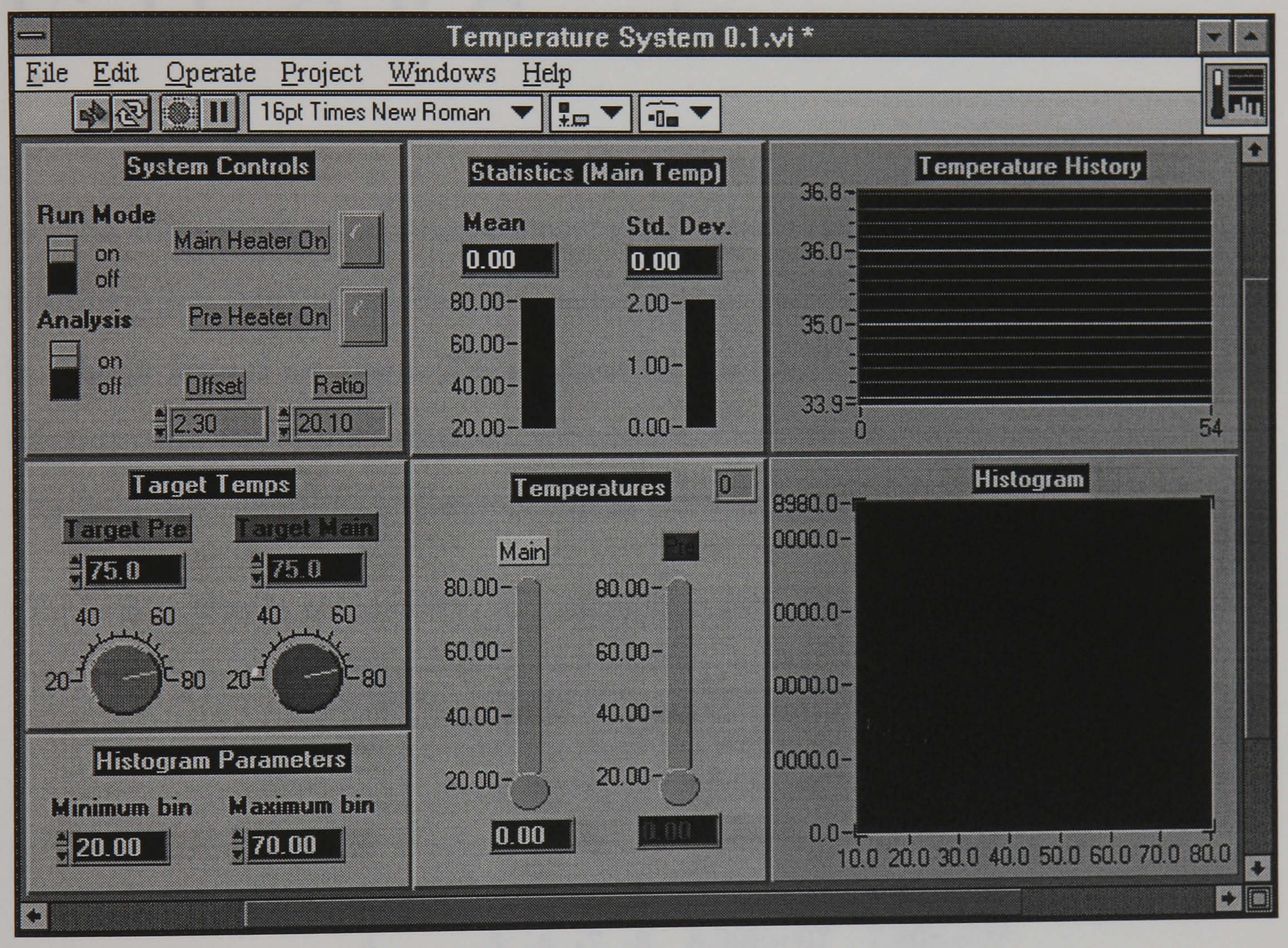


Figure A.3: Snapshot of the Temperature Control Program showing typical operation.

Appendix B

Viable Cell Counts

In total there were 16 experiments performed were viable counts (i.e. counts of colony forming units (cfu))of the bacteria cultures were performed. Here, the actual counts in tabular format are given. The notation for the dilutions denote the number of prior dilutions and the current dilution, therefore a serial dilution of -4 denotes that 3 prior dilutions were performed, therefore this is the fourth dilution.

Table B.1: Viable counts (cfu) in 1 ml of inoculum, experiment 1 (***Escherichia coli***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-4, -5, -6, -7	2.90E+06	2.90E+06
1	-4, -5, -6, -7	2.20E+06	2.70E+06
2	-4, -5, -6, -7	1.60E+07	9.00E+06
3	-5, -6, -7, -8	4.80E+07	4.00E+07
4	-5, -6, -7, -8	8.90E+07	7.80E+07
5	-6, -7, -8, -9	1.90E+08	1.60E+08
6	-6, -7, -8, -9	2.00E+08	1.80E+08
7	-6, -7, -8, -9	2.60E+08	1.40E+08
8	-6, -7, -8, -9	1.70E+08	2.10E+08
9	-6, -7, -8, -9	3.20E+08	2.10E+08
10	-6, -7, -8, -9	2.20E+08	2.70E+08
11	-6, -7, -8, -9	3.60E+08	1.70E+08
12	-6, -7, -8, -9	2.80E+08	3.10E+08

Table B.2: Viable counts (cfu) in 1 ml of inoculum For experiment 2 (***Escherichia coli***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-3, -4, -5, -6	1.20E+06	1.20E+06
1	-3, -4, -5, -6	2.50E+06	2.50E+06
2	-4, -5, -6, -7	7.10E+06	1.70E+07
3	-4, -5, -6, -7	3.40E+07	3.70E+07
4	-4, -5, -6, -7	7.30E+07	6.70E+07
5	-4, -5, -6, -7	1.10E+08	7.70E+07
6	-5, -6, -7, -8	1.40E+08	2.10E+08
7	-5, -6, -7, -8	2.60E+08	2.90E+08
8	-5, -6, -7, -8	2.90E+08	6.60E+08
9	-5, -6, -7, -8	2.40E+08	1.30E+09
10	-5, -6, -7, -8	5.40E+08	1.40E+09
11	-5, -6, -7, -8	3.40E+08	1.30E+09
12	-5, -6, -7, -8	3.50E+08	1.20E+09

Table B.3: Viable counts (cfu) in 1 ml of inoculum For experiment 3 (***Staphylococcus aureus***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	1.30E+05	1.30E+05
1	-2, -3 ,-4, -5	6.60E+05	3.70E+05
2	-3, -4, -5, -6	2.20E+06	1.70E+06
3	-3, -4, -5, -6	9.90E+06	4.90E+06
4	-4, -5, -6, -7	2.20E+07	2.70E+07
5	-4, -5, -6, -7	3.40E+07	3.50E+07
6	-5, -6, -7, -8	4.90E+07	7.10E+07
7	-5, -6, -7, -8	6.40E+07	6.00E+07
8	-5, -6, -7, -8	6.00E+07	4.70E+07
9	-5, -6, -7, -8	4.70E+07	6.60E+07
10	-5, -6, -7, -8	5.30E+07	5.10E+07
11	-5, -6, -7, -8	6.20E+07	6.00E+07
12	-5, -6, -7, -8	6.10E+07	6.00E+07

Table B.4: Viable counts (cfu) in 1 ml of inoculum For experiment 4 (***Staphylococcus aureus***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	2.40E+05	2.40E+05
1	-3, -4, -5, -6	3.00E+05	3.00E+05
2	-3, -4, -5, -6	7.50E+05	8.40E+05
3	-4, -5, -6, -7	6.20E+06	4.80E+06
4	-4, -5, -6, -7	1.10E+07	8.90E+06
5	-4, -5, -6, -7	1.70E+07	1.60E+07
6	-4, -5, -6, -7	2.90E+07	3.80E+07
7	-4, -5, -6, -7	4.40E+07	4.90E+07
8	-5, -6, -7, -8	5.20E+07	5.20E+07
9	-5, -6, -7, -8	5.90E+07	5.50E+07
10	-5, -6, -7, -8	5.90E+07	4.70E+07
11	-5, -6, -7, -8	6.00E+07	4.70E+07
12	-5, -6, -7, -8	5.90E+07	4.80E+07

Table B.5: Viable counts (cfu) in 1 ml of inoculum, experiment 5 (***Escherichia coli***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	1.93E+06	1.93E+06
1	-2, -3 ,-4, -5	1.43E+06	2.09E+06
2	-3, -4, -5, -6	8.40E+06	1.11E+07
3	-3, -4, -5, -6	6.10E+07	6.50E+07
4	-4, -5, -6, -7	1.30E+08	1.08E+08
5	-4, -5, -6, -7	1.02E+08	1.10E+08
6	-5, -6, -7, -8	9.10E+07	1.23E+08
7	-5, -6, -7, -8	1.58E+08	1.06E+08
8	-5, -6, -7, -8	1.10E+08	1.28E+08
9	-5, -6, -7, -8	9.30E+07	1.21E+08
10	-5, -6, -7, -8	1.69E+08	1.54E+08
11	-5, -6, -7, -8	1.99E+08	2.11E+08
12	-5, -6, -7, -8	2.00E+08	2.42E+08

Table B.6: Viable counts (cfu) in 1 ml of inoculum, experiment 6 (***Escherichia coli***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	2.05E+06	2.05E+06
1	-2, -3 ,-4, -5	3.92E+06	2.84E+06
2	-3, -4, -5, -6	1.09E+07	1.19E+07
3	-3, -4, -5, -6	6.40E+07	5.00E+07
4	-4, -5, -6, -7	1.18E+08	1.26E+08
5	-4, -5, -6, -7	1.05E+08	1.32E+08
6	-5, -6, -7, -8	1.04E+08	1.23E+08
7	-5, -6, -7, -8	1.68E+08	1.51E+08
8	-5, -6, -7, -8	1.96E+08	1.47E+08
9	-5, -6, -7, -8	2.30E+08	9.90E+07
10	-5, -6, -7, -8	1.80E+08	1.06E+08
11	-5, -6, -7, -8	2.90E+08	2.20E+08
12	-5, -6, -7, -8	1.80E+08	1.39E+08

Table B.7: Viable counts (cfu) in 1 ml of inoculum, experiment 7 (***Staphylococcus aureus***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	1.32E+06	1.32E+06
1	-2, -3 ,-4, -5	7.20E+05	1.30E+06
2	-2, -3 ,-4, -5	7.50E+06	2.30E+06
3	-3, -4, -5, -6	2.79E+06	4.90E+06
4	-3, -4, -5, -6	4.90E+06	8.50E+06
5	-3, -4, -5, -6	8.00E+06	1.55E+07
6	-4, -5, -6, -7	1.51E+07	3.16E+07
7	-4, -5, -6, -7	2.30E+07	5.40E+07
8	-4, -5, -6, -7	4.60E+07	1.03E+08
9	-4, -5, -6, -7	7.10E+07	1.20E+08
10	-4, -5, -6, -7	9.30E+07	1.74E+08
11	-4, -5, -6, -7	1.28E+08	2.15E+08
12	-4, -5, -6, -7	1.82E+08	2.63E+08

