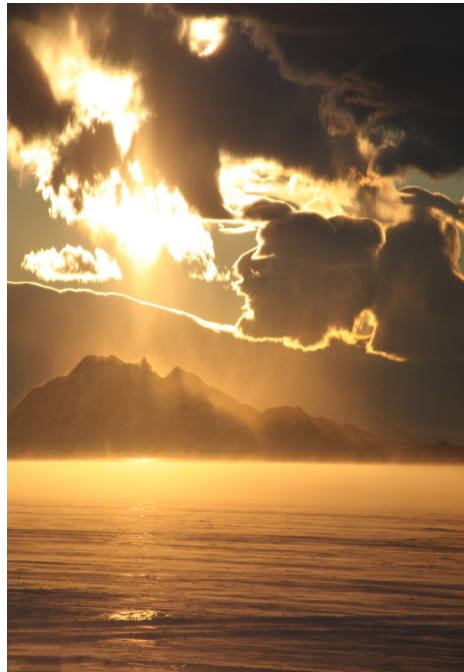




Intelligent Systems: Techniques and Applications

Edited by: Hines, Leeson, Martínez-Ramón
Pardo, Llobet, Iliescu, Yang

This book is concerned with the application of intelligent systems to the solution of realworld problems. It contains 19 chapters from leading researchers in the field. It is divided into five sections, each of which focuses on a specific area: Olfaction, Engineering, Telecommunications, Antenna Arrays, and Medical diagnosis/imaging. This book is of importance to all those who are interested in intelligent systems techniques and applications. For example students, researchers, and practitioners; in areas such as engineering, computer science and so on. Including those interested in artificial intelligence.

Drs Hines, Leeson, Iliescu and Mr. Yang are in the School of Engineering, University of Warwick, UK.

Dr Manel Martínez-Ramón is in the Universidad Carlos III de Madrid, Spain.

Dr Pardo is in CNR - INFM, Italy.

Dr Eduard Llobet is in the Department of Electronics, Universitat Rovira i Virgili, Spain

SHAKER
PUBLISHING

Photo by Steven Profaizer,
Courtesy of the National Science Foundation
Cover design by Jianhua Yang

ISBN 978-90-423-0345-4



Edited by: Hines, Leeson, Martínez-Ramón, Pardo, Llobet, Iliescu, Yang
INTELLIGENT SYSTEMS: TECHNIQUES AND APPLICATIONS

INTELLIGENT SYSTEMS: TECHNIQUES AND APPLICATIONS

Edited by: Evor Hines, Mark Leeson, Manel Martínez-Ramón
Matteo Pardo, Eduard Llobet, Daciana Iliescu, Jianhua Yang

INTELLIGENT SYSTEMS: TECHNIQUES AND APPLICATIONS

Edited by: Evor Hines, Mark Leeson, Manel Martínez-Ramón
Matteo Pardo, Eduard Llobet, Daciana Iliescu, Jianhua Yang

SHAKER
PUBLISHING

St. Maartenslaan 26, 6221 AX Maastricht, The Netherlands.

© Copyright Shaker Publishing 2008

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in The Netherlands.

ISBN: 978-90-423-0345-4

Shaker Publishing BV
St. Maartenslaan 26
6221 AX Maastricht
Tel: 043-3500424
Fax: 043-3255090
<http://www.shaker.nl>

INTELLIGENT SYSTEMS: TECHNIQUES AND APPLICATIONS

Dr Evor L. Hines, School of Engineering, University of Warwick; Coventry, CV4 7AL, UK; Email: E.L.Hines@Warwick.ac.uk; Website: <http://www.eng.warwick.ac.uk/staff/elh/>

Dr Mark S. Leeson, School of Engineering, University of Warwick, Coventry, CV4 7AL, UK; Email: Mark.Leeson@warwick.ac.uk

Dr Manel Martínez-Ramón, Universidad Carlos III de Madrid; Avda de la Universidad, 30; 28911 Leganés-Madrid, Spain; Email: manel@ieee.org

Dr Matteo Pardo, Sensor Laboratory, CNR – INFM; Via Valotti 9 25133 Brescia, Italy; Email: pardo@ing.unibs.it

Dr Eduard Llobet, Department of Electronics, Universitat Rovira i Virgili, Avda. Països Catalans, 26, Campus Sescelades, 43007 Tarragona, Spain; Email: eduard.llobet@urv.cat

Dr Daciana D. Iliescu, School of Engineering, University of Warwick, Coventry, CV4 7AL, UK; Email: D.D.Iliescu@warwick.ac.uk

Mr Jianhua Yang, School of Engineering, University of Warwick, Coventry, CV4 7AL, UK; Email: J.Yang.3@warwick.ac.uk

Preface

The area of Intelligent Systems (ISs) has expanded phenomenally over the years since the 1940s; both in terms of the range of techniques and also in terms of the number of applications wherein they have often provided a competitive edge when compared with others approaches. IS includes a range of techniques that work synergistically and provides, in one form or another, flexible data/information processing capabilities for handling real life situations. IS, unlike conventional techniques, can exploit the tolerance for imprecision, uncertainty/ambiguities, approximate reasoning and partial truth in order to achieve tractability, robustness, and low cost solutions.

The techniques are in general based on biologically inspired strategies for solving problems. At the current time the major categories of IS include neural networks (NNs), fuzzy logic/systems (FL/Ss), evolutionary computation/algorithms (EC/As) (including genetic algorithms (GAs), genetic programming (GP), evolutionary strategies (ES)), support vector machines (SVM), particle swarm optimization (PSO), memetic algorithms (MAs), and ant colony optimization (ACO). In addition hybrid combinations also play a mayor role. These include neuro-fuzzy, neuro-genetic, fuzzy-genetic systems and so on.

These techniques became more and more necessary and popular to meet needs such as: 1) Handling large datasets which are very complex and comprise various forms of uncertainty, in a robust and computationally efficient manner. 2) Much of the data are inherently uncertain and noisy, thus making FLS a natural choice for processing them. 3) The learning capability of NNs, supervised, unsupervised or hybrids can be utilized effectively when extracting patterns from large datasets. This is especially true in data-rich environments where the data may be mined. 4) Many of the tasks involve search and optimization of different criteria (such as energy, alignment score and overlap strength), while requiring robust, fast and close approximate solutions. 5) Evolutionary and other search algorithms such as ACO, PSO provide effective techniques to search and explore very large and multi-modal solution spaces. 6) Additionally, many of the

requirements involve multiple, and often conflicting, objectives thus lending themselves to the application of multi-objective optimization algorithms such as multi-objective GAs.

The focus of this book is on the application of IS Techniques to solve a range of problems in the areas of Olfaction, Engineering, Telecommunications, Antennas and Medical diagnosis/imaging.

This book brings together research articles by eminent experts and active practitioners; reporting recent advances in integrating IS techniques, either individually or in an hybridized manner, for problem solving in order to extract more and more meaningful information and insights. Typically the data to be considered for analysis include: sequence data and structure data, and data types that are typically complex in nature requiring advanced methods to deal with them. Characteristics of the techniques presented here include the use of domain-specific knowledge for reducing the search space, dealing with uncertainty, partial truth and imprecision, efficient linear and/or sub-linear scalability, incremental approaches to knowledge discovery, and an increased level and intelligence of interactivity with human experts and decision makers.

The book has five sections: Olfaction, Engineering, Telecommunications, Antennas and Medical diagnosis/imaging. Let us now consider each one in turn.

Section I – Olfaction

Odour classification systems used in machine olfaction, often termed electronic noses (ENs), have been gaining favour in a wide range of industrial applications. Recent advances in the field of ENs have led to new developments in sensor design, pre-processing for feature extraction and data processing techniques. As a result, the user of EN systems is provided with an increased amount of information for the discrimination of odours using multi-sensor arrays. In this section, EN applications for food analysis and gas sensing are blended with examples of novel processing and analysis methods.

Chapter 1, by Yang et al., applies a GA based sensor selection technique, known as GNMM (Genetic Neural Mathematical Method) to EN data. GNMM uses multilayer perceptrons (MLPs) as the classification engine but separates the optimal sensor subset selection process from the final classification. By utilizing GAs to identify optimal sensors that are subsequently fed as inputs to MLP

classifiers, GNMM simplifies the MLP structure and makes the final classification process more efficient. The effectiveness of GNMM is demonstrated by its application to case study data which has previously been explored in the literature via alternative methods. The results generated by GNMM are generally superior to previous methods demonstrating the effectiveness of the methodology.

The application of GA and NN methods to sensor data, here to virgin olive oil, is the focus of the second chapter by García-González and Aparicio. Although sensing mechanisms and the physical-chemical phenomena that cause sensor responses have been extensively investigated, interpretation of sensor signals is arduous when dealing with the complex mixtures of compounds such as food aromas. This is because there are many interactions between volatiles and the sensitive sensor material producing a single response involving unknown synergic and masking effects. Here, an experimental procedure is proposed to determine the individual contribution of volatile compounds in the sensor response, illustrated with the examples of the aroma of virgin olive oil and metal oxide sensors. Following discussion of preliminary studies, the use of a sensor array as a non-destructive detector in parallel with the flame detector in a gas chromatography (GC-EN) is described. Gas chromatography facilitates study of the relationship between volatile compounds and sensor responses. The chapter also contains a real application of the metal oxide semiconductor sensors to quality assessment.

In the third chapter, from Vergara and Llobet, the analysis of single gases and multi-component mixtures using semiconductor gas sensor arrays and the tools of ISs is investigated. Despite the extensive literature on this subject, important problems such as lack sensor selectivity and a lack of reproducibility remain as research topics. Recent progress in methods to systematically design the use of sensor dynamics has produced significant steps forward in recent years. In this chapter, an approach adapted from system identification that selects estimates of the impulse response of the sensor-gas system from the spectral components present is presented. The selection of the former results in the determination of the optimal temperature modulating frequencies because of the direct relationship between spectral components and modulating frequencies. A practical application is included to reinforce the utility of the method, which is generally applicable.

Chapter 4, by Pardo and De Vito, takes the application of generalized linear discrimination machines or support vector machines (SVMs) as its centre of attention. SVMs have recently been employed in the EN community for pattern analysis in both classification and regression problems. This contribution provides an overview of SVM theory, a review of artificial olfaction and descriptions of interesting classification and regression case histories. In common with any other machine learning technique, SVM requires ‘hyper parameter selection’. In the applications, hyper parameter tuning permits exploration of under- and over-fitting, which form the principal limitations of non-parametric learning techniques.

Chapter 5, from Vergara, Vilanova and Llobet, concludes the section with consideration of dynamic methods for improving the performance of semiconductor gas sensors. The lack of selectivity and the response drift present in semiconductor gas sensors have limited their spectrum of applications to qualitative gas analysis. IS offers tools to improve the gas sensing performance of well-known materials. Sensors in the presence of gas mixtures can be considered as dynamic systems, enabling the employment of methods that are used to study the properties of dynamic systems. This chapter reviews some of the latest developments in the field of gas sensor dynamics, and illustrates their performance for various gas analysis and electronic nose applications.

Section II – Engineering

The considerable breath of application areas in which IS methods are applicable is particularly demonstrated by this section. The chapters develop techniques that range from river flow to yield prediction, incorporating image and signal processing in addition to classification techniques.