Table B.8: Viable counts (cfu) in 1 ml of inoculum, experiment 8 (***Staphylococcus aureus***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	9.70E+05	9.70E+05
1	-2, -3 ,-4, -5	1.50E+06	2.70E+06
2	-2, -3 ,-4, -5	3.00E+06	7.40E+06
3	-3, -4, -5, -6	1.06E+07	1.31E+07
4	-3, -4, -5, -6	2.96E+07	5.00E+07
5	-3, -4, -5, -6	5.90E+07	5.00E+07
6	-4, -5, -6, -7	8.40E+07	1.26E+08
7	-4, -5, -6, -7	1.41E+08	1.12E+08
8	-4, -5, -6, -7	1.80E+08	1.80E+08
9	-4, -5, -6, -7	2.09E+08	2.80E+08
10	-4, -5, -6, -7	2.50E+08	3.70E+08
11	-4, -5, -6, -7	4.10E+08	3.22E+08
12	-4, -5, -6, -7	3.60E+08	3.00E+08

Table B.9: Viable counts (cfu) in 1 ml of inoculum, experiment 9 (***Pseudomonas aeruginosa***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	2.59E+06	2.59E+06
1	-2, -3 ,-4, -5	9.90E+05	1.02E+06
2	-3, -4, -5, -6	4.90E+06	4.40E+06
3	-3, -4, -5, -6	2.40E+07	1.41E+07
4	-4, -5, -6, -7	2.70E+07	1.63E+07
5	-4, -5, -6, -7	3.30E+07	2.80E+07
6	-5, -6, -7, -8	8.10E+07	6.50E+07
7	-5, -6, -7, -8	1.54E+08	8.50E+07
8	-5, -6, -7, -8	3.10E+08	2.73E+08
9	-5, -6, -7, -8	4.90E+08	5.30E+08
10	-5, -6, -7, -8	4.90E+08	6.80E+08
11	-5, -6, -7, -8	9.90E+08	8.60E+08
12	-5, -6, -7, -8	8.40E+08	1.08E+09

Table B.10: Viable counts (cfu) in 1 ml of inoculum, experiment 10 (***Pseudomonas aeruginosa***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	1.54E+06	1.54E+06
1	-2, -3 ,-4, -5	1.10E+06	1.48E+06
2	-3, -4, -5, -6	5.90E+06	5.00E+06
3	-3, -4, -5, -6	1.89E+07	1.30E+07
4	-4, -5, -6, -7	2.70E+07	2.20E+07
5	-4, -5, -6, -7	4.40E+07	2.00E+07
6	-5, -6, -7, -8	5.80E+07	7.10E+07
7	-5, -6, -7, -8	2.23E+08	2.14E+08
8	-5, -6, -7, -8	2.57E+08	3.13E+08
9	-5, -6, -7, -8	2.24E+08	4.00E+08
10	-5, -6, -7, -8	4.80E+08	3.90E+08
11	-5, -6, -7, -8	6.50E+08	5.50E+08
12	-5, -6, -7, -8	6.10E+08	5.00E+08

Table B.11: Viable counts (cfu) in 1 ml of inoculum, experiment 11 (***Streptococcus pyogenes***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-2, -3 ,-4, -5	7.90E+04	7.90E+04
1	-2, -3 ,-4, -5	2.10E+04	3.20E+04
2	-2, -3 ,-4, -5	2.20E+04	3.20E+04
3	-3, -4, -5, -6	2.00E+04	3.00E+04
4	-3, -4, -5, -6	3.00E+04	1.30E+05
5	-3, -4, -5, -6	4.00E+04	2.10E+05
6	-4, -5, -6, -7	8.00E+04	2.00E+05
7	-4, -5, -6, -7	2.64E+05	2.20E+05
8	-4, -5, -6, -7	4.40E+05	2.00E+05
9	-4, -5, -6, -7	6.30E+05	3.00E+05
10	-4, -5, -6, -7	8.20E+05	5.00E+05
11	-4, -5, -6, -7	1.00E+06	6.00E+05
12	-4, -5, -6, -7	1.00E+06	6.50E+05

Table B.12: Viable counts (cfu) in 1 ml of inoculum, experiment 12 (***Streptococcus pyogenes***).

Age (hrs)	Dilutions	Vessel 2	Vessel 3
0	-1, -2, -3, -4	4.80E+04	4.80E+04
1	-1, -2, -3, -4	4.80E+04	5.00E+04
2	-1, -2, -3, -4	3.50E+04	5.70E+04
3	-1, -2, -3, -4	3.20E+04	3.90E+04
4	-1, -2, -3, -4	2.40E+04	4.90E+04
5	-1, -2, -3, -4	3.00E+04	3.10E+04
6	-2, -3, -4, -5	2.50E+04	3.00E+04
7	-2, -3, -4, -5	3.60E+04	2.20E+04
8	-2, -3, -4, -5	2.20E+04	6.40E+04
9	-2, -3, -4, -5	2.00E+04	3.90E+04
10	-2, -3, -4, -5	2.20E+04	3.40E+04
11	-2, -3, -4, -5, -6	3.00E+04	1.61E+05
12	-2, -3, -4, -5, -6	3.80E+04	2.29E+05

Table B.13: Viable counts (cfu) in 1 ml of inoculum, experiment 13 (***Escherichia coli*** and ***Staphylococcus aureus***).

Age (hrs)	Dilutions	*Escherichia coli*		*Staphylococcus aureus*	
		Vessel 2	Vessel 3	Vessel 2	Vessel 3
0	-2, -3, -4, -5	2.50E+06	2.50E+06	7.40E+05	7.40E+05
1	-2, -3, -4, -5	1.75E+06	1.48E+06	1.60E+06	1.30E+06
2	-3, -4, -5, -6	9.20E+06	3.50E+06	3.10E+06	1.90E+06
3	-3, -4, -5, -6	4.70E+07	3.49E+07	4.10E+06	4.80E+06
4	-4, -5, -6, -7	6.80E+07	6.50E+07	1.00E+07	1.10E+07
5	-4, -5, -6, -7	8.10E+07	7.00E+07	1.10E+07	1.20E+07
6	-5, -6, -7, -8	1.06E+08	8.10E+07	3.20E+07	4.00E+07
7	-5, -6, -7, -8	1.86E+08	1.39E+08	1.80E+07	2.20E+07
8	-5, -6, -7, -8	2.70E+08	1.83E+08	2.30E+07	2.00E+07
9	-5, -6, -7, -8	4.50E+08	3.80E+08	2.20E+07	2.00E+07
10	-5, -6, -7, -8	6.30E+08	3.00E+08	2.40E+07	2.40E+07
11	-5, -6, -7, -8	7.40E+08	3.60E+08	2.80E+07	2.00E+07
12	-5, -6, -7, -8	7.40E+08	4.80E+08	2.40E+07	2.20E+07

Table B.14: Viable counts (cfu) in 1 ml of inoculum, experiment 14 (***Pseudomonas aeruginosa*** and ***Staphylococcus aureus***).

Age (hrs)	Dilutions	*Pseudomonas aeruginosa*		*Staphylococcus aureus*	
		Vessel 2	Vessel 3	Vessel 2	Vessel 3
0	-2, -3, -4, -5	1.14E+06	1.14E+06	3.70E+05	3.70E+05
1	-2, -3, -4, -5	8.70E+05	8.50E+05	1.04E+06	1.00E+06
2	-3, -4, -5, -6	4.50E+06	5.90E+06	1.70E+06	3.90E+06
3	-3, -4, -5, -6	1.37E+07	9.90E+06	5.90E+06	6.20E+06
4	-4, -5, -6, -7	2.60E+07	1.52E+07	1.34E+07	1.11E+07
5	-4, -5, -6, -7	2.80E+07	2.24E+07	3.00E+07	2.56E+07
6	-5, -6, -7, -8	5.30E+07	3.80E+07	2.00E+07	2.30E+07
7	-5, -6, -7, -8	1.61E+08	1.54E+08	4.40E+07	4.50E+07
8	-5, -6, -7, -8	5.40E+08	1.98E+08	6.30E+07	4.80E+07
9	-5, -6, -7, -8	4.80E+08	4.80E+08	1.83E+08	7.00E+07
10	-5, -6, -7, -8	5.31E+08	3.19E+08	1.92E+08	7.70E+07
11	-5, -6, -7, -8	5.88E+08	2.12E+08	2.01E+08	8.46E+07
12	-5, -6, -7, -8	6.50E+08	1.41E+08	2.10E+08	9.30E+07

Table B.15: Viable counts (cfu) in 1 ml of inoculum, experiment 15 (***Escherichia coli*** and ***Pseudomonas aeruginosa***).

Age (hrs)	Dilutions	*Escherichia coli*		*Pseudomonas aeruginosa*	
		Vessel 2	Vessel 3	Vessel 2	Vessel 3
0	-2, -3, -4, -5	3.60E+05	3.60E+05	1.66E+06	1.66E+06
1	-2, -3, -4, -5	3.00E+05	4.60E+05	1.00E+06	1.20E+06
2	-3, -4, -5, -6	4.10E+06	2.70E+06	1.02E+07	6.80E+06
3	-3, -4, -5, -6	2.50E+07	5.40E+06	2.20E+07	8.00E+06
4	-4, -5, -6, -7	6.40E+07	4.80E+07	2.50E+07	8.00E+06
5	-4, -5, -6, -7	7.80E+07	7.20E+07	5.00E+07	1.90E+07
6	-5, -6, -7, -8	9.00E+07	7.10E+07	1.00E+08	9.00E+07
7	-5, -6, -7, -8	6.90E+07	4.00E+07	5.30E+07	7.90E+07
8	-5, -6, -7, -8	7.00E+07	4.00E+07	4.80E+07	7.50E+07
9	-5, -6, -7, -8	6.50E+07	4.40E+07	5.90E+07	1.48E+08
10	-5, -6, -7, -8	6.80E+07	4.50E+07	1.08E+08	1.71E+08
11	-5, -6, -7, -8	7.00E+07	4.20E+07	1.01E+08	1.04E+08
12	-5, -6, -7, -8	7.20E+07	4.40E+07	1.58E+08	1.25E+08

Appendix C

Results of ANN Analysis

C.1 Initial Analysis On Data From Experiments 1 to 4

Table C.1: Classification of bacteria type, results of a 2 layer MLP trained using BP with momentum, trained from experiments 1 and 3 and tested with experiments 2 and 4, using all the feature-set types listed in table 4.1.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
dfn	338.80	85.56	11.39	3.06
dfs	514.93	64.72	15.83	19.44
dfa	368.08	75.28	11.11	13.61
dfv	82.18	93.89	2.50	3.61
rln	577.14	19.17	6.11	74.72
rls	530.37	58.06	13.89	28.06
rla	538.97	51.11	14.44	34.44
rlv	629.95	13.33	6.94	79.72
fdn	617.38	17.22	8.89	73.89
fds	517.33	61.67	13.61	24.72
fda	516.22	46.11	9.44	44.44
fdv	568.47	25.00	8.06	66.94

Table C.2: Classification of bacteria type, results of a 2 layer MLP trained using BP with momentum, trained from experiments 1 and 3 and tested with experiments 2 and 4, using all the feature-set types listed in table 4.2.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
afn	271.12	76.67	3.89	19.44
afs	298.24	56.39	2.50	41.11
afa	366.93	62.78	7.22	30.00
afv	226.19	82.22	5.56	12.22
mnn	308.59	88.33	11.67	0.00
mns	74.21	96.11	2.22	1.67
mna	530.93	47.78	9.44	42.78
mnv	487.91	20.28	1.67	78.06
frn	648.41	0.00	0.28	99.72
frs	675.81	53.33	16.39	30.28
fra	430.91	74.44	14.44	11.11
frv	707.62	0.28	0.56	99.17
mdn	1085.51	46.11	34.17	19.72
mds	851.73	61.39	27.50	11.11
mda	644.03	46.94	16.94	36.11
mdv	908.56	44.17	25.28	30.56
mfn	875.25	10.00	12.50	77.50
mfs	933.21	5.56	13.33	81.11
mfa	567.99	47.50	12.22	40.28
mfv	760.24	1.11	0.00	98.89
ffn	570.13	35.00	5.00	60.00
ffs	675.81	53.33	16.39	30.28
ffa	430.91	74.44	14.44	11.11
ffv	746.95	21.94	11.39	66.67

Table C.3: Classification of bacteria type, results of a 2 layer MLP trained using BP with momentum, trained from experiments 2 and 4 and tested with experiments 1 and 3, using all the feature-set types listed in table 4.1.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
dfn	314.97	78.06	8.33	13.61
dfs	424.41	68.33	11.39	20.28
dfa	433.35	76.39	14.72	8.89
dfv	298.78	75.56	6.94	17.50
rln	330.09	71.11	6.94	21.94
rls	677.59	0.00	0.28	99.72
rla	680.35	60.00	21.39	18.61
rlv	413.45	47.22	1.94	50.83
fdn	304.56	79.72	10.00	10.28
fds	723.62	15.28	9.72	75.00
fda	628.27	58.06	17.78	24.17
fdv	469.16	45.28	3.89	50.83

Table C.4: Classification of bacteria type, results of a 2 layer MLP trained using BP with momentum, trained from experiments 2 and 4 and tested with experiments 1 and 3, using all the feature-set types listed in table 4.2.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
afn	472.13	65.28	9.72	25.00
afs	750.79	42.22	21.94	35.83
afa	349.96	61.11	4.17	34.72
afv	375.48	42.22	2.78	55.00
mnn	541.76	49.72	18.33	31.94
mns	304.26	73.89	3.89	22.22
mna	470.83	58.89	12.22	28.89
mnv	625.15	46.11	19.44	34.44
frn	600.22	0.00	0.83	99.17
frs	52.90	93.06	0.28	6.67
fra	639.63	66.94	21.11	11.94
frv	714.34	0.00	0.00	100.00
mdn	519.15	78.33	17.22	4.44
mds	567.98	76.11	19.17	4.72
mda	617.26	67.78	24.44	7.78
mdv	292.80	70.00	3.89	26.11
mfn	521.48	44.72	7.50	47.78
mfs	885.30	3.33	3.06	93.61
mfa	422.78	71.39	12.50	16.11
mfv	677.26	7.50	7.22	85.28
ffn	702.63	0.00	0.00	100.00
ffs	52.90	93.06	0.28	6.67
ffa	639.63	66.94	21.11	11.94
ffv	947.90	12.78	13.33	73.89

Table C.5: Classification of culture growth phase, results of a 2 layer MLP trained using BP with momentum, trained from experiments 1 and 3 and tested with experiments 2 and 4, using all the feature-set types listed in table 4.1.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
dfn	740.20	25.56	7.50	66.94
dfs	691.46	50.83	14.72	34.44
dfa	870.27	44.44	26.11	29.44
dfv	670.30	25.28	4.44	70.28
rln	717.09	0.00	0.00	100.00
rls	529.03	55.56	14.17	30.28
rla	599.55	58.06	16.67	25.28
rlv	728.53	0.00	0.00	100.00
fdn	604.10	28.33	8.61	63.06
fds	579.11	26.67	3.33	70.00
fda	579.55	2.22	4.72	73.06
fdv	636.38	27.78	8.06	64.17

Table C.6: Classification of culture growth phase, results of a 2 layer MLP trained using BP with momentum, trained from experiments 1 and 3 and tested with experiments 2 and 4, using all the feature-set types listed in table 4.2.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
afn	835.83	26.11	13.06	60.83
afs	640.72	53.61	14.17	32.33
afa	597.04	70.00	18.33	11.67
afv	742.92	0.00	0.00	100.00
mnn	928.13	19.72	9.44	70.83
mns	771.62	31.39	13.89	54.72
mna	695.03	57.78	18.06	24.17
mnv	741.43	8.33	6.94	84.72
frn	587.25	9.44	1.11	89.44
frs	558.23	52.50	12.50	35.00
fra	549.14	23.89	3.61	72.50
frv	669.29	1.11	0.56	98.33
mdn	761.39	48.89	20.28	30.83
mds	799.95	0.00	0.00	100.00
mda	997.97	33.61	25.28	41.11
mdv	664.87	32.33	8.89	58.89
mfn	776.80	0.00	0.28	99.72
mfs	742.75	0.00	0.00	100.00
mfa	790.27	3.61	1.39	95.00
mfv	731.27	0.00	0.00	100.00
ffn	653.76	4.72	2.50	92.78
ffs	558.23	52.50	12.50	35.00
ffa	549.14	23.89	3.61	72.50
ffv	630.06	0.56	1.11	98.33

Table C.7: Classification of culture growth phase, results of a 2 layer MLP trained using BP with momentum, trained from experiments 2 and 4 and tested with experiments 1 and 3, using all the feature-set types listed in table 4.1.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
dfn	691.07	41.67	10.56	47.78
dfs	478.14	45.83	4.17	50.00
dfa	708.27	53.61	20.56	25.83
dfv	572.03	47.78	8.61	43.61
rln	807.08	0.00	0.00	100.00
rls	878.47	0.00	0.56	99.44
rla	898.21	50.56	25.56	23.89
rlv	705.61	0.00	0.00	100.00
fdn	752.10	0.28	0.83	98.89
fds	777.43	25.56	5.00	69.44
fda	909.35	48.33	26.67	25.00
fdv	673.92	26.94	19.44	53.61

Table C.8: Classification of culture growth phase, results of a 2 layer MLP trained using BP with momentum, trained from experiments 2 and 4 and tested with experiments 1 and 3, using all the feature-set types listed in table 4.2.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
afn	581.18	46.39	6.39	47.22
afs	390.71	76.11	10.00	13.89
afa	423.38	80.28	13.61	6.11
afv	616.29	4.72	3.06	92.22
mnn	766.12	15.28	20.83	63.89
mns	802.36	30.00	21.22	48.89
mna	1168.41	44.17	28.61	27.22
mnv	761.93	0.00	0.00	100.00
frn	648.96	6.94	3.61	89.44
frs	531.55	36.39	5.83	57.78
fra	846.49	51.11	27.50	21.39
frv	625.56	20.83	3.89	75.28
mdn	914.84	3.61	10.00	86.39
mds	757.29	0.00	0.00	100.00
mda	1121.57	31.11	31.67	37.22
mdv	774.55	31.67	13.61	54.72
mfn	770.13	0.00	0.00	100.00
mfs	794.08	0.00	0.00	100.00
mfa	823.08	0.00	0.28	99.72
mfv	741.87	0.00	0.00	100.00
ffn	820.48	14.47	8.06	77.78
ffs	531.55	36.39	5.83	57.78
ffa	846.49	51.11	27.50	21.39
ffv	780.29	3.61	1.67	94.72

C.2 Initial Analysis On Data From Experiments 5 to 12

Table C.9: Classification of bacteria type, results of a 2 layer MLP trained using BP with momentum, trained from experiments 1 and 3 and tested with experiments 2 and 4, using all the feature-set types listed in table 4.1.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
dfn	2134.37	24.58	10.42	65.00
dfs	2107.89	5.97	4.44	89.58
dfa	2276.12	3.06	6.81	90.14
dfv	2068.08	16.67	14.86	68.47
rln	2162.88	10.83	38.89	50.28
rls	2132.04	23.75	36.53	39.72
rla	2202.66	15.69	19.58	64.72
rlv	2212.84	4.17	21.39	74.44
fdn	2100.33	13.33	4.72	81.94
fds	2021.43	31.25	32.78	35.97
fda	2190.22	17.22	24.44	58.33
fdv	1944.94	27.22	29.86	42.92

Table C.10: Classification of bacteria type, results of a 2 layer MLP trained using BP with momentum, trained from experiments 5, 7, 9 and 11 and tested with experiments 6, 8, 10 and 12, Using All The Feature-Set Types Listed in Table 4.2.

'feature-set'	SSE	Right (%)	Wrong (%)	Unknown (%)
afn	1827.07	53.19	36.81	10.00
afs	2308.13	15.00	17.50	67.50
afa	2467.92	24.58	33.33	13.47
afv	1869.67	52.36	17.64	30.00
mnn	1155.00	73.61	25.56	0.83
mns	2343.88	21.81	37.50	40.69
mna	2530.37	16.11	27.22	56.67
mnv	1733.07	55.14	36.39	8.47
frn	2064.10	26.53	33.61	39.86
frs	2011.55	29.03	35.83	35.14
fra	1964.32	15.00	11.94	73.06
frv	2013.45	17.08	19.58	63.33
mdn	1951.68	39.44	38.75	21.81
mds	2268.45	21.81	36.53	41.67
mda	2258.38	2.50	9.44	88.06
mdv	1934.30	24.86	27.22	60.14
mfn	1941.45	33.89	36.94	29.17
mfs	2040.51	28.89	34.72	36.39
mfa	2136.70	47.50	12.22	40.28
mfv	1931.35	26.25	22.92	50.83
ffn	1850.41	60.58	38.17	1.25
ffs	1957.93	35.83	30.14	34.03
ffa	1990.18	14.31	13.89	71.81
ffv	2077.83	17.08	18.56	64.35