Chapter 6, from Yang et al., presents the application of a new technique to the prediction of the longitudinal dispersion coefficient in rivers. The longitudinal dispersion coefficient is a key variable in describing longitudinal transport in a river. In recent years, NNs have become established as tools for environmental modellers, overcoming some of the difficulties associated with traditional statistical approaches. Little attention has been paid to date, however, to the task of selecting the most appropriate NN inputs. In this chapter, the use of a GA based variable selection technique is applied to longitudinal dispersion

coefficient prediction. The GNMM of chapter one is adapted to this purpose with extremely good results, which are favourably compared with principal component analysis (PCA) and the use of a self-organizing map (SOM).

Remote sensing offers the seventh chapter, from Camps-Valls et al., in particular the classification of the images obtained in this area. Remote sensing of the Earth is an active research field of science concerned with acquisition of data by airborne or satellite sensors, and its subsequent transmission and processing. Remote sensing imaging instruments acquire information in several spectral bands that ultimately characterize the pixel in the acquired image. Here the latest developments based on kernel methods for remote sensing image classification are summarised. In particular, the work focuses on various types of SVM, kernel-based fusion of heterogeneous information, multi-temporal classification kernel schemes and the specific problem of change detection with kernels. Methods are analyzed theoretically and through numerical and visual inspection of the classification maps in a number of real scenarios.

There follows a chapter from Sloper, Miotto and Hines, which considers the use of computational intelligence in the context of a large data acquisition project at the European CERN facility. The ATLAS particle physics experiment encompasses a trigger and data acquisition system (TDAQ), which is large, heterogeneous and distributed. This chapter presents the design and development an automatic system for detecting and appropriately dealing with the different types of errors, which may occur in TDAQ. The size and complexity of the system introduces considerable uncertainty and “noise” in the form of irrelevant data or incomplete information. The outputs from fuzzy C-means (FCM), PCA and SOM analysis feed MLP based NNs that are trained to recognize error scenarios with extremely reliable results.

In chapter nine, Li et al. consider the application of unsupervised learning to failure modes in fibre reinforced plastic. The different types of damage occurring in composite materials may be discriminated by analysing acoustic emission (AE). The problem is then the decision as to which failure mechanism produces which AE signature. Here this question is elegantly addressed by obtaining the two most significant frequency components from the AE signal via a fast Fourier transform (FFT). The data are then analyzed using a SOM and FCM, giving successful classification and the identification of plausible physical categories.

Section III – Telecommunications

This section illustrates the range of application areas where IS techniques have been implemented in telecommunications applications. The need to allocate wavelengths or frequencies with constraints leads naturally to difficult computational problems, ideal for stochastic optimisation. In this part, applications in the radio frequency and optical domains are presented since the appropriate techniques map easily from one physical layer to another with adjustment of the fitness function and constraints.

Chapter 10, by An, Hines and Leeson, demonstrates the use of GA and fuzzy logic (FL) methods for cellular radio network optimisation. A cellular radio typically comprises mobile stations and fixed based stations. To achieve high quality mobile communications the coverage of the latter is limited to cells in small geographic areas. This concept produces two technically challenging engineering problems, namely that of channel allocation and call blocking need to be addressed for optimum system performance. The former is formulated as a signal-to-interference ratio (SIR) maximization problem that is implemented using GAs, whereas the latter is tackled using FL with a simple channel borrowing technique. It is shown that the GA optimized channel allocation scheme consistently optimises the network and that the FL system produces a substantial reduction in blocked calls.

Chapter 11, by Kaviani et al., addresses the design of resilient optical core networks using GAs. A key issue in such networks is the assignment of wavelengths to meet the routing needs of the traffic to be conveyed. This is a difficult mathematical problem and thus ideal for GA application, and the inclusion of a back up transmission path to cover fibre failure increases the complexity. In the chapter, lightpaths are encoded into chromosomes made up of a fixed number of genes equal to the number of entries in the traffic demand matrix. Each gene represents one valid path, producing a coding via a variable length binary string. In each new generation, each member of the population represents a set of valid but possibly incompatible paths and some must be discarded. The best paths according to fitness function are assigned the minimum number of wavelengths within the problem constraints. The method is shown to be efficient and able to design survivable optical mesh transport networks.

In the twelfth chapter, from Van Vaerenbergh and Santamaría, the focus on radio frequency transmission, in particular multiple-input multiple-output or

MIMO systems. MIMO systems offer significant increases in spectral efficiency over traditional single-input single-output (SISO) transmission. A number of computationally efficient algorithms for time-invariant MIMO symbol detection exist but their application in time-varying environments is difficult because they need perfect receiver channel state information. The clustering approach to this problem exploits the fact that when channels are excited by signals from a finite alphabet, the noisy observations tend to be clustered around a finite number of so-called channel states. Once these have been estimated (typically by applying any supervised clustering technique based on a known training sequence), channel equalization reduces to a classification problem, which can efficiently be solved using a radial basis function network (RBFN). The time-varying MIMO case produces non-convex clusters shapes that overlap, meaning that conventional clustering algorithms fail. Here, a new version of a spectral clustering technique is utilised to deliver better results than state-of-the-art MIMO decoding techniques for time-correlated channels.

Chapter 13 from Ren et al. considers the routing of traffic in ad hoc networking. A mobile ad hoc network (MANET) consists of a collection of wireless nodes, which dynamically form a temporary network, without using any existing infrastructure of wired networks or centralized administration. Here, the use of swarm intelligence (SI) to tackle traffic routing is the central thrust. Following a comparison between existing methods, SILS, a hybrid routing protocol for MANETs based on SI is presented. The lack of link stability, arising from the constant node movement, is identified as key to the optimisation of a routing protocol. Preliminary results show that SILS performs well in comparison with the popular routing techniques employed today. Further work to optimise the parameters of SILS and to introduce GA assisted methods is discussed.

Section IV – Antennas

The design of the antennas needed for modern communication systems offers substantial opportunity to apply IS methods. Traditionally, an approximate design has been produced and then refinements made by empirical methods and laboratory tests. IS offers the prospect of producing designs that are close to finished, requiring only minimal final adjustments in the field. A flavour of the range of work in this field is presented within this section.

The synthesis of antenna arrays provides the fourteenth chapter, by González-Ayestarán and Las-Heras. They address the design problem of calculating the feeding values that must be applied to the ports of an antenna array in order to obtain a specified radiation pattern, known as synthesis. This problem is typically solved using antenna models that do not take into account the real properties of the elements of the array, assuming a certain degree of error in the solution because of the complexity of developing accurate models able to include the real antenna behaviour in the different synthesis schemes. Support vector regression (SVR) can be used to develop such accurate models from experimental or simulated data accounting for all the real properties of the elements of the array. Furthermore, these SVR-based models can be easily included in different synthesis schemes or in novel SVR-based schemes able to solve synthesis problems with specifications given in different manners. Support vector multi-regression (SVMR) can also be used to develop efficiently 3D models of the arrays. SVR is shown to be a flexible and powerful tool for the modelling and synthesis of array antennas.

In Chapter 15, Randazzo and Pastorino review the use of SVMs for the estimation of the directions of arrival of electromagnetic waves. Determining the incident angles of waves impinging on receiving antennas is of fundamental importance not only for the development of smart antennas for wireless communications but also in many other applications related to imaging and security. The main advantage of the SVM based approach is related to the ability of the system to determine incident angles not included in the original training set. This property is a consequence of the excellent generalization capability of SVMs, which have been proven in several applications related to antennas and in other areas of applied electromagnetics. The chapter presents a brief review of the mathematical formulation of the estimation algorithm and then discusses recent relevant results from the literature.

Kernel Methods for smart antennas by Martínez-Ramón et al. are the focus of this the sixteenth chapter. Adaptive array processing improves performance of receivers in time varying scenarios. Linear methods provide easy to implement, adaptive and low computational burden solutions. For that reason, methods such as least mean squares (LMS) and recursive least squares (RLS) are widely used in practice. Nonlinear algorithms are usually impractical due to their high complexity and because they often need large training datasets for their parameters to be adjusted. Also, it may be hard to obtain adaptive versions of

them. Kernel methods, together with SVMs provide nonlinear, low complex machines that have good generalization performance. Here we present a nonlinear and complex-valued SVM that is applied to antenna array processing, and its adaptive version, and we compare it with the standard methods in several multiuser varying scenarios.

Section V – Medical diagnosis/imaging

Medical data are well known to be noisy in the sense that there are often incomplete records, wrongly entered numbers or some other outcome of the process of human interaction in a hospital setting. Historically, health professionals have applied medical expertise to make diagnostic decisions in the presence of imperfect information. Modern IS methods are now available to aid this process and some of these are presented in this section.

Chapter 17, by Empson et al., outlines an approach to find optimal clusters in patients who have non-specific low back pain (NSLBP). The results of SOM training using clinical data from a study are presented. Clustering is then performed using both the k-means (KM) and FCM clustering algorithms to enable extraction of the patient subgroup memberships from the SOM. The clustering results are shown to have better agreement than when the data is clustered directly, and the clusters found fall into clinically recognizable groups.

The eighteenth chapter, from Alonso-Atienza et al., tackles the detection of ventricular fibrillation in its early stages to substantially reduce the risk of sudden cardiac death. This, in turn, allows ‘recovering shock therapy’ to be administered in a timely manner. A number of classification methods have previously been proposed, to characterise ventricular fibrillation by temporal, spectral or time-frequency measurements. There is thus a large amount of information about arrhythmia conveyed by the variables present. Here, special attention is given to the feature selection problem to discover the most informative features to allow effective pathology discrimination. There is a summary of filter and wrapper methods from the literature, and a comparative analysis of several representative databases. The results obtained are good, and show that significant information about ventricular fibrillation may be retrieved by a moderate number of features. Further work on increasing the detection accuracy of these schemes, and to establish a clinical correspondence of these relevant variables is also discussed.

The provision of a decision support tool for the anaesthetist, and modelling the knowledge of the anaesthetist in evaluating patients during pre-operative assessment are the subjects of the final chapter, by Folland, Hines and Morgan. An anaesthetist must decide prior to a surgical operation whether the general physical state of the patient is robust enough for the stresses that are a natural part of surgery. The work is based on an evolutionary fuzzy mixture model (EFMM) based system that encapsulates the decision strategy of an anaesthetist in determining patient suitability for surgical anaesthesia. Methods for automatically identifying cardio-respiratory pathology and for fusing this information with other patient information are demonstrated. It is shown that a genetic program can evolve the fuzzy system such that classification accuracies in excess of ninety per cent can be attained.

In summary, the chapters on IS techniques and their application to a range of problems provides a representative selection of the available methods and their effectiveness in real domains. While the field is rapidly evolving with the availability of new data and new tools, these chapters clearly indicate the importance and potential benefit of synergistically combining the potentials of the various techniques in order to be better able to face new challenges. The book will be useful to graduate students and researchers in computer science, bioinformatics, computational and molecular biology, electrical engineering, system science and information technology; both as text and reference book for some parts of the curriculum. Moreover, researchers and practitioners in diverse industries, and in research and development laboratories will also benefit from this book.

We take this opportunity to thank all the authors for contributing chapters related to their current research work that provide the state of the art in advanced methods for solving such problems.

*Evor Hines
Mark Leeson
Manel Martínez-Ramón
Matteo Pardo
Eduard Llobet
Daciana Iliescu
Jianhua Yang
June, 2008*

ACKNOWLEDGEMENTS

We would like to thank our respective universities for their encouragement and support to produce this book. We would also like to thank Jianhua Yang for his tireless efforts in the design work on the book cover and in the typesetting of this manuscript; and to Vera van Ratingen of Shaker Publishing BV for her support in the production of this book.

In addition last but not least we would like to thank our respective families and friends for their patience, understanding and support during the preparation of this manuscript.

Evor Hines
Mark Leeson
Manel Martínez-Ramón
Matteo Pardo
Eduard Llobet
Daciana Iliescu
Jianhua Yang

Table of Contents

Preface	v
Acknowledgements	xv
List of Contributors	xxi

Section I: Intelligent Systems and Olfaction

Chapter 1: Optimising the Number of Electronic Nose Sensors Needed using Genetic Algorithms and Neural Networks	3
<i>Jianhua Yang, Evor L. Hines, Daciana D. Iliescu, Mark S. Leeson and Pascal Boilot</i>	
Chapter 2: A Comprehensive Study of Sensor Responses to Fat Food Aroma – Using Genetic Algorithms and Neural Networks: The Particular Case of Virgin Olive Oil	21
<i>Diego L. Garc ía-Gonz ález and Ramón Aparicio</i>	
Chapter 3: Systematically Improving the Performance of Semiconductor Gas Sensors: Optimization of the Temperature Modulation Mode via Pseudo-random Sequences – Using Fuzzy Artmap Neural Networks	53
<i>Alexander Vergara and Eduard Llobet</i>	
Chapter 4: Support Vector Machines for Electronic Nose Pattern Analysis	109
<i>M. Pardo and S. De Vito</i>	
Chapter 5: Dynamic Methods for Improving the Performance of Semiconductor Gas Sensors	133
<i>Alexander Vergara, Xavier Vilanova and Eduard Llobet</i>	

Section II: Intelligent Systems and Engineering

Chapter 6: Multi-input Optimisation of River Flow Parameters and Rule Extraction Using Genetic-Neural Technique	173
<i>Jianhua Yang, Evor L. Hines, Ian Guymmer, Daciana D. Iliescu and Mark S. Leeson</i>	
Chapter 7: Kernel Machines in Remote Sensing Image Classification	199
<i>Gustavo Camps-Valls, J. Muñoz-Marí L. Gámez-Chova and J. Calpe-Maravilla</i>	
Chapter 8: Dynamic Error Recovery in the ATLAS TDAQ System: Using Multi-Layer Perceptron Neural Networks.	223
<i>John Erik Sloper, Giovanna Lehmann Miotto and Evor L Hines</i>	
Chapter 9: Discriminating Fiber-Reinforced Plastic Signatures Related to Specific Failure Mechanisms: Using Self-Organizing Maps and Fuzzy C-Means	249
<i>XuQin Li, Carlos Ramirez, Evor L. Hines, Mark S. Leeson, Phil Purnell and Mark Pharaoh</i>	

Section III: Intelligent Systems and Telecommunications

Chapter 10: Optimization of Cellular Radio Network using Genetic Algorithm and Fuzzy Logic	287
<i>Jie An, Evor L. Hines and Mark S. Leeson</i>	
Chapter 11: Design of Survivable Dense-Wavelength-Division-Multiplexing Optical Mesh Transport Networks Using Genetic Algorithms	319
<i>Yousef S. Kaviani, W. Ren, Evor L. Hines, Mark S. Leeson, Habib F. Rashvand and Majid Naderi</i>	
Chapter 12: A Spectral Clustering Approach for Blind Decoding of MIMO Transmissions over Time-Correlated Fading Channels.	351
<i>Steven Van Vaerenbergh and Ignacio Santamaría</i>	

Chapter 13: Swarm Intelligence Based Link Stability Routing Optimisation Algorithm for Mobile Ad hoc Network	379
<i>Wei Ren, Evor L. Hines, Mark S. Leeson, Yousef S. Kavian and Daciana Iliescu</i>	

Section IV: Intelligent Systems and Antennas

Chapter 14: Support Vector Regression for the Modeling and Synthesis of Antenna Arrays	399
<i>Rafael González-Ayestarán and Fernando Las-Heras</i>	
Chapter 15: Support Vector Machines for the Estimation of the Directions of Arrival of Electromagnetic Waves	423
<i>Andrea Randazzo and Matteo Pastorino</i>	
Chapter 16: Kernel Methods for Smart Antennas.	441
<i>M. Martínez-Ramón, A. Navia-Vázquez, R. Jordán and C. G. Christodoulou</i>	

Section V: Intelligent Systems and Medical diagnosis/imaging

Chapter 17: A Method of Clustering a Self Organising Map Neural Network to Determine Subgroups of Non-Specific Lower Back Pain Patients.	459
<i>Sam Empson, Chris McCarthy, Ranjit Lall, Evor Hines, Mark Leeson and Daciana Iliescu</i>	
Chapter 18: Learning the Relevant Features of Ventricular Fib-rillation from Automatic Detection Algorithms	505
<i>Felipe Alonso-Atienza, Alfredo Rosado-Muñoz, JoséLuis Rojo-Álvarez and Gustavo Camps-Valls</i>	
Chapter 19: Evolutionary Fuzzy Mixture Model based Decision Support System for an Anaesthetist.	535
<i>Ross Folland, Evor Hines and David Morgan</i>	

Index.	561

List of Contributors

A. Navia-Vázquez

Dept. of Signal Theory and
Communications
Universidad Carlos III de Madrid
Avda Universidad, 30, 28911
Leganés, Madrid, Spain
manel@tsc.uc3m.es

Alexander Vergara

Associate Researcher
MINOS
Department of Electronic
Engineering
University Rovira i Virgili
alexander.vergara@urv.cat

Alfredo Rosado-Muñoz

Dpt. Enginyeria Electrònica
Univ. València, Spain
alfredo.rosado@uv.es

Andrea Randazzo

Department of Biophysical and
Electronic Engineering
University of Genoa
Via Opera Pia 11A - 16145 Genova

Italy

randazzo@dibe.unige.it

C. G. Christodoulou

Dept. of Electrical and Computer
Engineering
The University of New Mexico
1, University of New Mexico
Albuquerque, NM 87131-0001
USA

Carlos Ramirez

International Automotive Research
Centre
Warwick Manufacturing Group
University of Warwick, UK
carlos.ramirez@warwick.ac.uk

Christopher McConkey

Research Fellow, Senior Statistician
Health Sciences Research Institute
Medical School Building
Gibbet Hill Campus
University of Warwick
Coventry, CV4 7AL, UK
c.j.mccarthy@warwick.ac.uk

Daciana D. Iliescu

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
d.d.udrea@warwick.ac.uk

David Morgan

Safe Surgery Systems Ltd.
Lincoln House, Birmingham
Heartlands Hospital
Bordesley Green East, Birmingham
B9 5SS, UK

Diego L. García-González

Instituto de la Grasa (CSIC)
dluisg@cica.es

Eduard Llobet

MINOS
Department of Electronic
Engineering
University Rovira i Virgili
eduard.llobet@urv.cat

Evor L. Hines

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
e.l.hines@warwick.ac.uk

Felipe Alonso-Atienza

Dpt. Teoría de la Señal y
Comunicaciones
Univ. Rey Juan Carlos
Madrid, Spain
felipe.alonso@urjc.es

Fernando Las-Heras

Universidad de Oviedo
Departamento de Ingeniería
Eléctrica
Campus Universitario, 33203 Gijón
Asturias, Spain
rayestaran@uniovi.es

Giovanna Lehmann Miotto

CERN, PH Department
giovanna.lehmann@cern.ch

Gustavo Camps-Valls

Dpt. Ingeniería Electrónica
Universitat de València, Spain.
C/ Dr. Moliner, 50. Burjassot
(València), Spain.
gustavo.camps@uv.es

Habib Rashvand

Visiting Professor
School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
h.rashvand@warwick.ac.uk

Ian Guymer

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
i.guymer@warwick.ac.uk

John Erik Sloper

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
j.e.sloper@warwick.ac.uk

Ignacio Santamaría

Department of Communication
Engineering
University of Cantabria
Santander, 39005, SPAIN

José Luis Rojo-Álvarez

Dpt. Teoría de la Señal y
Comunicaciones
Univ. Rey Juan Carlos
Madrid, Spain
joseluis.rojo@urjc.es

J. Calpe-Maravilla

Dpt. Enginyeria Electrònica
Universitat de València, Spain

L. Gómez-Chova

J. Calpe-Maravilla
Dpt. Enginyeria Electrònica
Universitat de València, Spain

J. Muñoz-Marí

Dpt. Enginyeria Electrònica
Universitat de València, Spain

M. Martínez-Ramón

Dept. of Signal Theory and
Communications
Universidad Carlos III de Madrid
Avda Universidad, 30, 28911
Leganés, Madrid, Spain
manel@tsc.uc3m.es

Jianhua Yang

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
j.yang.3@warwick.ac.uk

Majid Naderi

Associate Professor
Department of Electrical
Engineering
Iran University of Science &
Technology (IUST)

Jie An

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
j.an@warwick.ac.uk

Narmak, 16844, Tehran, Iran
m_naderi@iust.ac.ir

Mark Pharaoh

International Automotive Research
Centre
Warwick Manufacturing Group
University of Warwick
Coventry, CV4 7AL, UK
m.w.pharaoh@warwick.ac.uk

Mark S. Leeson

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
m.s.leeson@warwick.ac.uk

Matteo Pardo

Sensor Laboratory
CNR – INFM
Via Valotti 9 25133 Brescia, Italy
pardo@ing.unibs.it

Matteo Pastorino

Department of Biophysical and
Electronic Engineering
University of Genoa
Via Opera Pia 11A - 16145 Genova
Italy
pastorino@dibe.unige.it

Pascal Boilot

Technical Support Engineer
Alpha MOS
boilot@alpha-mos.com

Phil Purnell

Civil and Mechanical Division
University of Warwick
Coventry, CV4 7AL, UK
p.purnell@warwick.ac.uk

R. Jordán

Dept. of Electrical and Computer
Engineering
The University of New Mexico
1 University of New Mexico
Albuquerque, NM 87131
USA

Rafael González-Ayestarán

Universidad de Oviedo
Departamento de Ingeniería
Eléctrica
Campus Universitario, 33203 Gijón
Asturias, Spain
rayestaran@uniovi.es

Ramón Aparicio

Research Professor
Instituto de la Grasa (CSIC)
aparicio@cica.es

Ranjit Lall

Research Fellow Statistician
Health Sciences Research Institute
A1.02 Medical School Building
Gibbet Hill Campus
University of Warwick
Coventry, CV4 7AL, UK
ranjit.lall@warwick.ac.uk

Ross Folland

Safe Surgery Systems Ltd.
Lincoln House, Birmingham
Heartlands Hospital
Bordesley Green East, Birmingham
B9 5SS
UK
ross.folland@safesurgerysystems.com