Appendix D

Data Pre-Processing and Normalisation Program Listings

D.1 Data Pre-Processing

This program took data from the electronic nose and pre-processed it by feature-extraction. The program took command line arguments to control its behaviour, namely the algorithm to use, the number of gas sensors, the number of non-gas sensors, the number of cycles and the file name to input.

```
//-------------------------------------------------------------
// Program name: data pre-processor
// File name: preproc.cpp
// Author: Mark Craven
// Date last modified: 7/6/96
// Language: C++
// Command line: preproc alg_no no_gas no_nongas max_cycles base_filename
// Program description: Performs various preprocessing algorithms
//
// pre-processing algorithms
//
// alg_no abbrev description
//
//  1 df  difference
//  2 rl  relative
//  3 fd  fractional difference
//  4 af  absolute final
//  5 mn  minimum
```

```
//  6 fr  final relative
//  7 md  modified difference
//  8 mf  modified fractional difference
//  9 ff  final fractional difference
//  10  ba  baseline
//  11  tm  transient model + mn
//  12  tr  transient model
//
//-------------------------------------------------------------------

#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define VERSION "preproc ver 1.0 30/1/97"
#define MAX_LINE_SIZE 200 // maximum characters in one line
#define DELAY 0           // delay before max or min readings taken
#define NO_MEAS_CYL 120   // number of measurements per "smell" cycle
#define EXP_PERIOD 12     // number of hours colonies were counted
#define SENSOR_MIN 0.0    // minimum voltage value from any sensor
#define SENSOR_MAX 10.0   // maximum voltage value from any sensor

double transient(double *pVolts)
{
  double Min=10.0,
  thresh;

  for (int zz=0;zz<NO_MEAS_CYL;++zz)
    if (pVolts[zz] <Min)
      Min =pVolts[zz];

  thresh =0.80*( pVolts[0]-Min );

  for (zz =0;zz<NO_MEAS_CYL;++zz)
    if ( (pVolts[0]-pVolts[zz]) > thresh )
      break;

  return( (double)zz/NO_MEAS_CYL );
}

int main(int argc, char *argv[], char *envp[])
{
  cout << "Labview preprocessor V1.0" << endl; // Ouput info to screen

  // Check for incorrect number of arguments
```

```
if (argc !=6)
{
  cout << "Usage: preproc alg_no no_gas no_nongas max_cycles base_filename"
  << endl;
  exit(1);
}

// Initialise and check algorithm string from argument 3 value
char sAlgorithm[3];
int nAlgorithm =atoi(argv[1]);

switch(nAlgorithm)
{
  case 1:
    strcpy(sAlgorithm,"df");  // difference model
    break;

  case 2:
    strcpy(sAlgorithm,"rl");  // relative model
    break;

  case 3:
    strcpy(sAlgorithm,"fd");  // fractional difference model
    break;

  case 4:
    strcpy(sAlgorithm,"af");  // absolute final output model
    break;

  case 5:
    strcpy(sAlgorithm,"mn");  // minimum output model
    break;

  case 6:
    strcpy(sAlgorithm,"fr");  // final relative model
    break;

  case 7:
    strcpy(sAlgorithm,"md");  // modified difference model
    break;

  case 8:
    strcpy(sAlgorithm,"mf");  // modified fractional difference model
    break;

  case 9:
    strcpy(sAlgorithm,"ff");  // final fractional difference model
    break;
```

```
case 10:
  strcpy(sAlgorithm,"ba");   // baseline model
  break;

case 11:
  strcpy(sAlgorithm,"tm");   // transient model
  break;

  case 12:
    strcpy(sAlgorithm,"tr");    // transient model
    break;

  default:
    cout << "Incorrect algorithm number specified, must be 1 to 10"
    << endl; // error in argument
    exit(1);
}

// Initialise number of gas and non-gas sensors (hence no. of columns known)
const int nNoGas =atoi(argv[2]);
const int nNoNonGas =atoi(argv[3]);

// Initialise maximum number of cycles to process
const int nMaxCycles =atoi(argv[4]);

// Initialise input and output file names and open files, check for errors
char *sInFile =new char [strlen(argv[5])+5];
char *sOutFile =new char [strlen(argv[5])+5];

strcpy(sInFile,argv[5]);
strcpy(sOutFile,argv[5]);

fstream InputFile(strcat(sInFile,".dat"),ios::in),
// construct file input object
OutputFile(strcat(strcat(sOutFile,"."),sAlgorithm),ios::out);
// construct file output object

if ( InputFile.fail() || OutputFile.fail() )
// if a file cannot be opened then exit
{
  cout << "File Error!" << endl;
  exit(1);
}

// set some number formatting options for O/P streams (file and console)
cout.setf(ios::showpoint);
cout.setf(ios::fixed, ios::floatfield);
```

```
OutputFile.setf(ios::showpoint);
OutputFile.setf(ios::fixed, ios::floatfield);
cout.precision(4);
OutputFile.precision(4);

char pTmpStr[200];  // temporary storage for reading lines
double *pReadings =new double[nNoGas+nNoNonGas];
// temporary space for readings
// define the parameters to be extracted

double *pVFinalRef =new double[nNoGas];
double *pVMaxRef =new double[nNoGas];
double *pVMinRef =new double[nNoGas];
double *pVFinalOdour =new double[nNoGas];
double *pVMaxOdour =new double[nNoGas];
double *pVMinOdour =new double[nNoGas];

double *pTMaxRef =new double[nNoGas];
double *pTMinRef =new double[nNoGas];
double *pTMaxOdour =new double[nNoGas];
double *pTMinOdour =new double[nNoGas];

double *pVAveNonGas =new double[nNoNonGas];

double **ppVSensor =new double*[nNoGas];
for (int counter=0;counter<nNoGas;++counter)
  ppVSensor[counter] =new double[NO_MEAS_CYL];

// Initialise Parameters

for (int xx=0;xx<(nNoGas+nNoNonGas);++xx) // zero workspaces
  pReadings[xx] =0.0;

for (xx=0;xx<nNoGas;++xx)
{
  pVFinalRef[xx] =SENSOR_MIN;
  pVMaxRef[xx] =SENSOR_MIN;
  pVMinRef[xx] =SENSOR_MAX;
  pVFinalOdour[xx] =SENSOR_MIN;
  pVMaxOdour[xx] =SENSOR_MIN;
  pVMinOdour[xx] =SENSOR_MAX;
  pTMaxRef[xx] =0.0;
  pTMinRef[xx] =0.0;
  pTMaxOdour[xx] =0.0;
  pTMinOdour[xx] =0.0;
}
```

```
for (xx=0;xx<nNoNonGas;++xx)
  pVAveNonGas[xx] =0.0;

// Output that file headrer is being read
cout << "File header" << endl;

// Output information into output file
OutputFile << "#" << VERSION << " : " << sAlgorithm << endl;
int CurrChannel=0, // current channel being read
PrevChannel=0, // previous channel that was read
nCycleCount=1,
time =1;

double channel,
       feature1,
 feature2,
 feature3;

// Read in input file header and output to console and outputfile
while ( InputFile.get() == '#' )
{
  InputFile.getline(pTmpStr,MAX_LINE_SIZE);
  cout << pTmpStr << endl;
  OutputFile << "#" << pTmpStr << endl;
}

// Set input file pointer to begining of first data line
InputFile.seekg(-1,ios::cur);

// Output column labels to console and output file
OutputFile << "#Ch No.\t";
cout << "#Ch No.\t";

switch(nAlgorithm)
{
  case 1:
  case 2:
  case 3:
  case 6:
  case 7:
  case 8:
  case 9:
  case 10:
  case 12:
    for (xx=0;xx<nNoGas;++xx)
    {
OutputFile << "Gas" << (xx+1) << "\t";
```

```
cout << "Gas" << (xx+1) << "\t";
    }
    break;

  case 4:
  case 5:
    for (xx=0;xx<nNoGas;++xx)
    {
OutputFile << "Gas" << (xx+1) << "a" << "\t";
cout << "Gas" << (xx+1) << "a" << "\t";
OutputFile << "Gas" << (xx+1) << "b" << "\t";
cout << "Gas" << (xx+1) << "b" << "\t";
    }
    break;

  case 11:
    for (xx=0;xx<nNoGas;++xx)
    {
OutputFile << "Gas" << (xx+1) << "a" << "\t";
cout << "Gas" << (xx+1) << "a" << "\t";
OutputFile << "Gas" << (xx+1) << "b" << "\t";
cout << "Gas" << (xx+1) << "b" << "\t";
OutputFile << "Gas" << (xx+1) << "c" << "\t";
cout << "Gas" << (xx+1) << "c" << "\t";
    }
    break;
}


for (xx=0;xx<nNoNonGas;++xx)
{
  OutputFile << "NGas" << (xx+1) << "\t";
  cout << "NGas" << (xx+1) << "\t";
}

OutputFile << endl;
cout << endl;

// lets mark the beginning of data within the input file
streampos marker = InputFile.tellg();

// main loop: mode =0 for channel 2, mode = 1 for channel 3
for (int mode=0;mode<2;++mode)
{
  while ( ( !InputFile.eof() ) && ( nCycleCount <=nMaxCycles ) )
  {
    // read channel number, store as current channel and convert to number
    InputFile >> channel;
```

```
CurrChannel =(int)channel;
switch(CurrChannel)
{
  case 3:
    CurrChannel =1;
    break;

  case 12:
    CurrChannel =2;
    break;

  case 48:
    CurrChannel =3;
    break;

  default:
    cout << "Error in channel notation in input file" << endl;
    exit(1);
}