Sam Empson

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
s.empson@warwick.ac.uk

Saverio de Vito

ENEA, Napoli, Italy
saverio.devito@portici.enea.it

Steven Van Vaerenbergh

Department of Communication
Engineering

University of Cantabria

Santander, 39005, SPAIN

steven.nacho@gtas.dicom.unican.es

Wei Ren

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
w.ren@warwick.ac.uk

Xavier Vilanova

Electronics Department of
Electronic Engineering
University Rovira i Virgili
Avda. Països Catalans, 26, 43007
Tarragona, Spain
xavier.vilanova@urv.cat

XuQin Li

School of Engineering
University of Warwick
Coventry, CV4 7AL, UK
xu-qin.li@warwick.ac.uk

Yousef S. Kavian

Faculty of Engineering
Chamran University of Ahvaz
(CUA)
Ahvaz, Iran

Section I: Intelligent Systems and Olfaction

Chapter 1

Optimising the Number of Electronic Nose Sensors Needed using Genetic Algorithms and Neural Networks

Jianhua Yang, Evor L. Hines, Daciana D. Iliescu, Mark S. Leeson and Pascal Boilot

1.1 INTRODUCTION

Odour classification systems used in machine olfaction, which are often called electronic noses (ENs), have been gaining favour in a wide range of industrial applications (Hines, Llobet et al. 1999; Llobet, Hines et al. 1999; Gardner, Shin et al. 2000; Dutta, Hines et al. 2003). An EN is a device that is designed to detect and discriminate among complex odours using a sensor array (Pearce, Schiffman et al. 2003). Typically, it comprises three main functional components: a sampling unit, a signal processing unit, and an odour classification unit (Phaisangittisagul and Nagle 2007). The sampling unit, which is analogous to biological olfactory receptors, typically consists of for example an array of gas sensors. Unlike traditional analytical methods, EN sensor responses do not provide information on the nature of the compounds under investigation, but only give a ‘digital fingerprint’ of the odour, which can be subsequently investigated by means of data processing methods (Ulivieri, Distant et al. 2006). Thus, the composition of the sensor array of an electronic olfactory system is a fundamental choice which impacts significantly on the effectiveness of the overall system.

Recent advances in the field of ENs have led to new developments in sensors, feature extraction (pre-processing), and data processing techniques (Gardner, Boilot et al. 2005). As a result, the user of EN systems is provided with an

increased amount of information for the discrimination of odours when using multi-sensor arrays. However, even if each sensor is linked to specific classes of compounds, not all the sensors contribute to the characterisation of the odour which is being analysed (Ballabio, Cosio et al. 2006). Furthermore, not all of them are relevant to the particular pattern recognition (PARC) classification task. This problem is known as the curse of high dimensionality (Gardner, Boilot et al. 2005). The aim of sensor selection is to eliminate redundant, noisy, or irrelevant sensors and thus find an optimal subset from an array of high dimensionality. By optimising the array size, the overall system performance can potentially be increased by maximising the information content and hence increasing the predictive accuracy.

PARC analysis has been reported in the literature that determines an optimal subset of sensors for machine olfaction. For example, Gardner et al. (2005) introduced a modified genetic algorithm (GA) called V-integer GA. In this V-integer GA, each chromosome was used with integer values from one to a pre-defined number of features/sensors representing the selected subset of features, and evaluated using probabilistic neural network (PNN) classifiers within the population. They also compared this V-integer GA with other search methods such as sequential forward or backward searches (SFS or SBS) and normal (binary) GAs. For the dataset used in their work, SFS achieved over 89% correct classification by selecting just three features, whereas SBS needed at least five features to reach the same level. With binary genes GAs, the dimensionality is reduced by 50–60% and the classification rates are on average 91%. Considering eight, six or four features, the optimal subsets returned by the V-integer genes GA selections have dimensionality reduced by over 80% and on average achieve around 90% correct classification. These results showed that the V-integer genes GA approach is an accurate search method when compared to some other feature selection techniques such as SFS or SBS. However, in the V-integer GA, the number of sensors to be selected has to be defined in advance – in other words, there is potentially a lack of flexibility in some application scenarios.

On the other hand, Ballabio, Cosio et al. (2006) suggested a chemometric approach based on a partial ordering technique and the Hasse matrix. In this approach, the Hasse matrix can be obtained from each EN data sequence and the similarity between two sequences can then be evaluated with the definition of a distance between the corresponding Hasse matrices. Since all the signals which are temporarily selected are intrinsically ordered, the data provided by the EN can also be considered to be sequential data and can consequently be characterized as such. In this way, a similarity/diversity measure can thus be applied in order to characterize the class discrimination capability of each EN sensor. The distance based on the Hasse matrix is then used to link each EN time

profile to a meaningful mathematical term (the Hasse matrix), which can be subsequently explored using multivariate analysis. However, in this model there is an absence of a proper classifier. The consequence of this is that the results of sensor selection are not comparable. In their case study, two sensors were selected out of a total of 15 to distinguish two features. This result was also confirmed by principal component analysis (PCA). However, if the number of features increases, PCA may not be able to handle the problem and thus the whole method may fail to work efficiently.

Recently, a Spanish research group (Gualdron, Llobet et al. 2006; Llobet, Gualdron et al. 2007) have reported successful techniques for EN sensor selection. In the first case (Gualdron, Llobet et al. 2006), by evaluating different variable selection techniques (including deterministic and stochastic methods) coupled with neural network-based classifiers, they proposed a two-step strategy for sensor selection: a coarse selection based on a variance criterion followed by a simulated annealing (SA) process based on either the fuzzy ARTMAP or the PNN. As a result, a success rate of 91.66% in simultaneous identification was obtained using only nine input variables (out of the 120 available) in their application. However, in this approach when computing the variance of each sensor, dependence (linear and/or nonlinear) between two sensors was not considered, and as such the selected subset may still contain redundant features and thus may not be the optimal subset. In the second case (Llobet, Gualdron et al. 2007), a three-step strategy for feature selection was presented: the first two steps were aimed at removing noisy, non-informative and highly collinear features; the third step makes use of a stochastic variable selection method (SA) to further reduce the number of variables. However, in this approach the threshold values for the discrimination ability and collinearity were both set heuristically. Therefore it is still possible that irrelevant sensors are not filtered out; and vice versa.

The current chapter is concerned with the application of a GA-based pattern classifier to the problem of optimal EN sensor selection (GNMM, Genetic Neural Mathematical Method) (Yang, Hines et al. 2007; Yang, Hines et al. 2007; Yang, Hines et al. 2008). Further details about this GNMM technique can be found in the chapter titled “A Genetic Algorithm-Based Input Determination Method for Neural Networks in the Prediction of Longitudinal Dispersion Coefficient in Rivers”. This current chapter is organized as follows: in the next section we will give a brief description of different Intelligent Systems (ISs) based PARC techniques that can be used in the field of EN data classification. Through this description we will show that the combination of GA and MLP is a natural choice; after that, we will present an introduction of GNMM and apply it to an

EN dataset collected from bacteria that cause eye infection; finally we make our conclusions.

1.2 INTELLIGENT SYSTEMS (ISS) BASED PATTERN RECOGNITION (PARC) FOR ELECTRONIC NOSES

PARC has been a research topic for many years and excellent introductory books are available; see for example Bishop (2006) and Theodoridis and Koutroumbas (2006). This section provides a short overview of different ISs based PARC techniques that have been implemented to analyse EN data. A detailed description of these methods can be found in (Hines, Boilot et al. 2003) and a recent review was given in (Scott, James et al. 2006).

Table 1: Classification of ISs-based PARC techniques

	ANN	Fuzzy systems	Evolutionary algorithms
Supervised	MLP, PNN, RBF	FIS, FCM	GA
Unsupervised	SOM	Fuzzy ARTMAP	

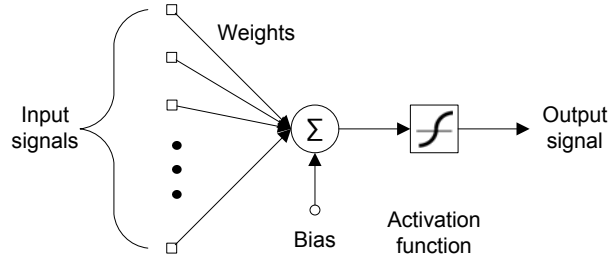


Figure 1: Illustration of an artificial neuron

Compared with statistical/chemometric methods (e.g. PCA), which are parametric based on the assumption that the spread of the sensor data can be described by probability density function (PDF), ISs-based PARC techniques offer advantages such as learning capabilities, self-organization, generalization and noise tolerance. According to the biological inspiration that each model is built on, these techniques may be split into three sub-groups: artificial neural networks (ANNs), fuzzy systems and evolutionary algorithms. Another criterion to group these techniques may be based on the way in which the training pattern is being fed into the system: either supervised, unsupervised or reinforcement. The classification of some commonly used techniques includes for example

multi-layer perceptron (MLP), probabilistic neural network (PNN), radial basis function (RBF), self-organizing map (SOM), fuzzy inference systems (FIS), fuzzy c-means (FCM), fuzzy ARTMAP, and genetic algorithms (GAs), see Table 1.

1.2.1 Artificial Neural Networks (ANNs)

Figure 1 is a representation of an artificial neuron. An artificial neuron is a model of a biological neuron. Each neuron typically receives signals from outside, or from other neurons. When fired, these neurons compute a net input signal as a function of the respective weights. The net signal serves as input to the activation function using which the neuron then determine the output signal.

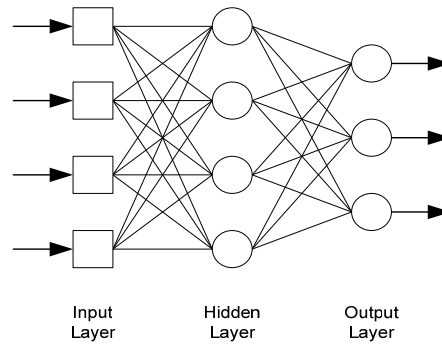


Figure 2: A three-layer MLP

An ANN is a layered network of artificial neurons. It is characterized by three main properties: the activation function, the network topology, and the learning algorithm (Haykin 1999). The MLP is the most popular arrangement of neurons in odour classification. An MLP usually consists of three layers: the input layer, the hidden layer and the output layer, as shown in Figure 2. The number of input neurons is typically determined to correspond to the number of sensors in the array. The number of neurones in the hidden layer is determined experimentally and the number of odours to be classified generally determines the number of output neurons. The most popular method for training the MLP is based on minimisation of a cost function and is known as the back-propagation (BP) algorithm (Engelbrecht 2007). In BP, learning iterations (i.e. epochs) consists of two phases: in the feedforward pass, the actual output values of the MLP for each

training pattern are calculated; in the backward propagation, any error signal is propagated back from the output layer toward the input layer. Weights are then adjusted as functions of the backpropagated error signal.

RBF networks are supervised learning paradigms very similar to MLP except that they use radial basis transfer functions for the hidden layer rather than linear or sigmoid ones. The RBFs operational principle is that it paves the input space with overlapping receptive fields, as they classify data using hyper-spheres rather than hyper-planes (Lin and Lee 1996). The RBF network is basically trained by a hybrid learning rule: unsupervised learning in the input layer and supervised learning in the output layer. Hence RBF implementations differ mainly in the choice of heuristics used for selecting basis function centres and widths.