// read data from ALL sensors and store in array
for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
  InputFile >> pReadings[xx];

switch(CurrChannel)
{
  case 1:
  // if the current channel is 1 and the previous channel
  // was 2 the end of "smell"
  if ( PrevChannel ==2 )
  {
    // only process if in channel 2 mode (mode=0)
    if ( mode ==0 )
    {
      // output previous channel (i.e. the last "smell"
      // odour phase channel)
      OutputFile << PrevChannel;
      cout << PrevChannel << ":";

// using necessary parameters, calculate features
for (xx=0;xx<nNoGas;++xx)
{
  switch(nAlgorithm)
  {
  case 1:
    // difference model
    feature1 =pVMaxRef[xx]-pVMinOdour[xx];
    feature2 =0.0;
```

```
    feature3 =0.0;
    break;

    // relative model
  case 2:
    feature1 =pVMinOdour[xx]/pVMaxRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Fractional difference model
  case 3:
    feature1 =(pVMaxRef[xx]-pVMinOdour[xx])/pVMaxRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Absolute final output model
  case 4:
    feature1 =pVFinalOdour[xx];
    feature2 =pVFinalRef[xx];
    feature3 =0.0;
    break;

    // Minimum output model
  case 5:
    feature1 =pVMinOdour[xx];
    feature2 =pVMinRef[xx];
    feature3 =0.0;
    break;

    // Final Relative model
  case 6:
    feature1 =pVFinalOdour[xx]/pVFinalRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Modified difference model
  case 7:
    feature1 =(pVMaxOdour[xx]-pVMinOdour[xx])-
    (pVMaxRef[xx]-pVMinRef[xx]);
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Modified fractional difference model
  case 8:
```

```
    feature1 =(pVMaxOdour[xx]-pVMinOdour[xx])/
    (pVMaxRef[xx]-pVMinRef[xx]);
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Final fractional difference mode
  case 9:
    feature1 =(pVFinalOdour[xx]-pVFinalRef[xx])/pVFinalRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Baseline mode
  case 10:
    feature1 =pVFinalRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Transient model + mn
  case 11:
    //feature1 =(pVFinalRef[xx]-pVMinOdour[xx])/pTMinOdour[xx];
    //feature2 =0.0;
    feature1 =pVMinRef[xx];
    feature2 =pVMinOdour[xx];
    feature3 =transient(ppVSensor[xx]);
    break;

    // Transient model
  case 12:
    feature1 =transient(ppVSensor[xx]);
    feature2 =0.0;
    feature3 =0.0;
    break;

  } //end of case

  // Output featres to console and outputfile
  OutputFile << "\t" << feature1;
  cout << "\t" << feature1;
  if(feature2 !=0.0)
  {
    OutputFile << "\t" << feature2;
    cout << "\t" << feature2;
  }
  if(feature3 !=0.0)
  {
```

```
      OutputFile << "\t" << feature3;
      cout << "\t" << feature3;
    }
  } // end of for

  // Output non-gas sensor features
  for (xx=0;xx<nNoNonGas;++xx)
  {
    feature1 =pVAveNonGas[xx]/(NO_MEAS_CYL*2);
      OutputFile << "\t" << feature1;
      cout << "\t" << feature1;
    }

    // Finish current output line
    OutputFile << endl;
    cout << endl;
  } //end of if mode

  time =1;
} //end of if channel

if ( PrevChannel ==3 )
{
  if (mode ==1)
  {
    // output previous channel (i.e. the last "smell" odour
    // phase channel)
    OutputFile << PrevChannel;
    cout << PrevChannel << ":";

    // using necessary parameters, calculate features
    for (xx=0;xx<nNoGas;++xx)
    {
      switch(nAlgorithm)
      {
      case 1:
        // difference model
        feature1 =pVMaxRef[xx]-pVMinOdour[xx];
        feature2 =0.0;
        feature3 =0.0;
        break;

        // relative model
      case 2:
        feature1 =pVMinOdour[xx]/pVMaxRef[xx];
        feature2 =0.0;
        feature3 =0.0;
        break;
```

```
    // Fractional difference model
  case 3:
    feature1 =(pVMaxRef[xx]-pVMinOdour[xx])/pVMaxRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Absolute final output model
  case 4:
    feature1 =pVFinalOdour[xx];
    feature2 =pVFinalRef[xx];
    feature3 =0.0;
    break;

    // Minimum output model
  case 5:
    feature1 =pVMinOdour[xx];
    feature2 =pVMinRef[xx];
    feature3 =0.0;
    break;

    // Final Relative model
  case 6:
    feature1 =pVFinalOdour[xx]/pVFinalRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Modified difference model
  case 7:
    feature1 =(pVMaxOdour[xx]-pVMinOdour[xx])-
    (pVMaxRef[xx]-pVMinRef[xx]);
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Modified fractional difference model
  case 8:
    feature1 =(pVMaxOdour[xx]-pVMinOdour[xx])/
    (pVMaxRef[xx]-pVMinRef[xx]);
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Final fractional difference mode
  case 9:
    feature1 =(pVFinalOdour[xx]-pVFinalRef[xx])/pVFinalRef[xx];
```

```
    feature2 =0.0;
    feature3 =0.0;
    break;

    // Baseline mode
  case 10:
    feature1 =pVFinalRef[xx];
    feature2 =0.0;
    feature3 =0.0;
    break;

      // Transient model +mn
  case 11:
    //feature1 =(pVFinalRef[xx]-pVMinOdour[xx])/pTMinOdour[xx];
    //feature2 =0.0;
    feature1 =pVMinRef[xx];
    feature2 =pVMinOdour[xx];
    feature3 =transient(ppVSensor[xx]);
    break;

    // Transient model
  case 12:
    feature1 =transient(ppVSensor[xx]);
    feature2 =0.0;
    feature3 =0.0;
    break;
  } //end of case

  // Output featres to console and outputfile
  OutputFile << "\t" << feature1;
  cout << "\t" << feature1;
  if(feature2 !=0.0)
  {
    OutputFile << "\t" << feature2;
    cout << "\t" << feature2;
  }
  if(feature3 !=0.0)
  {
    OutputFile << "\t" << feature3;
    cout << "\t" << feature3;
  }

} // end of for

// Output non-gas sensor features
for (xx=0;xx<nNoNonGas;++xx)
{
  feature1 =pVAveNonGas[xx]/(NO_MEAS_CYL*2);
```

```
      OutputFile << "\t" << feature1;
      cout << "\t" << feature1;
    }

    // Finish current output line
    OutputFile << endl;
    cout << endl;
  } //end of if mode
  time =1;
    ++nCycleCount;
  }

  // if the current channel is 1 and the previous channel
  // was 2 or 3 the reset parameters
  if (PrevChannel !=1)
  {
    for (xx=0;xx<nNoGas;++xx)
    {
      pVFinalRef[xx] =SENSOR_MIN;
      pVMaxRef[xx] =SENSOR_MIN;
      pVMinRef[xx] =SENSOR_MAX;

      pTMaxRef[xx] =0.0;
      pTMinRef[xx] =0.0;
    }

    for (xx=0;xx<nNoNonGas;++xx)
      pVAveNonGas[xx] =0.0;
  }

  // Measure parameters for channel 1 (ref) phase
  for (xx=0;xx<nNoGas;++xx)
  {
    if ( ( pReadings[xx] > pVMaxRef[xx] ) && ( time >=DELAY ) )
    {
      pVMaxRef[xx] =pReadings[xx];
      pTMaxRef[xx] =time;
    }

    if ( (pReadings[xx] < pVMinRef[xx] ) && ( time >=DELAY ) )
    {
      pVMinRef[xx] =pReadings[xx];
      pTMinRef[xx] =time;
    }

    pVFinalRef[xx] =pReadings[xx];
  }
```

```
    PrevChannel =CurrChannel;
    break;

  case 2:
  case 3:
    if ( (PrevChannel !=2) && (PrevChannel !=3) )
    {
      for (xx=0;xx<nNoGas;++xx)
      {
        pVFinalOdour[xx] =SENSOR_MIN;
        pVMaxOdour[xx] =SENSOR_MIN;
        pVMinOdour[xx] =SENSOR_MAX;

        pTMaxOdour[xx] =0.0;
        pTMinOdour[xx] =0.0;
      }
      time =1;
  }

  for (xx=0;xx<nNoGas;++xx)
  {
    if ( ( pReadings[xx] <pVMinOdour[xx] ) && ( time >=DELAY ) )
    {
      pVMinOdour[xx] =pReadings[xx];
      pTMinOdour[xx] =time;
    }
    if ( ( pReadings[xx] >pVMaxOdour[xx] ) && ( time >=DELAY ) )
    {
      pVMaxOdour[xx] =pReadings[xx];
      pTMaxOdour[xx] =time;
    }

    pVFinalOdour[xx] =pReadings[xx];
    ppVSensor[xx][time-1] =pReadings[xx];
  }

  PrevChannel =CurrChannel;
  break;

  } //end of switch

  for (xx=0;xx<nNoNonGas;++xx)
    pVAveNonGas[xx] +=pReadings[nNoGas+xx];

  ++time;
} // end of while

InputFile.seekg(marker);// reset to beginning of file
```

```
    InputFile.clear(0);

    nCycleCount =1;
    cout << "resetting file pos" << endl;
  } // end of for

  cout << "Finished" << endl;

  delete[] pReadings;
  delete[] pVMaxRef;
  delete[] pVMinRef;
  delete[] pVFinalRef;
  delete[] pVMaxOdour;
  delete[] pVMinOdour;
  delete[] pVFinalOdour;
  delete[] pTMaxRef;
  delete[] pTMinRef;
  delete[] pTMaxOdour;
  delete[] pTMinOdour;
  delete[] pVAveNonGas;

  for (counter=0;counter<nNoGas;++counter)
    delete[] ppVSensor[counter];
    delete[] ppVSensor;

    return 0;
  }
```