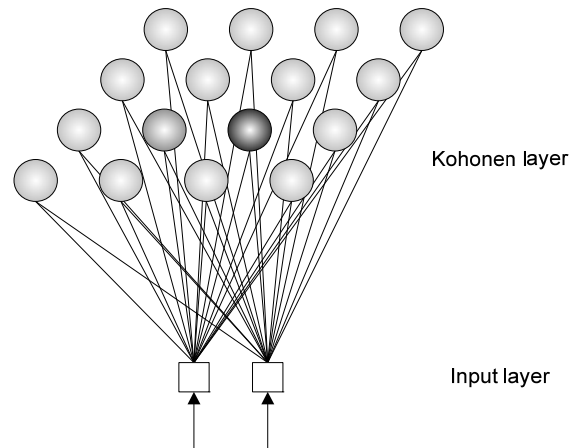


Figure 3: Self-organising map with one Kohonen layer

PNNs also have layered structures as in MLPs. Unlike MLPs or RBFs, they usually consist of four layers: an input layer stores the new pattern while it is serially passed through the hidden layer; a hidden layer computes the Euclidean distance between the new pattern and the training set pattern; a pattern/summation layer consists of one neuron for each data class and sums the outputs from all hidden neurones of each respective data class; and finally a decision layer where the estimated probability of the new pattern being a member of that data class is computed. By using a Gaussian activation function in the hidden layer, a PNN can compute nonlinear decision boundaries which approach the Bayes optimal (Specht 1988).

The SOM is a feed forward unsupervised learning network (Kohonen 2001). SOM applied to EN systems typically contain a two-dimensional single layer of

neurons in addition to an input layer of branched nodes, as in Figure 3. SOM training is based on a competitive learning strategy: measured based on the Euclidean distance, the best neuron learns by shifting its weights from inactive connections to active ones. Neurons close to the winner are also updated according to the neighbourhood relationships. In this way, SOMs effectively cluster the input vectors through a competitive learning process, while maintaining the topological structure of the input space.

The main potential drawbacks of the MLP-BP networks are that they quite often suffer from becoming stuck in a local minimum and they may require long learning periods in order to encode the training patterns. This also holds for training the ordinary RBF and SOM networks, since the network parameter tuning method resorts to a gradient-descent type algorithm, which normally requires iterative and long training (Hoya 2005). In comparison, PNN does not require iterative training of the weight vectors. This is probably the reason why PNN was used in (Gualdron, Llobet et al. 2006; Llobet, Gualdron et al. 2007). On the other hand, however, the PNN is typically limited to applications involving relatively small datasets; large datasets would lead to large network architectures and may increase the rate of misclassification (Scott, James et al. 2006).

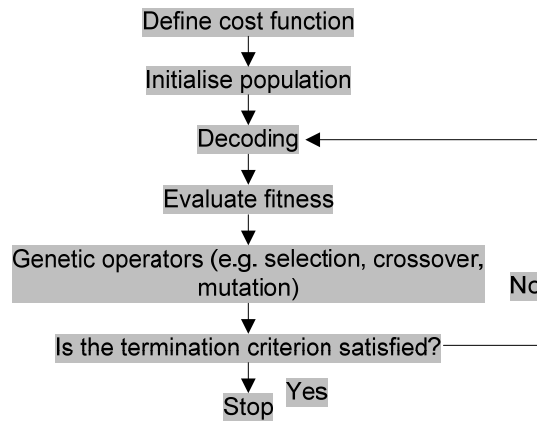


Figure 4: Flowchart of a typical GA

1.2.2 Fuzzy Systems and Evolutionary Algorithms

Fuzzy logic is a tool for embedding structured human knowledge into workable algorithms (Kecman 2001; Bezdek 2005). Fuzzy logic methods can be used to design ISs on the basis of knowledge expressed in a common language. In the

field of machine olfaction, it is attractive when odour samples are described by olfactory descriptors, such as peppery, floral, or sweet, and intensity attributes, such as quite, very, or strong.

GAs are a powerful optimisation technique, based upon the underlying principles of natural evolution and selection (Holland 1992). The main procedures in the GA optimisation process are shown in Figure 4. The basic idea is to maintain a population of chromosomes, representing candidate solutions to the problem being solved. The possible solutions are generally coded as binary strings and these strings are equivalent to biological chromosomes. Each bit of the binary string (chromosome) is referred to as a gene. A GA starts off with a population of randomly generated chromosomes and advances towards better chromosomes by applying genetic operators that are based on genetic processes occurring in nature (i.e. selection, crossover and mutation) (Mitchell 1996; Haupt and Haupt 2004). This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals from which they were created.

GAs are well suited to the task of selecting an appropriate combination of inputs to a model as they have the ability to search through large numbers of combinations, where interdependencies between variables may exist (Reeves and Rowe 2003; Rothlauf 2006). Another popular stochastic optimization method called simulated annealing (SA), has been used in (Gualdron, Llobet et al. 2006; Llobet, Gualdron et al. 2007). Although SA is not biological based, GA and SA share very similar theoretical roots (Davis 1987). However, GAs are population-based approaches, where there is the concept of competition (i.e. selection) between candidate solutions to a given problem. Furthermore, SAs generate new candidate points in the neighbourhood of the current point, while GAs allow the examination of points in the neighbourhood of two (or more) candidate solutions via the use of genetic operators such as crossover (De Castro 2006). Therefore, GA tends to improve the solution consistently when given more time.

1.3 THE GENETIC NEURAL MATHEMATICAL METHOD (GNMM)

From the descriptions in the previous section it is obvious that the PARC techniques have relative strengths and weaknesses depending on the application being considered. Therefore, it is of interest to explore hybrid methods which for example incorporate two (or more) techniques and thus hopefully takes advantage of them. Here we utilize a GA-based input determination method for ANNs called GNMM (Genetic Neural Mathematical Method) (Yang, Hines et al. 2007; Yang, Hines et al. 2007; Yang, Hines et al. 2008). A detailed discussion of

GNMM and its applications in the field of civil engineering is presented in the chapter titled “A Genetic Algorithm-Based Input Determination Method for Neural Networks in the Prediction of Longitudinal Dispersion Coefficient in Rivers”. In the rest of the chapter, we will firstly give a brief introduction to GNMM in the context of EN data feature selection, and then apply the method to an EN dataset to demonstrate its usefulness.

1.3.1 The Curse of Dimensionality and GNMM

As mentioned in the section Artificial Neural Networks (ANNs), the MLP is the most popular arrangement of neurons in odour classification. However, as the number of sensors in an EN array increase, the number of weights an MLP optimize during training grows exponentially. For example, a fully connected array of 10 sensors with 10 neurons in the hidden layer classifying 6 different odours would need 160 weights to be learnt. If the number of sensors in this MLP increases by 1, the weights would increase by 10. On the other hand, increasing the dimensionality rapidly leads to the point where there may not be enough training data for the MLP to be trained optimally, in which case the MLP may provide a very poor representation of the input/output mapping. This is the phenomenon often referred to as the curse of dimensionality (Bishop 1995). Thus, the objective of any sensor selection algorithm should be to reduce the dimensionality and also realise optimum PARC accuracy.

The basic idea behind GNMM is that GAs serve as a wrapper technique for MLPs to select optimal sensor combinations as inputs; while MLPs perform the classification tasks and provide evaluations for each sensor combination. In the GA process, as shown in Figure 4, firstly an initial population of chromosomes was randomly generated, in which each chromosome consists of as many genes as there are elements in the sensor array. The encoding of a gene is binary, meaning that a particular sensor is considered as an input variable (represented by ‘1’) or not (represented by ‘0’). The assessment of the fitness of a chromosome uses the Mean Squared Error (MSE) where a three-layer MLP is being trained with the input sensor subset and the output target for a relatively small number of epochs. For simplicity the encoding of the target is decimal instead of binary, which means that there will always be only one neuron in the output layer of the MLP. The GA optimisation process is then realised by altering MLP training epochs and the GA’s population size, generation size, crossover probability and mutation probability. As a result, sensors which occur most

frequently throughout all the populations can therefore be identified. The final sensor subset formed by these sensors is the subset that produces the minimal classification error. This is the first step in GNMM, i.e. sensor selection. Once input sensors are determined, GNMM processes the EN data using an MLP-BP for a much larger number of epochs. At this time, the MLP's internal weights are finely adjusted in an attempt to better model the input/output relationship and thus to improve the classification accuracy.

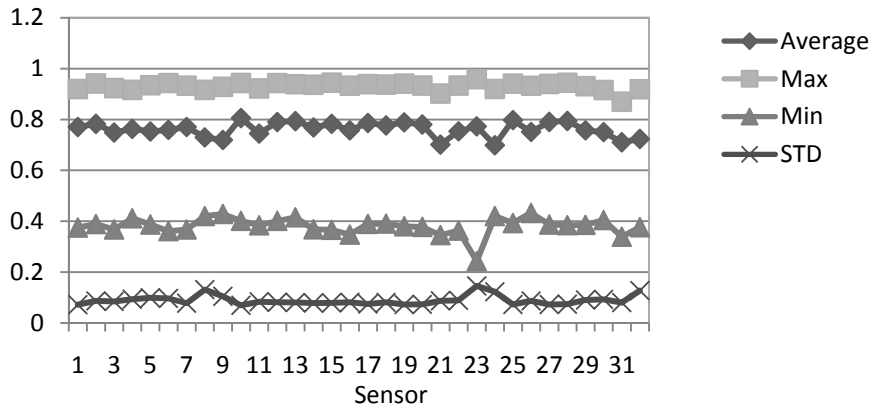


Figure 5: Statistics of the dataset

Due to the randomness in the GA and MLP initialization, there is uncertainty associated with the GAs and MLPs. This means that a sensor combination's performance may not stay constant. In GNMM, this problem is solved by applying two techniques: one is introducing an elite group into the GA; that is a proportion of the collection of chromosomes that performed best and were made exempt from genetic operators are retained in the next generation (Reeves and Rowe 2003; Haupt and Haupt 2004). The other is the concept of *appearance percentage*, which is defined as the average of the chance that a sensor appears in the winning combination (chromosome). In addition to sensor/feature selection and classification, GNMM also offers a mechanism to extract regression rules from a trained MLP. This is realised by dividing the input space into four sub-domains. For a detailed description of GNMM, please see chapter "A Genetic Algorithm-Based Input Determination Method for Neural Networks in the Prediction of Longitudinal Dispersion Coefficient in Rivers".

1.3.2 The Dataset

The EN dataset used as the case study has previously been investigated in (Boilot, Hines et al. 2002; Gardner, Boilot et al. 2005). The data were collected using a Cyranose 320 EN to sample three dilutions of six eye bacteria species. The EN comprises of an array of 32 sensors, and each dilution of these six bacteria was measured ten times. This gives a total of 180 samples belonging to six categories. For details about bacteria that cause eye infections and the experimental protocol/methodology, please refer to (Boilot, Hines et al. 2002).

Statistics of the dataset are shown in Figure 5, in which STD stands for standard deviation and is calculated according to $\sqrt{\sum(x - \bar{x})^2 / (n - 1)}$, where x is the data samples for each sensor and n is the total number of samples i.e. 180. It can be seen that in Figure 5 the maximum value for each sensor varies within a small range. This is because all the signals were produced by the same type of carbon black polymer composite resistors. However, the minimum values have a bigger variation and so have the mean values, due to the fact that EN sensors react differently to different odours. This feature helps to distinguish odours from the EN data. It is noticeable that the STD of sensor 8, 23, 24 and 32 are considerably higher than others. These findings may indicate sensors that would appear in the optimal subset.

1.3.3 Results and Discussions

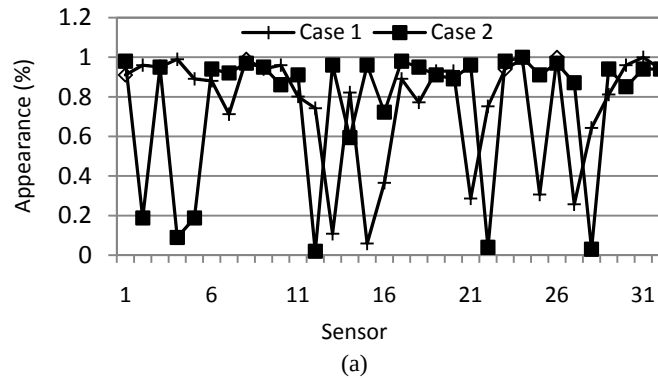
Table 2: GNMM configurations for the EN data

Case	Population	Generation	Epochs
1	30	100	20
2	30	100	50
3	30	150	20
4	30	150	50
5	30	200	20
6	30	200	50

GNMM was implemented using MATLAB version 7.2 on a Sun workstation equipped with UltraSPARC III (900 MHz) CPUs. As suggested in the literature (Schaffer, Caruana et al. 1989), a relatively small population size and large mutation/crossover rate can achieve thorough search in the search space. Thus the population size was kept small (30) for all GNMM runs and the mutation/crossover rate was set to be 0.8/0.01 respectively. For the MLP classifier training, the Levenberg-Marquardt Algorithm (a fast variation of BP)

was used, and the learning rate was set to be 0.01. For the EN data used in this case study, GNMM was run six times. Other configurations of GNMM can be seen in Table 2.

Defining a sensor's appearance percentage as a sensor's accumulated appearance in the winning chromosome of each generation divided by the total number of generations, the appearance percentage of these 32 sensors in each of these six cases is shown in Figure 6 (a), (b) and (c), and the mean appearance percentage is shown in (d). Apart from illustrating each sensor's ability to be selected in the winning chromosome (i.e. the sensor subset performing the most accurate classification), Figure 6 also demonstrates the importance of repeating the GA's optimization processes. For example, sensor 6 performed quite well in case 1, 2, 4 and 6; however, this is not the case for case 3 and 5. By calculating an averaged 'appearance', we now know that the chance of sensor 6 being picked in the optimal sensor subset is quite low (~60%). On the other hand, sensor 19 approaches zero in case 3 and 5. But from Figure 6 (d) it can be seen that sensor 19 is not the worst one. To summarise Figure 6, by calculating the appearance percentage of each sensor, we smooth out the curve formed from a single GNMM implementation, and thus minimize the randomness associated with GA and MLP.



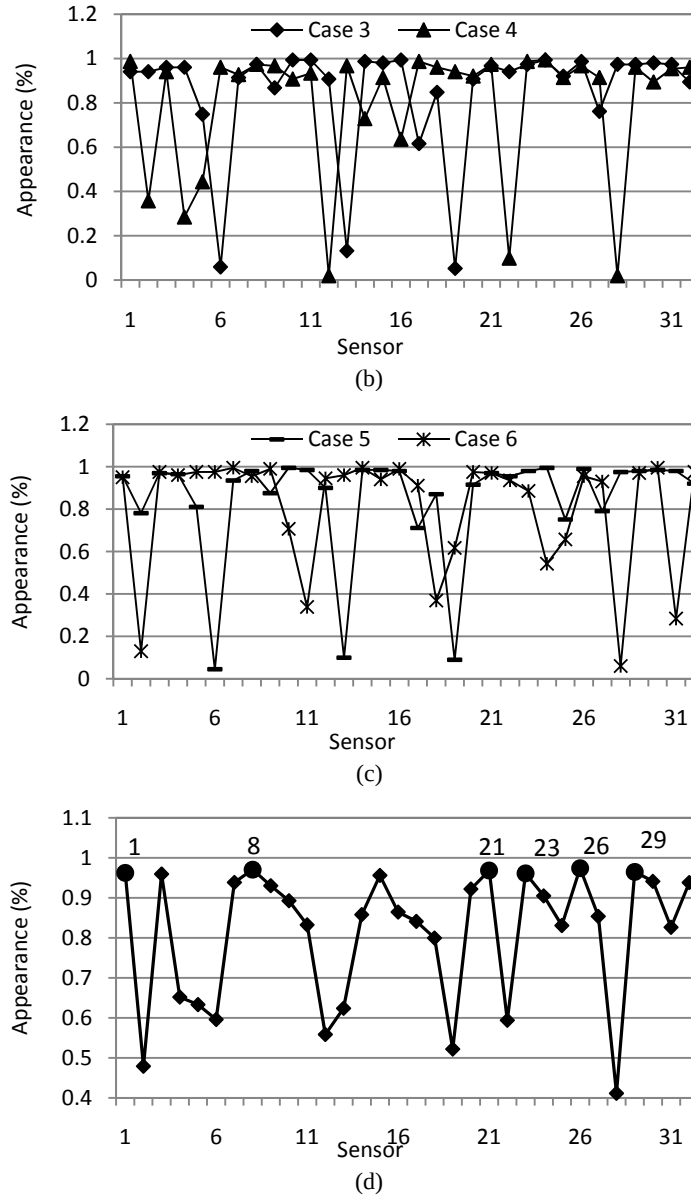
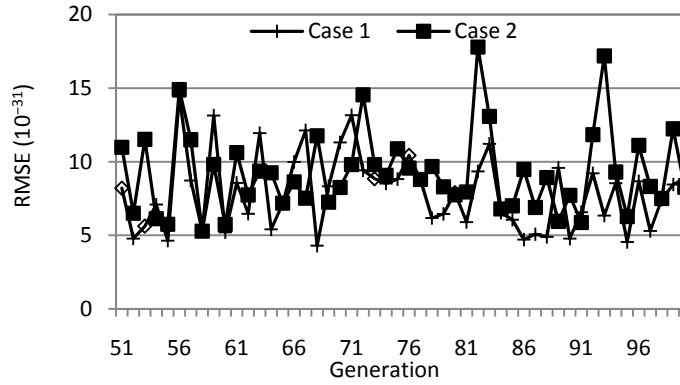
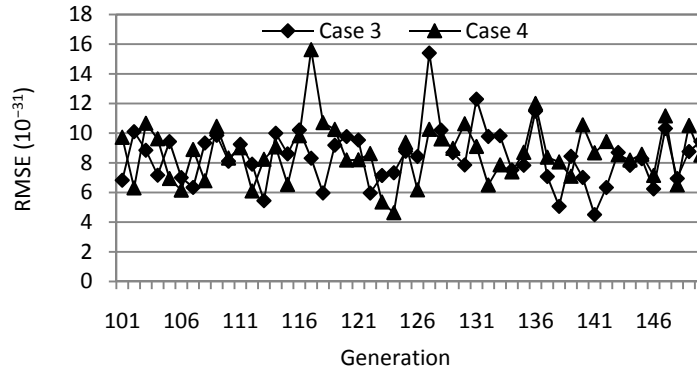


Figure 6: Appearance of each sensor in one case ((a), (b) and (c)) and the mean appearance for all cases ((d))

Figure 6 (d) also indicates that the best sensors are not quite distinguishable. However, a line can still be drawn to select the most important ones. By drawing a line at *appearance percentage* = 95%, we identified 6 sensors, which are illustrated by solid black dots in Figure 6 (d). These sensors and their mean *appearance percentage* are: 1 (96%), 8 (97%), 21 (97%), 23 (96%), 26 (97%) and 29 (96%). Reviewing the STDs of this sensor array, where sensor 8, 23, 24 and 32 are considerably higher than others, it seems reasonable that some sensors (8 & 23) which contain the most diverse data were included in this optimal subset.



(a)



(b)

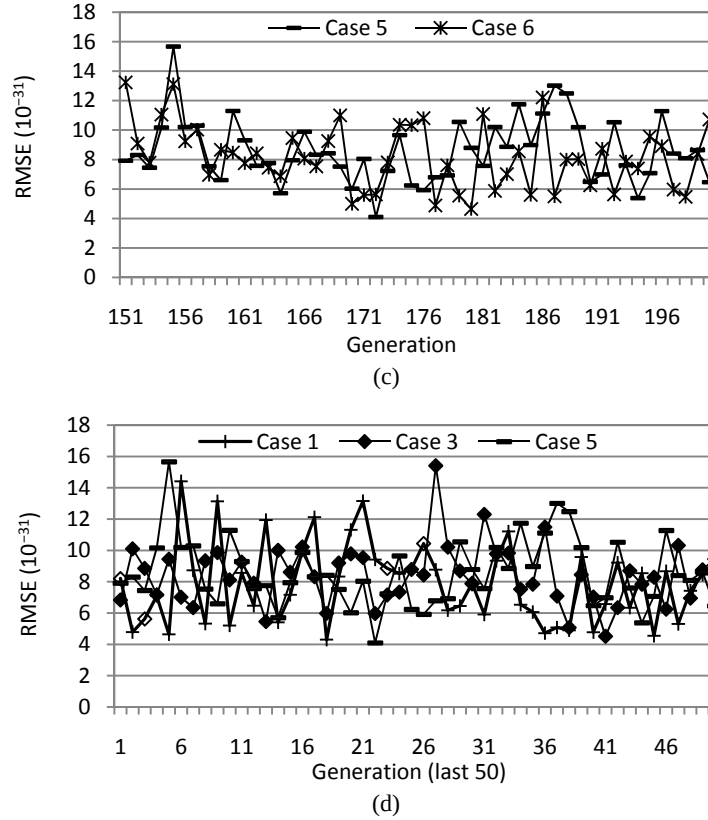


Figure 7: Comparisons of the last 50 generations' RMSE for each case

Figure 7 provides comparisons of the RMSE for the last 50 generations for each case. These comparisons can help us understand GNMM's optimization process. Given that the other configurations are the same and only training epochs for the MLP classifier vary, which is the case illustrated by Figure 7 (a), (b) and (c), it is evident that larger epochs generally produce smaller values and variation of the optimization error (see Table 2). On the other hand, if the epochs are the same and the generations are different, larger generations normally yield lower error as in Figure 7 (d). However, the most outstanding feature that can be found from Figure 7 is that later generations do not necessarily generate better performance. This finding, from another point of view, validates the importance of the mean appearance percentage.

Once the optimal subset is found, GNMM's implementation goes into the next stage – classification. In the first step, the whole EN data were randomly divided into two subsets: one for training and one for validation. As a result, GNMM achieved 100% accuracy for both subsets. In the second step, the data were randomly divided into three equal subsets: one for training, one for testing, and one for validation. Once again, GNMM achieved 100% classification rate in recognizing the training and validation subsets. For the testing set, a 93% accuracy was achieved.