D.2 Data Normalisation

This program took data that was output from the pre-processing program and applied one of several normalisation algorthms. The normalisation algorithm, number of gas sensor, number of non-gas sensors, the number of cycles and the filename to input were specified as arguments on the command line.

```
//-------------------------------------------------------------------
// Program name: data normaliser
// File name: normal .cpp
// Author: Mark Craven
// Date last modiified: 31/1/97
// Language: C++
// Command line: normal alg_no no_gas no_nongas max_cycles filename
// Program description: Performs various preprocessing algorithms
//
// normalising algorithms
//
// alg_no abbrev description
//
// 1    __n     none
// 2    __s     sensor(gas) normalisation (column 0->1)
// 3    __a     auto-scaling
// 4    __v     vector (gas components) normalisation (length =1)
//
//-------------------------------------------------------------------

#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define VERSION "normal ver 1.0 10/6/96"
#define MAX_LINE_SIZE 300 // maximum characters in one line
#define LIMIT 0.0 // number of std devs to limit, 0=no limiting
#define SENSOR_MAX 10.0
#define SENSOR_MIN 0.0

int main(int argc, char *argv[], char *envp[])
{
  cout << "Labview data normaliser" << endl;

  // if incorrect number of arguments then exit
  if (argc !=6)
```

```
{
  cout << "Usage: normal alg_no no_gas no_nongas max_cycles filename"
  << endl;
  exit(1);
}

// Initialise and check algorithm string from argument 3 value
char sAlgorithm[2];
int nAlgorithm =atoi(argv[1]);

switch(nAlgorithm)
{
case 1:
    strcpy(sAlgorithm,"n");
    break;

case 2:
    strcpy(sAlgorithm,"s");
    break;

case 3:
    strcpy(sAlgorithm,"a");
    break;

case 4:
    strcpy(sAlgorithm,"v");
    break;

default:
    cout << "Incorrect algorithm number specified, must be 1 to 9"
    << endl;  // error in argument
    exit(1);
}

// Initialise number of gas and non-gas sensors (hence no. of columns known)
int nNoGas =atoi(argv[2]);
int nNoNonGas =atoi(argv[3]);

// Initialise maximum number of cycles to process
int nMaxCycles =atoi(argv[4]);

// Initialise input and output file names and open files, check for errors
char *sInFile =new char [strlen(argv[5])+2];
char *sOutFile =new char [strlen(argv[5])+2];

strcpy(sInFile,argv[5]);
strcpy(sOutFile,argv[5]);
```

```
// construct file input object
fstream InputFile(sInFile,ios::in),
// construct file output object (append algorithm string)
OutputFile(strcat(sOutFile,sAlgorithm),ios::out);

// if a file cannot be opened then exit
if ( InputFile.fail() || OutputFile.fail() )
{
  cout << "File Error!" << endl;
  exit(1);
}

// set some number formatting options for O/P streams (file and console)
cout.setf(ics::showpoint);
cout.setf(ios::fixed, ios::floatfield);
OutputFile.setf(ios::showpoint);
OutputFile.setf(ios::fixed, ios::floatfield);
cout.precision(4);
OutputFile.precision(4);

// temporary storage for reading lines
char pTmpStr[200];
// temporary space for readings
double *pReadings =new double[nNoGas+nNoNonGas];

// Output that file headrer is being read
cout << "File header" << endl;

// Output information into output file
OutputFile << "#" << VERSION << " : " << sAlgorithm << endl;

// Read in input file header and output to console and outputfile
while ( InputFile.get() == '#' )
{
  InputFile.getline(pTmpStr,MAX_LINE_SIZE);
  cout << pTmpStr << endl;
  OutputFile << "#" << pTmpStr << endl;
}

// Set input file pointer to begining of first data line
InputFile.seekg(-1,ios::cur);
int xx;

// lets mark the beginning of data within the input file
streampos marker = InputFile.tellg();

int channel;
double scale;
```

```
// stores average for each variable (one for channel 2 and for 3)
double *pAveRead =new double[nNoGas];

// stores standard deviation for each variable
double *pStdDev =new double[nNoGas];

// stores minimu and maximum for seach gas sensor
double *pMax2 =new double[nNoGas];
double *pMin2 =new double[nNoGas];
double *pMax3 =new double[nNoGas];
double *pMin3 =new double[nNoGas];

// zero workspaces
for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
pReadings[xx] =0.0;

for (xx=0;xx<nNoGas;++xx)
{
  pAveRead[xx] =0.0;
  pStdDev[xx] =0.0;

  pMax2[xx] =SENSOR_MIN;
  pMin2[xx] =SENSOR_MAX;
  pMax3[xx] =SENSOR_MIN;
  pMin3[xx] =SENSOR_MAX;
}

switch(nAlgorithm)
{
  case 1:
    while ( !InputFile.eof() )
      {
        InputFile >> channel;  // read channel

        if (InputFile.eof()) break;

        OutputFile << channel;
        cout << channel;

// read data from ALL sensors and store in array
for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
{
  InputFile >> pReadings[xx];
  OutputFile << "\t" << pReadings[xx];
  cout << "\t" << pReadings[xx];
}
```

```
        OutputFile << endl;
        cout << endl;
      }
      break;

  case 2:
    while ( !InputFile.eof() )
      {
        InputFile >> channel;  // read channel

if (InputFile.eof()) break;

// read data from ALL sensors and store in array
for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
  InputFile >> pReadings[xx];

for (xx=0;xx<nNoGas;++xx)
  switch(channel)
  {
  case 2:
    if (pReadings[xx] <pMin2[xx])
      pMin2[xx] =pReadings[xx];
    if (pReadings[xx] >pMax2[xx])
      pMax2[xx] =pReadings[xx];
    break;

  case 3:
    if (pReadings[xx] <pMin3[xx])
      pMin3[xx] =pReadings[xx];
    if (pReadings[xx] >pMax3[xx])
      pMax3[xx] =pReadings[xx];
    break;
  }
}

InputFile.seekg(marker);// reset to beginning of file
InputFile.clear(0);

while ( !InputFile.eof() )
{
  InputFile >> channel;  // read channel

  if (InputFile.eof()) break;

  OutputFile << channel;
  cout << channel;

  // read data from ALL sensors and store in array
```

```
for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
  InputFile >> pReadings[xx];

for (xx=0;xx<nNoGas;++xx)
  switch(channel)
  {
    case 2:
      OutputFile << "\t"
      << (pReadings[xx]-pMin2[xx])/(pMax2[xx]-pMin2[xx]);
      cout << "\t"
      << (pReadings[xx]-pMin2[xx])/(pMax2[xx]-pMin2[xx]);
      break;

    case 3:
      OutputFile << "\t"
      << (pReadings[xx]-pMin3[xx])/(pMax3[xx]-pMin3[xx]);
      cout << "\t"
      << (pReadings[xx]-pMin3[xx])/(pMax3[xx]-pMin3[xx]);
      break;
  }

  for (xx=nNoGas;xx<(nNoGas+nNoNonGas);++xx)
  {
        OutputFile << "\t" << pReadings[xx];
        cout << "\t" << pReadings[xx];
      }

      OutputFile << endl;
      cout << endl;
    }
    break;

case 3:
  // First Read values to calculate mean
  while ( !InputFile.eof() )
  {
    InputFile >> channel;  // read channel
    if (InputFile.eof()) break;

    // read data from ALL sensors and store in array
    for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
      InputFile >> pReadings[xx];

    for (xx=0;xx<nNoGas;++xx)
      pAveRead[xx] +=pReadings[xx];

}
```

```
for (xx=0;xx<nNoGas;++xx)
  pAveRead[xx] /=nMaxCycles;

InputFile.seekg(marker);// reset to beginning of file
InputFile.clear(0);

// Second Read values to calculate std dev
// (actually variance initially)
while ( !InputFile.eof() )
{
  InputFile >> channel;  // read channel

  if (InputFile.eof()) break;

  for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
    // read data from ALL sensors and store in array
    InputFile >> pReadings[xx];

    for (xx=0;xx<nNoGas;++xx)
    pStdDev[xx] +=( ( pReadings[xx] -pAveRead[xx] ) *
    ( pReadings[xx] -pAveRead[xx] ) );
}

// convert variances to std dev
for (xx=0;xx<(nNoGas);++xx)
{
  pStdDev[xx] /=(nMaxCycles-1);
  pStdDev[xx] =sqrt(pStdDev[xx]);
}

InputFile.seekg(marker);        // reset to beginning of file
InputFile.clear(0);

while ( !InputFile.eof() )
{
  InputFile >> channel;

  if (InputFile.eof()) break;
  OutputFile << channel;
  cout << channel;

  for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
    // read data from ALL sensors and store in array
    InputFile >> pReadings[xx];

  for (xx=0;xx<nNoGas;++xx)
  {
    pReadings[xx] -=pAveRead[xx];
```

```
pReadings[xx] /=pStdDev[xx];

// limit the population to +/- "LIMIT" std dev
if (LIMIT !=0.0)
  if (pReadings[xx] > LIMIT)
    pReadings[xx] = LIMIT;
  else if (pReadings[xx] < (-1.0*LIMIT) )
    pReadings[xx] = -1.0*LIMIT;
    OutputFile << "\t" << pReadings[xx];
    cout << "\t" << pReadings[xx];
  }

  for (xx=nNoGas;xx<(nNoGas+nNoNonGas);++xx)
  {
    OutputFile << "\t" << pReadings[xx];
    cout << "\t" << pReadings[xx];
  }

  OutputFile << endl;
  cout << endl;
}
break;
```

```
case 4:
  while ( !InputFile.eof() )
  {
    scale =0.0;
    InputFile >> channel;  // read channel

    if (InputFile.eof()) break;

    OutputFile << channel;
    cout << channel;

    // read data from ALL sensors and store in array
    for (xx=0;xx<(nNoGas+nNoNonGas);++xx)
      InputFile >> pReadings[xx];

    for (xx=0;xx<nNoGas;++xx)
      scale += ( pReadings[xx] * pReadings[xx] );

    scale =1/(sqrt(scale));

    for (xx=0;xx<nNoGas;++xx)
    {
      OutputFile << "\t" << ( scale*pReadings[xx] );
      cout << "\t" << ( scale*pReadings[xx] );
```

```
    }

    for (xx=nNoGas;xx<(nNoGas+nNoNonGas);++xx)
    {
      OutputFile << "\t" << pReadings[xx];
      cout << "\t" << pReadings[xx];
    }
  }

    OutputFile << endl;
    cout << endl;

    break;
}

delete[] pReadings;
delete[] pAveRead;
delete[] pStdDev;

delete[] pMax2;
delete[] pMin2;
delete[] pMax3;
delete[] pMin3;

  return 0;
}
```

D.3 Pattern File Generation

This program took data files generated by the normalisation program and output pattern files that were suitable for input to the SNNS neural network simulator. The position of the classes within the file, the number of columns, the number of cycles and the filname to input were specified on the command line.

```
//-------------------------------------------------------------
// Program name: labview data parser
// File name: pattern.cpp
// Author: Mark Craven
// Date last modiified: 6/2/97
// Language: C++
// Command line: pattern s1 f1 s2 f2 s3 f3 s4 f4 cols max full_filename
// Program description: Converts data files to SNNS pattern files
//
//-------------------------------------------------------------

#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <string.h>

#define VERSION "pattern ver 1.0 6/2/97"
#define MAX_LINE_SIZE 200 // maximum characters in one line
#define DELAY 5           // delay before max or min readings taken
#define NO_MEAS_CYL 240   // number of measurements per "smell" cycle
#define EXP_PERIOD 12     // number of hours colonies were counted
#define SENSOR_MIN 0.0    // minimum voltage value from any sensor
#define SENSOR_MAX 10.0   // maximum voltage value from any sensor

int main(int argc, char *argv[], char *envp[])
{
  // Check for incorrect number of arguments
if (argc !=12)
{
  cout << "Usage: pattern s1 f1 s2 f2 s3 f3 s4 f4 cols max full_filename"
  << endl;
  exit(1);
}

// Initialise and check algorithm string from argument 3 value
int nStart[4],
  nFinish[4];
int xx;
```

```
for (xx=0;xx<4;++xx)
{
  nStart[xx] =atoi(argv[(xx*2)+1]);
  nFinish[xx] =atoi(argv[(xx*2)+2]);
}

// Initialise maximum number of cycles to process
int nMaxCycles =atoi(argv[10]);

// Initialise number of columns to use
int nCols =atoi(argv[9]);

// Initialise input and output file names and open files, check for errors
char *sInFile =new char [strlen(argv[11])];

strcpy(sInFile,argv[11]);

fstream InputFile(sInFile,ios::in); // construct file input object

if ( InputFile.fail() ) // if a file cannot be opened then exit
{
  cout << "File Error!" << endl;
  exit(1);
}

// set some number formatting options for O/P streams (file and console)
cout.setf(ios::showpoint);
cout.setf(ios::fixed, ios::floatfield);
cout.precision(4);

char pTmpStr[200];  // temporary storage for reading lines
double *pReadings =new double[nCols]; // temporary space for readings

// Initialise Parameters

for (xx=0;xx<nCols;++xx) // zero workspaces
  pReadings[xx] =0.0;

int nChannel=0, // current channel being read
  nCycleCount=1,
  time =1;

// Read in input file header and output to console and outputfile
while ( InputFile.get() == '#' )
  InputFile.getline(pTmpStr,MAX_LINE_SIZE);

// Set input file pointer to begining of first data line
```

```
InputFile.seekg(-1,ios::cur);

// lets mark the beginning of data within the input file
streampos marker = InputFile.tellg();

// main loop
while ( ( !InputFile.eof() ) && ( nCycleCount <=nMaxCycles ) )
{
  // read channel number, store as current channel and convert to number
  InputFile >> nChannel;

  if ( InputFile.eof() ) break;

  // read data from ALL sensors and store in array
  for (xx=0;xx<nCols;++xx)
    InputFile >> pReadings[xx];
  // Dummy readline

  InputFile.getline(pTmpStr,MAX_LINE_SIZE);
  cout << "# Input pattern " << nCycleCount << ":" << endl;

  for (xx=0;xx<nCols;++xx)
    cout << pReadings[xx] << " ";

    cout << endl;
    cout << "# Output pattern " << nCycleCount << ":" << endl;

    for (xx=0;xx<4;++xx)
      if ((nStart[xx] !=-1) && (nFinish[xx] !=-1))
        if ( (nCycleCount >=nStart[xx]) && (nCycleCount <=nFinish[xx]) )
          cout << "1 ";
        else
          cout << "-1 ";

    cout << endl;

    ++nCycleCount;
  }

    delete[] pReadings;

    return 0;
  }
```

References

[1] G. H. Dodd, P. N. Bartlett, and J. W. Gardner. Odours - the stimulus for an electronic nose. In Gardner and Bartlett [45], chapter 1, pages 1–11. ISBN 0-7923-1693-2.