The optimal subset selected by GNMM ({1, 8, 21, 23, 26, 29}) is different compared with results obtained by (Boilot, Hines et al. 2002; Gardner, Boilot et al. 2005) ({8, 11, 15, 23, 31, 32}), who applied an integer-base GA using PNN classification rate as the fitness function. However, in their work, there is an absence of a mechanism that minimizes GA's randomness. As we already know from Figure 6 that a GA's single run may not be representative of its overall performance, it is therefore necessary to run GA several times. Furthermore, a 100% and 93% classification rate compares favourably with the results from the previous work (90.6%) using the above six-sensor subset ({8, 11, 15, 23, 31, 32}).

1.4 CONCLUSIONS

To conclude, in this chapter, we introduced a GA-based sensor selection technique for EN data called GNMM. GNMM uses MLPs as the classification engine. However, it separates the optimal sensor subset selection process from the final classification. By repeating the GA sensor selection process several times and applying a technique called *appearance percentage*, GNMM is able to minimize the randomness associated with GAs and MLPs. By utilizing GAs to identify optimal sensors that are being input into MLP classifiers, GNMM simplifies the classifier – MLPs – structure and makes the final classification process more efficient.

The effectiveness of GNMM is demonstrated by means of case study data, which has previously been explored by other authors using various methods. The number of sensors selected (i.e. 6) was deliberately made the same as those that have appeared in the literature. By comparing the results generated by GNMM to those presented in the literature, the effectiveness of this methodology is demonstrated.

References

- Ballabio, D., M. S. Cosio, et al. (2006). "A chemometric approach based on a novel similarity/diversity measure for the characterisation and selection of electronic nose sensors." *Analytica Chimica Acta* 578(2): 170-177.
- Bezdek, J. C. (2005). *Fuzzy models and algorithms for pattern recognition and image processing*. New York, Springer.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford, Oxford University Press.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, Springer.
- Boilot, P., E. L. Hines, et al. (2002). "Classification of bacteria responsible for ENT and eye infections using the cyranose system." *IEEE Sensors Journal* 2(3): 247-252.
- Davis, L. (1987). *Genetic algorithms and simulated annealing*. London; Los Altos, Calif., Pitman ; Morgan Kaufmann Publishers.
- De Castro, L. N. (2006). *Fundamentals of natural computing basic concepts, algorithms, and applications*. Boca Raton, Chapman & Hall/CRC.
- Dutta, R., E. L. Hines, et al. (2003). "Non-destructive egg freshness determination: An electronic nose based approach." *Measurement Science and Technology* 14(2): 190-198.
- Engelbrecht, A. P. (2007). *Computational intelligence : an introduction*. Chichester, England ; Hoboken, NJ, John Wiley & Sons.
- Gardner, J. W., P. Boilot, et al. (2005). "Enhancing electronic nose performance by sensor selection using a new integer-based genetic algorithm approach." *Sensors and Actuators B: Chemical* 106(1): 114-121.
- Gardner, J. W., H. W. Shin, et al. (2000). "An electronic nose system for monitoring the quality of potable water." *Sensors and Actuators B: Chemical* 69(3): 336-341.
- Gualdron, O., E. Llobet, et al. (2006). "Coupling fast variable selection methods to neural network-based classifiers: Application to multisensor systems." *Sensors and Actuators B: Chemical* 114(1): 522-529.
- Haupt, R. L. and S. E. Haupt (2004). *Practical genetic algorithms*. Hoboken, N.J., John Wiley.
- Haykin, S. S. (1999). *Neural networks : a comprehensive foundation*. Upper Saddle River, N.J., Prentice Hall.
- Hines, E. L., P. Boilot, et al. (2003). *Pattern analysis for electronic noses. Handbook of machine olfaction : electronic nose technology*. T. C. Pearce, S. S. Schiffman, H. T. Nagle and J. W. Gardner. Weinheim [Germany], Wiley-VCH: p.p. 133-160.
- Hines, E. L., E. Llobet, et al. (1999). "Neural network based electronic nose for apple ripeness determination." *Electronics Letters* 35(10): 821-823.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MA, USA, MIT Press Cambridge.
- Hoya, T. (2005). *Artificial mind system : Kernel memory approach*. Berlin New York, Spinger.
- Kecman, V. (2001). *Learning and soft computing : support vector machines, neural networks, and fuzzy logic models*. Cambridge, Mass., MIT Press.

- Kohonen, T. (2001). Self-organizing maps. Berlin ; New York, Springer.
- Lin, C. T. and C. S. G. Lee (1996). Neural fuzzy systems : a neuro-fuzzy synergism to intelligent systems. Upper Saddle River, NJ, Prentice Hall PTR.
- Llobet, E., O. Gualdron, et al. (2007). "Efficient feature selection for mass spectrometry based electronic nose applications." *Chemometrics and Intelligent Laboratory Systems* 85(2): 253-261.
- Llobet, E., E. L. Hines, et al. (1999). "Fuzzy ARTMAP based electronic nose data analysis." *Sensors and Actuators, B: Chemical* B61(1-3): 183-190.
- Mitchell, M. (1996). An introduction to genetic algorithms. Cambridge, Mass., MIT Press.
- Pearce, T. C., S. S. Schiffman, et al. (2003). Handbook of machine olfaction : electronic nose technology. Weinheim [Germany], Wiley-VCH.
- Phaisangittisagul, E. and H. T. Nagle (2007). "Enhancing multiple classifier system performance for machine olfaction using odor-type signatures." *Sensors and Actuators B: Chemical* 125(1): 246-253.
- Reeves, C. R. and J. E. Rowe (2003). Genetic algorithms : principles and perspectives : a guide to GA theory. Boston, Kluwer Academic Publishers.
- Rothlauf, F. (2006). Representations for genetic and evolutionary algorithms. Heidelberg, Springer.
- Schaffer, J. D., R. A. Caruana, et al. (1989). "A study of control parameters affecting online performance of genetic algorithms for function optimization." *Proceedings of the third international conference on Genetic algorithms*: 51-60.
- Scott, S. M., D. James, et al. (2006). "Data analysis for electronic nose systems." *Microchimica Acta* 156(3-4): 3-4.
- Specht, D. F. (1988). Probabilistic neural networks for classification, mapping, or associative memory, San Diego, CA, USA, Publ by IEEE, New York, NY, USA.
- Theodoridis, S. and K. Koutroumbas (2006). Pattern recognition. San Diego, CA, Academic Press.
- Ulivieri, N., C. Distant, et al. (2006). "IEEE1451.4: A way to standardize gas sensor." *Sensors and Actuators B: Chemical* 114(1): 141-151.
- Yang, J., E. L. Hines, et al. (2007). GNMM and accurate longitudinal dispersion coefficient prediction. the seventh UK CARE Annual Meeting, ISBN-13: 978-0-9551965-2-2. Greenwich, London.
- Yang, J., E. L. Hines, et al. (2007). Prediction of Longitudinal Dispersion Coefficient in Rivers using Variables Identified by Genetic Algorithms. The Fifth International Symposium on Environmental Hydraulics (ISEH V). Tempe, Arizona.
- Yang, J., E. L. Hines, et al. (2008). A Genetic Algorithm-Artificial Neural Network Method for the Prediction of Longitudinal Dispersion Coefficient in Rivers. *Advancing Artificial Intelligence through Biological Process Applications*. A. Porto, A. Pazos and W. Buño, Idea Group Inc.

Chapter 2

A Comprehensive Study of Sensor Responses to Fat Food Aroma – Using Genetic Algorithms and Neural Networks: The Particular Case of Virgin Olive Oil

Diego L. Garc ía-Gonz ález and Ramón Aparicio

2.1 INTRODUCTION

Systems based on sensor arrays or electronic noses (ENs) have proven to be rapid, objective and non-destructive instruments to analyse food aroma (Garc ía-Gonz ález and Aparicio, 2002a). Since most of the volatile compounds originate from the oxidation of lipids, the EN has significant potential for application in odour analysis of fatty products. Several sensor systems, most of them based on metal oxide sensors, have been applied to evaluate fatty food aroma. In the particular area of virgin olive oil aroma, some applications have been developed to classify samples using ‘quality categories’ (Garc ía-Gonz ález and Aparicio, 2003), variety (Cimato et al. 2006) or geographical area of origin (Pardo et al. 2000). However, such systems are not being extensively exploited in the food industry yet despite their potential as an on-line screening method and the profusion of literature in recent years reporting promising results. This delay in extensive application of such techniques is partially due to for example the difficulty in finding agreement between sensor responses and human odour perceptions; hence this results in a lack of understanding of the ‘information’ being provided via such sensors. A study of the relationship between both kinds of information - chemical, from the compounds, and physico-chemical, from sensor signals - requires further ‘information’ as to which volatiles are mainly

responsible for the overall sensor response as well as in the determination of their contribution to the aroma.

Any study intended to identify the relationship between odours and sensor responses should take into account the fact that odours are generally defined in qualitative and quantitative terms. In other words, from the chemical point of view the perception and intensity of an odour are understood to be determined by the presence and concentration of volatile compounds (Figure 1). On the other hand, from the standpoint of sensory perception, the aroma is characterized by odour intensity, odour threshold, and their corresponding descriptive attributes. In addition, the sensor responses do not only depend on the presence of compounds interacting with sensitive material, but also on many other parameters that are involved such as type of sensitive material, flow and type of carrier gas, and the kinetics of the adsorption/desorption processes. In order to define the relationships between these three aspects - aroma, EN data, and volatile compounds - a new methodology needs to be developed to express the agreement between sensor response, concentration of individual compounds, and sensory characteristics of each one of them.

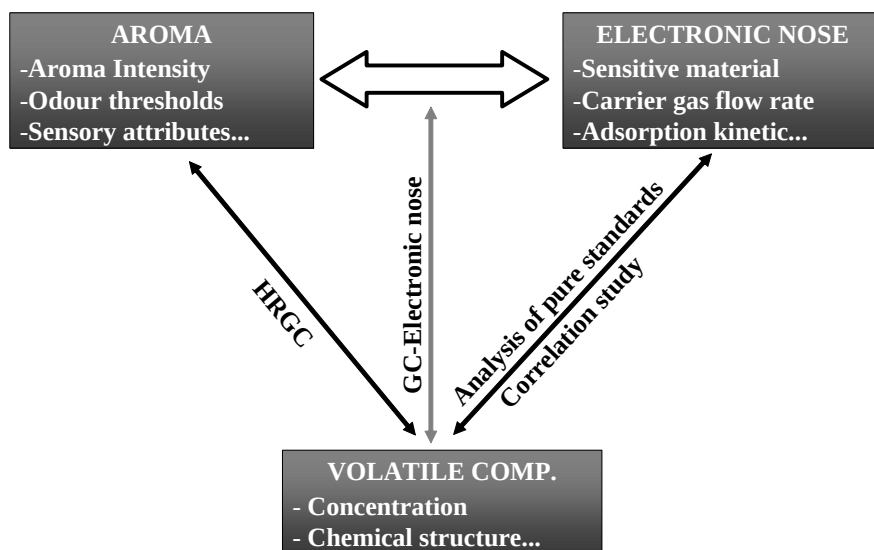


Figure 1: Schematic of the relationships between aroma and electronic nose data by means of the volatile composition of the sample.