[2] W. J. Freeman and Y. Yao. Chaos in the biodynamics of pattern recognition by neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1, pages 243–249, 1990.

[3] J. W. Gardner and P. N. Bartlett. A brief history of electronic noses. *Sensors and Actuators B*, 18:211–220, 1994.

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

[5] M. S. Nayak, R. Dwivedi, and S. K. Srivastava. Application of iteration technique in association with multiple regression method for identification of mixtures of gases using an integrated gas-sensor array. *Sensors and Actuators, B*, 23:11–16, 1994.

[6] T. C. Pearce, J. W. Gardner, S. Friel, P. N. Bartlett, and N. Blair. Electronic nose for monitoring the flavour of bears. *Analyst*, 118:371–377, April 1993.

[7] P. M. Schweizer-Berberich, S. Vaihinger, and W. Göpel. Characterisation of food freshness with sensor arrays. *Sensors and Actuators B*, 18-19:282–290, 1994.

[8] T. Börjesson, H. Sundgren, T. Eklöv, T. Jonsson, and J. Schnürer. Grain odour classification with an electronic nose. In *The 8th International Conference on Solid State Sensors and Actuators*, Transducers '95 - Eurosensors IX, pages 11–12, Stockholm, Sweden, June 1995.

[9] F. Winquist, E. G. Hörnsten, H. Sundgren, and I. Lundström. Performance of an electronic nose for quality estimation of ground meat. *Measurement Science and Technology*, 4:1493–1500, 1993.

[10] H. Nanto, H. Sokooshi, T. Kawai, and T. Usuda. Freshness detection of sea foods using tin oxide thin film gas sensor. In *Technical Digest of the 10th Sensor Symposium*, pages 195–198, 1991.

[11] E. W. Nester, C. Evans Roberts, B. J. McCarthy, and N. N. Pearsall. *Microbiology: Molecules, Microbes, and Man.* Holt, Rinehart and Winston, Inc., 1973. ISBN 0-03-078580-4.

[12] G. J. Tortora, B. R. Funke, and C. L. Case. *Microbiology an Introduction, Second Edition.* The Benjamin/Cummings Publishing Company, Inc., 1986. ISBN 0-8053-9315-3.

[13] J. A. Washington, editor. *Laboratory Procedures in Clinical Microbiology.* Springer-Verlag New York, Inc., 1981. ISBN 0-387-90531-6.

[14] E. J. Stokes and G. L. Ridgway. *Clinical Bacteriology, Fifth Edition.* Edward Arnold (Publishers) Ltd, 1980. ISBN 0-7131-4361-4.

[15] R. J. Elliot-Martin, P. N. Bartlett, and J. W. Gardner. Monitoring of breath odour in diary cattle. In *Proceedings of 1st Int. Symposium on Olfaction and Electronic Noses*, pages 26–27, Toulouse, France, September 1994.

[16] R. J. Elliot, T. T. Mottram, J. W. Gardner, P. J. Hobbs, and P. N. Bartlett. Method of sampling breath as a monitor of health in dairy cattle. *Journal of Agricultural Engineering Research*, 1997. In press.

[17] F. Winquist, H. Arwin, E. Lund, R. Forster, C. Day, and I Lundström. Screening of irradiated tomatoes by means of an electronic nose. In *The 8th International Conference on Solid-State Sensors and Actuators, and Eurosensors IX*, pages 691–694, Stockholm, Sweden, June 1995.

[18] L. Moy, T. Tan, and J. W. Gardner. Monitoring the stability of perfume and body odours with an 'electronic nose'. *Perfumer and Flavourist*, 19:11–16, July-August 1994.

[19] K. Persuad and G. H. Dodd. Analysis of discrimination mechanisms of the mammalian olfactory system using a model nose. *Nature*, 299:352–355, 1982.

[20] R. W. Moncrieff. An instrument for measuring and classifying odours. *Journal of Applied Physiology*, 16:742, 1961.

[21] W. F. Wilkens and A. D. Hatman. An electronic analog for the olfactory process. *Ann. NY Acad. Sci.*, 116:608, 1964.

[22] T. M. Buck, F. G. Allen, and M. Dalton. Detection of chemical species by surface effects on metals and semiconductors. In T. Bregman and A. Dravnieks, editors, *Surface Effects in Detection*, chapter 1. Spartan Books Inc., USA, 1965.

[23] A. Dravneiks and P. J. Trotter. Polar vapour detection based on thermal modulation of contact potentials. *Journal of Science Instrumentation*, 42:624, 1965.

[24] S. Vaihinger and W. Göpel. Multicomponent analysis in chemical sensing. In W. Göpel, T. A. Jones, M. Kleitz, I. Lundström, and T. Seiyama, editors, *Sensors: A Comprehensive Study*, volume 2/3 of *Chemical Sensors*, chapter 6, pages 191–237. VCH, Weinheim, 1990.

[25] J. W. Gardner and P. N. Bartlett. Pattern recognition in gas sensing. In P. Moseley, J. Norris, and D. Williams, editors, *Techniques and Mechanisms in Gas Sensing*, pages 347–380. Adam Hilger, Bristol, 1991.

[26] R. H. Wright. *The science of smell.* London: Allen, Unwin, 1964.

[27] J. L. Davis and H. Eichenbaum. *Olfaction - A Model System for Computational Neuroscience.* MIT Press, London, England, 1991. ISBN 0-262-04124-3.

[28] M. C. Meilgaard. Flavor chemistry of beer: Part ii: Flavor and threshold of 239 aroma volatiles. *MBAA Tech. Quartley*, pages 151–168, 1975.

[29] G. Sicard, A. Duchamp, M. F. Revial, and A. Holley. Odour discrimination by frog olfactory cells: a recapitulative study. In H. van der Starre, editor, *Olfaction and Taste*, volume VII, pages 171–174. London: IRL Press, 1980.

[30] G. M. Shepherd. Fundemental anatomy, physiology and plasticity of the olfactory system. In *Olfaction - A Model System for Computational Neuroscience* [27], chapter 1, pages 3–41. ISBN 0-262-04124-3.

[31] R. K. Poole and C. S. Dow, editors. *Microbial gas metabolism, mechanistic, metabolic and biotechnological aspects*, volume 14 of *Special Publications of the Society for General Microbiology.* Society for General Microbiology, Academic, London, 1985.

[32] J. S. Kauer. Contributions of topography and parallel processing to odour coding in the vertebrate olfactory pathway. *trends in Neuroscience*, 14:79–85, 1991.

[33] S. Firestein and G. M. Shepherd. Olfactory receptors share antagonist homology with other g-protein coupled receptors. *J. Physiol. (Lond.)*, 430:135–158, 1991.

[34] L Buck and R Axel. A novel multigene family may encode odorant receptors: a molecular basis for odor recognition. *Cell*, 65:175–187, 1991.

[35] P. Gouras. Color vision. In E. R. Kandel and J. H. Schwartz, editors, *Principles of Neural Science*, pages 384–395. Elsevier, New York, 2 edition, 1985.

[36] G. M. Shepherd. *Neurobiology.* Oxford University Press, New York, 2 edition, 1988.

[37] D. Schild. *Chemosensory Information Processing*, volume H39 of *NATO ASI.* Springer-Verlag, Berlin, 1990.

[38] D. Lancet, J. S. Kauer, C. A. Greer, and G. M. Shepherd. High resolution 2-deoxyglucose localisation in olfactory epithelium. *Chem. Senses*, 6:343–349, 1981.

[39] G. M. Shepherd. The olfactory glomerulus: its significance for sensory processing. In Y. Katzuki, R. Morgren, and T. Sato, editors, *Brain Mechanisms of Sensation*, pages 209–223. Wiley, New York, 1981.

[40] J. Pager. A selective modulation of olfactory input suppressed by lesions of the anterior limb on the anterior commissure. *Physiological Behavior*, 13:523–526, 1974.

[41] J. J. Hopfield. Olfactory computation and object perception. In *Proc. Natl. Acad. Sci.*, volume 88 of *Neurobiology*, pages 6462–6466, USA, August 1991.

[42] T. J. Sejnowski, P. K. Kienker, and G. M. Shepherd. Simple pattern recognition models of olfactory discrimination. *Society Neuroscience Abstracts*, 11:970, 1985.

[43] H. Abe, T. Yoshimura, S. Kanaya, Y. Takahashi, and S. Sasaki. Extended studies of the automated odor-sensing system based on plural semiconductor gas sensors with computerised pattern recognition techniques. *Anal. Chim. Acta*, 215:155–168, 1988.

[44] H. V. Shurmer, J. W. Gardner, and H. T. Chan. The application of discrimination techniques to alcohols and tobaccos using tin oxide sensors. *Sensors and Actuators*, 18:361–371, 1989.

[45] J. W. Gardner and P. N. Bartlett, editors. *Sensors and Sensory Systems for an Electronic Nose*. NATO ASI, Series E: Applied Sciences - Vol. 212. Kluwer Academic Publishers, 1992. ISBN 0-7923-1693-2.

[46] T. Nakamoto, K. Fukunishi, and T. Morizumi. Identification capability of odor sensor using quartz-resonator array and neural network pattern recognition. *Sensors and Actuators, B1*, pages 473–476, 1990.

[47] J. W. Gardner and P. N. Bartlett. Odour sensors for an electronic nose. In *Sensors and Sensory Systems for an Electronic Nose* [45], pages 31–51. ISBN 0-7923-1693-2.

[48] B. S. Hoffheins. Using sensor arrays and pattern recognition to identify organic compounds. Msc thesis, The University of Tennessee, Knoxville, US, June 1990. Published by Oak Ridge National Laboratory, Oak Ridge, Tenessee, US.

[49] G. Heiland. Homogeneous semiconducting gas sensors. *Sensors and Actuators*, 2:343–362, 1982.

[50] J. W. Gardner. Detection of vapours and odours from a multisensor array using pattern recognition. part 1: principal components and cluster analysis. *Sensors and Actuators B*, 4:108–116, 1991.

[51] J. W. Gardner, H. V. Shurmer, and T. T. Tan. Application of an electrnic nose to the discrimination of coffee. *Sensors and Actuators B*, 6:71–75, 1992.

[52] P. Horgan, E.L. Hines, and I. A. Ene. Body odour classification using a neural network based electronic nose. Technical report NNBEN4, Department of Engineering, University of Warwick, Coventry, United Kingdom, September 1993.

[53] M. Egashira, Y. Shimizu, and Y. Takao. Trimethylamine sensor based on semiconductive metal oxides for detection of fish freshness. *Sensors and Actuators B*, 1:108–112, 1990.

[54] D. M. Wilson and S. P. DeWeerth. Odor discrimination using steady-state and transient characteristics of tin-oxide sensors. *Sensors and Actuators, B*, pages 123–128, 1995.

[55] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University (Clarendon) Press, 1995. ISBN 0-19-853864-56.

[56] M. Schweizer-Berberich, J. Göppert, A. Hierlemann, J. Mitrovics, U. Weimar, W. Rosenstiel, and W. Göpel. Application of neural-network systems to the

dynamic response of ploymer-based sensor arrays. *Sensors and Actuators, B*, 26-27:232–236, 1995.

[57] J. Samitier, J. M. López-Villegas, S. Marco, L. Cámara, A. Pardo, and O. Ruiz. A new method to analyse signal transients in chemical sensors. *Sensors and Actuators B*, 18-19:308–312, 1994.

[58] S. M. Weiss and C. A. Kulikowski. *Computer Systems That Learn.* Morgan Kaufmann Publishers, Inc., 1991. ISBN 1-55860-065-5.

[59] S. Haykin. *Neural Networks, A Comprehensive Foundation.* Macmillan College Publishing Company, Inc., 1994. ISBN 0-02-352761-7.

[60] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood Limited, 1992. ISBN 0-13-106360.

[61] C. Di Natale, F. Davide, and A. D'Amico. Pattern recognition in gas sensing: well-stated techniques and advances. *Sensors and Actuators B*, 23:111–118, 1995.

[62] W. S. Sarle. Neural networks and statistical models. In *Proceedings of the 19th Annual SAS Users Group International Conference*, pages 1–13, April 1994.

[63] J. W. Gardner and P. N. Bartlett. Pattern recognition in gas sensing. In P. Moseley, J. Norris, and D. E. Williams, editors, *Techniques and Mechanisms in Gas Sensing*, page 347. Adam-Hilger, Bristol, 1991.

[64] J. W. Gardner and P. N. Bartlett. Pattern recognition in odour sensing. In *Sensors and Sensory Systems for an Electronic Nose* [45], pages 161–179. ISBN 0-7923-1693-2.

[65] J. W. Gardner, H. V. Shurmer, and T. T. Tan. Application of an electronic nose to the discrimination of coffees. In *Proceedings of Eurosensors V Conference*, Rome, Italy, October 1991.

[66] C. Hierold and R. Müller. Quantitive analysis of gas mixtures with non-selective gas sensors. *Sensors and Actuators*, 17:587–592, 1989.

[67] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[68] T. Aishima. Aroma discrimination by pattern recognition analysis of responses from semiconductor gas sensor arrays. *Journal of Agricultural Food Chemistry*, 39:752–756, 1991.

[69] M. Nakamura, I. Sugimoto, H. Kuwano, and R. Lemos. Chemical sensing by analysing dynamics of plasma polymer film-coated sensors. *Sensors and Actuators, B*, 20:231–237, 1994.

[70] J. Mitrovics, U. Weimar, and W. Göpel. Linearisation in multicomponent analysis based on a hybrid sensor-array with 19 sensor elements. In *The 8th International Conference on Solid-State Sensors and Actuators, and Eurosensors IX*, Transducers '95 - Eurosensors IX, pages 707–710, Stockholm, Sweden, June 25-27 1995.

[71] A. Nigrin. *Neural Networks for Pattern Recognition.* MIT Press, Cambridge, MA, US, 1993.

[72] J. M. Zurada. *Introduction to Artificial Neural Systems.* PWS Publishing Company, Boston, US, 1992.

[73] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathmatical Biophysics*, 5:115–133, 1943. Reprinted by Anderson and Rosenfeld (1988).

[74] D. O. Hebb. *The Organisation of Behaviour.* John Wiley, New York, 1949.

[75] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organisation in the brain. *Psychological Review*, 65:386–408, 1958.

[76] B. Widrow and M. E. Hoff. Adaptive switching circuits. *IRE WESCON Convention Record, New York*, 4:96–104, 1960. Reprinted in Anderson and Rosenfeld (1988).

[77] M. L. Minsky and S. A. Papert. *Perceptrons.* MIT Press, Cambridge, MA, US. 1969.

[78] S. Grossberg. Adaptive pattern classification and universal recording: 1. parallel development and coding of neural detectors. *Biological Cybernetics*, 23:121–134, 1976.

[79] S. Grossberg. Adaptive pattern classification and universal recording: 2. feedback, expectation, olfaction, illusions. *Biological Cybernetics*, 23:187–202, 1976.

[80] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences of the USA*, volume 79, pages 2554–2558, 1982.

[81] T. Kohonen. Self-organised formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[82] T. Kohonen. Clustering, taxonomy, and topological maps of patterns. In *Proceedings of the 6th International Conference on Pattern Recognition*, pages 114–128, Munich, Germany, 1982.

[83] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propogation. In D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1: Foundations, pages 318–362. MIT Press, Cambridge, MA, US, 1986. Reprinted in Anderson and Rosenfeld (1988).

[84] D. B. Parker. Learning-logic: Casting the cortex of the human in silicon. Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, US, 1985.

[85] Y. LeCun. Une procedure d'apprentissage pour reseau a seuil assymetrique. *Cognitiva*, pages 559–604, 1985.

[86] P. J. Werbos. *Beyond regression: new tools for prediction and analysis in the behavioural sciences.* PhD thesis, Harvard University, Boston, MA, US, 1974.

[87] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[88] H. Sundgren, F. Winquist, I. Lukkari, and I Lundström. Artificial neural networks and gas sensor arrays: quantification of individual components in a gas mixture. *Measurement Science and Technology*, 2:464–469, 1991.

[89] G. S. Broten and H. C. Wood. A neural network approach to analysing multicomponent mixtures. *Measurement Science and Technology*, 4:1096–1105, 1993.

[90] B. Yea, R. Konishi, T. Osaki, and K. Sugahara. The disrimination of many kinds of odor species using fuzzy reasoning and neural networks. *Sensors and Actuators A*, 45:159–165, 1994.

[91] B. Hivert, M. Hoummady, J.M. Henrioud, and D. Hauden. Feasibility of surface acoustic wave (saw) sensor array processing with formal neural networks. *Sensors and Actuators B*, 18-19:645–648, 1994.

[92] W. Ping and X Jun. A novel recognition method for electronic nose using artificial neural network and fuzzy recognition. *Sensors and Actuators, B*, 37(3):169–174, 1996.

[93] R. A. Lemos, M. Hakamura, I. Sugimoto, and H. Kuwano. A self-organising map for chemical vapor classification. In *Technical Digest, Transducers '93*, Yokohama, Japan, June 1993.

[94] C. Di Natale, F. Davide, A. D'Amico, and W. Göpel. Sensor arrays calibration with enhanced neural networks. *Sensors and Actuators B*, 18-19:654–658, 1994.

[95] C. Di Natale, F. Davide, and A. D'Amico. A self-organising system for pattern classification: time varying statistics and sensor drift effects. *Sensors and Actuators B*, 26-27:237–241, 1995.

[96] T. Moriizumi, T. Nakamoto, and Y. Sakuraba. Pattern recognition in electronic noses by artificial neural network models. In Gardner and Bartlett [45], chapter 14, pages 217–236. ISBN 0-7923-1693-2.

[97] M. Schweizer-Berberich, A. Hierlemann, K. Bodenhoöfer, J. Mitrovics, T. Kerner, U. Weimar, and W. Göpel. Evaluation of dynamic sensor signals by artificial neural networks. In *The 8th International Conference on Solid State Sensors and Actuators*, Transducers '95 - Eurosensors IX, pages 679–682, June 1995.

[98] J. W. Gardner, E. L. Hines, and C. Pang. Detection of vapours and odours from a multisensor array using pattern recognition: Self-organising adaptive resonance techniques. *Measurement and Control*, 29(6):172–178, 1996.

[99] P. D. Wasserman. *Advanced Methods in Neural Computing*. Von Nostrand Reinhold, 1993. ISBN 0-442-00461-3.

[100] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, US, 1992.

[101] S. Zaromb and J. R. Stetter. Theoretical basis for identification and measurements of air contaminants using an array of sensors having partially overlapping selectivity. *Sensors and Actuators*, 6:225–243, 1984.

[102] K. Takahashi and S. Nozaki. From intelligent sensors to fuzzy sensors. *Sensors and Actuators A*, 40:89–91, 1994.

[103] D. Vlachos and J. Avaritsiotis. Fuzzy neural networks for gas sensing. In *The 8th International Conference on Solid State Sensors and Actuators, and Eurosensors IX*, Transducers '95 - Eurosensors IX, pages 703–706, Stockholm, Sweden, June 25-29 1995.

[104] T. Nakamoto, A. Fukuda, and T. Moriizumi. Perfume and flavour identification by odor sensing system using quartz-resonator sensor array and neural-network pattern recognition. *Sensors and Actuators B*, 10:85–90, 1993.

[105] A. A. Mousa-Bahia, G. S. V. Coles, M. J. Gibson, and M. J. Willet. The characterisation of semiconductor gas sensors ii: A critical comparison with respect to products of combustion. *Sensors and Actuators B*, 18-19:668–674, 1994.

[106] B. Bourrounnet, T. Talou, and A. Gaset. Application of a multi-gas-sensor device in the meat industry for boar-taint detection. *Sensors and Actuators B*, 26-27:250–254, 1995.

[107] T. Börjesson, T. Eklöv, A. Jonsson, H. Sundgren, and J. Schnürer. Electronic nose for odor classification of grains. *Analytical Techniques and Instrumentation*, 73(4):457–461, 1996.

[108] G. Jobst, I. Moser, P. Svasek, M. Varahram, and G. Urban. Application of minaturized liquid handling system with integrated biosensor array for milk analysis. In *The 8th International Conference on Solid-State Sensors and Actuators, and Eurosensors IX*, Transducers '95 - Eurosensors IX, pages 473–474, Stockholm, Sweden, June 25-27 1995.

[109] K. C. Persaud, S. M. Khaffaf, P. J. Hobbs, and R. W. Sneath. Assessment of conducting polymer odour sensors for agricultural malodour measurements. In *Conducting Polymer Odour Sensors In Agriculture*, pages 495–505. Oxford University Press, 1996.

[110] T. D. Gibson, O. Prosser, J. N. Hulbert, R. W. Marshall, P. Corcoran, P. Lowery, and E. A. Ruck-Keane. Detection and simultaneous identification of microorgansisms from headspace samples using an electronic nose. In *Proceedings of Eurosensors X*, pages 1341–1344, Leuven, Belgium, September 8-11 1996.

[111] P. T. Moseley and B. C. Tofield, editors. *Solid State Gas Sensors*. Adam Hilger, Bristol, 1987.

[112] Institute for Parallel and Distributed High Performance Systems (IPVR), Univeristy of Stuttgart, Germany. *SNNS Version 4.1, Stuttgart Neural Network Simulator User Manual*, 1995. Report No. 6/95.

[113] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis, Third Edition*. Prentice-Hall International, Inc., 1992. ISBN 0-13-041807-2.

[114] B. G. Tabachnick and L. S. Fidell. ***Using Multivariate Statistics***. HarperCollins Publishers, Inc., 2 edition, 1989. ISBN 0-06-046571-9.

[115] C. Di Natale, S. Marcos, F. Davide, and A. D'Amico. Sesnor-array calibration time reduction by dynamic modeling. ***Sensors and Actuators, B***, 25(1-3):578–583, 1995.

[116] L. A. Zadeh. Fuzzy sets. ***Information and Control***, 8:338–353, 1965.

[117] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. In ***IEE Transactions on Systems, Man and Cybernetics***, volume SMC-3, pages 28–44, 1973.