The present chapter describes a novel approach whose objective is to quantify the individual contribution of volatile compounds to the overall metal oxide gas sensor responses when they are exposed to the sensitive material. The potential of this methodology for routine analysis of food aroma in contrast to conventional ENs is also discussed. In addition this chapter explores the possibility of using sensors as an alternative to classical chromatographic detectors. The use of a sensor array as a gas chromatographic (GC) detector system opens up the option wherein it can be used in tandem with other analytical techniques since sensors, unlike GC detector systems, are not sample destructive. Furthermore, a sensor array used as a detector allows us to obtain as much information as possible from the sample aroma and the content of the volatiles that have a major odour impact; once the appropriate sensors are selected. Such a methodology will provide more information than a GC chromatogram or single EN sensor responses. The peculiarities, problems, solutions and feasibility of this approach will be explored in this chapter in the framework of the particular case of virgin olive oil as a good example of fatty food aroma.

2.2 BACKGROUND

Some attempts to interpret the meaning of EN sensor data have been made through, for example, studies of the correlation between sensor signals and the concentrations of volatile compounds in the food quantified by GC (García-González and Aparicio, 2002b). An alternative to this method is the sequential analysis of the standards of the volatiles that are commonly present in the target food (García-González *et al.* 2007). However, food aroma is typically due to the presence of several hundred volatiles, which makes this approach difficult to implement. Furthermore, that procedure does not take into account the synergic and masking effects between odorants when interacting with sensor surface. A new approach based on the separation of the volatiles followed by their sequential exposure to sensors makes it possible to investigate the individual contribution of each volatile to the overall sensor response but now taking into account the actual concentration of the volatile in the sample and the possible interaction between volatiles. For this purpose, a silica column is coupled to a sensor array in order to collect a sequential series of sensor responses, each one

of them being the result of the interaction between a single compound, or a small group of compounds, and the sensor sensitive material.

The coupling of GC with an EN (GC-EN) has previously been used to correlate the intensity of sensor signals with the structure of volatile compounds (Kohl *et al.*, 2000; Masila *et al.* 1999), to analyse simple mixtures of volatiles (Williams and Pappas, 1999) or to remove a masking component (Ragazzo-Sánchez *et al.*, 2005). Other research groups are investigating the pros and cons of micromachined gas chromatographic columns in-tandem with sensor arrays (Zampolli *et al.*, 2005). Nevertheless, the coupling of gas chromatography-sensor array (GC-EN) has not been used yet to analyse food aroma composed of a complex mixture of volatile compounds such as virgin olive oil. In such foodstuffs, the variety of volatile compounds, together with the slow baseline recovery of most of sensor types, prevents a complete separation of compounds in terms of sensor response. Therefore, the hyphenated technique GC-EN could only be applied to complex food aroma after the experimental variables have been optimised and the choice of an appropriate data processing strategy to extract ‘information’; even when the peaks in the sensor signals are due to more than one compound delivered from the column, has been made. Furthermore, the interpretation of results provided by GC-EN is in need of a prior in-depth knowledge and experience of the volatile compounds responsible for food aroma.

2.3 CORRELATION STUDIES BETWEEN VOLATILE COMPOSITION AND SENSOR RESPONSES

The objective of this study is to compare the results obtained with a separative technique GC with those obtained by a non separative technique such as ENs. This comparison is carried out under the assumption that both techniques behave in a similar way when a compound is analysed at different concentrations. Also, it must be assumed that the volatile compounds interact with the sensitive material to the same degree either if they are isolated or in combination with others. These two assumptions may be questioned when this approach is applied to foods with a complex aroma where the concurrence of interactions of many volatile compounds with the sensors may cause synergistic and/or antagonistic processes. However, this procedure is appropriate in the case of volatiles that are present at sufficient concentration to produce high sensor responses. The authors carried out a study to interpret the sensor responses to ‘sensory defects’ of virgin olive oil (Garcia-Gonzalez and Aparicio, 2002b). Figure 2 shows the responses

of a metal oxide semi-conductor (MOS) sensor to virgin olive oil subjected to oxidation by UV irradiation, the concentration of hexanal in the sample is quantified by GC, and the intensity of the rancid attribute according to the official sensory assessment (EC, 2003). The evolution of the three types of ‘information’ over time is quite similar, which proves that a relationship may be found by a correlation study despite the fact that many other volatiles also interact with the sensors. The correlation study carried out in this work suggests that the sensor response is correlated with the concentration of hexanal and the scores for the rancid attribute produced regression coefficients higher than 0.85%. Similar results were found when the correlation study was carried out with other sensory defects (such as fusty, winey-vinegary) and other volatile compounds (e.g. acetic acid) (García-González and Aparicio, 2000c).

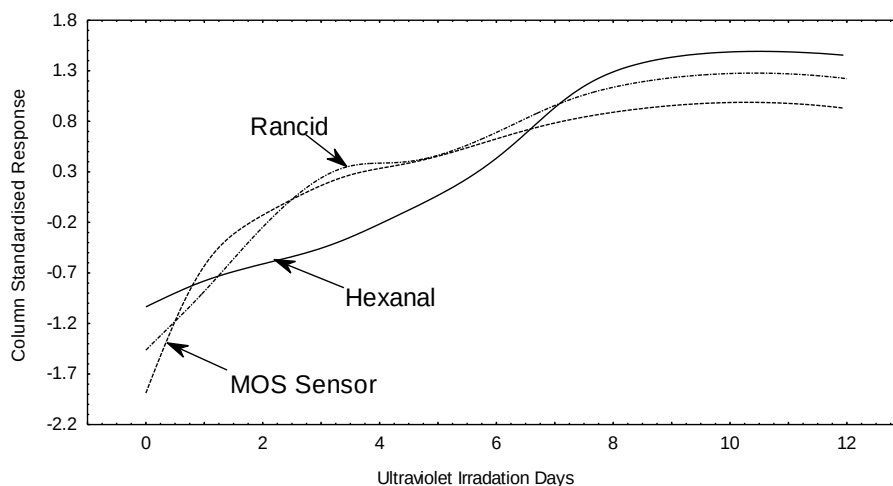


Figure 2: Results of an oxidation process of the samples irradiated with an ultraviolet beam for 12 days: rancid sensory defect evaluated by panellists, hexanal quantified by DHS-GC, and the maximum values of the first windowed time slicing of sensors 1 and 6. Results fitted by the least-square algorithm (García-González and Aparicio, 2002).

However, the aroma of food samples are composed of many volatile compounds that are present at a wide range of concentration. Table 1 shows the volatile compounds identified in virgin olive oil. Based on this information the indication is that it is difficult to reach credible conclusions from the results if the samples selected for a correlation study do not include all the quality levels and all the possible concentration values for all the compounds. Thus, Table 2 shows the results of a correlation study between the responses of two types of MOS sensors (Cr_3O_2 doped with Ti, and SnO_2 doped with Pd and Pt) and the

concentration data of volatiles quantified in 14 virgin olive oils of different categories of quality and six oil standards with the following defects; bitter fusty, winey-vinegar, muddy-sediment and rancid.

Compounds that were correlated with the sensor responses at a regression coefficient $R > 0.80$ (R1 in Table 2) were hexanal, 2-heptanone, ethanol, 2-methyl-3-buten-2-ol, 6-methyl-5-hepten-2-one, octanal, and acetic acid. However, when the correlation study was carried out considering only those oils of good quality (extra-virgin and virgin categories), other volatile compounds also showed regression coefficients higher than 0.80 (R2 in Table 2) in the case of Cr_3O_2 sensor, such as (E)-2-hexenal and (Z)-3-hexenyl acetate, characterized by bitter almond and fruity odour respectively. On the other hand, in this second correlation study, only the responses of SnO_2 sensors were highly correlated with the concentration of 1-octanol. This fact indicates that these sensors only show high responses with volatiles that are present at high concentration in olive oils with low quality, while the Cr_3O_2 sensors employed in this study are sensitive to those volatile compounds characteristics of high quality olive oils.

The volatile compounds that are correlated with the sensor responses may not be necessarily significant from a sensory point of view since only those volatile compounds with an odour threshold lower than their actual concentrations may be perceived by the human nose. Therefore, in order to provide an appropriate sensory interpretation of the sensor responses, data for concentration and odour thresholds expressed as odour activity values (OAV) should be considered; OAV being the ratio between the odour threshold and the concentration in the sample and it is higher than 1 for those compounds that are actual contributors to the food aroma. Table 3 shows the range of concentration for some volatile compounds quantified in the extra virgin olive oil (high quality) and lampante olive oil (low quality), and the odour thresholds for each one of them. Thus, important volatiles in lampante oils in terms of their contribution to sensor responses, and their sensory perception, are acetic acid, (Z)-3-hexenyl acetate, ethanol, hexanal, (E)-2-hexenal, 6-methyl-5-hepten-2-one and ethyl propanoate. On the other hand, concerning virgin olive oils with good quality, fewer compounds can be highlighted. However, it is well known that the sensors show a response when they are exposed to the aroma of virgin olive oil, and this fact points out that a correlation study is not enough to explain the sensor responses.

Table 1: Some of the volatile compounds identified in different categories of virgin olive oil by several researchers. Note: numbers in brackets correspond to compounds in Figure 3.

Aldehydes	Alcohols	Esters
Acetaldehyde	Methanol	Methyl acetate (2)
2-Methylbutanal	Ethanol	Ethyl acetate (3)
3-Methylbutanal	2-Methyl-1-butanol	Butyl acetate
2-Methyl-2-butenal	3-Methyl-1-butanol (10)	2-Methylbutyl acetate
Pentanal	2-Methyl-3-butenol	Isopentyl acetate
(E)-2-Pentenal	2-Methyl-1-propanol	Hexyl acetate (12)
(Z)-2-Pentenal	1-Pentanol	2-Hexenyl acetate
Hexanal (7)	3-Pentanol	3-Hexenyl acetate
2-Hexenal	1-Penten-3-ol	(Z)-3-Hexenyl acetate (13)
(E)-2-Hexenal (11)	(Z)-2-Pentenol	Octyl acetate
(Z)-2-Hexenal	3-Hexen-1-ol	2-Ethylphenyl acetate
3-Hexenal	(E)-3-Hexen-1-ol (15)	Benzyl acetate
(Z)-3-Hexenal	(Z)-3-Hexen-1-ol (16)	Phenethyl acetate
2,4-Hexadienal	2-Hexen-1-ol	Ethyl propanoate (4)
Heptanal	(E)-2-Hexen-1-ol (17)	Propyl propanoate
(E)-2-Heptenal	(Z)-2-Hexen-1-ol	Ethyl 2-methylpropanoate
(Z)-2-Heptenal	4-Hexen-1-ol	Propyl 2-methylpropanoate
2,4-Heptadienal	1-Hexanol (14)	Methyl butanoate
Octanal	1-Heptanol	Ethyl butanoate
(E)-2-Octenal	1-Octanol	Methyl 2-methylbutanoate
Nonanal	1-Octen-3-ol (19)	Ethyl 2-methylbutanoate
(E)-2-Nonenal	2-Octen-1-ol	Methyl 3-methylbutanoate
2,4-Nonadienal	1-Nonanol	Ethyl 3-methylbutanoate
(E)-2-Decenal	1-Decanol	Butyl 3-methylbutanoate
Decanal (20)	Lavandulol	Methyl pentanoate
2,4-Decadienal	Linalool	Methyl hexanoate
(E)-2-Undecenal	Benzyl alcohol	Ethyl hexanoate
	2-Phenylethanol	Methyl heptanoate
	A-Terpineol	Methyl octanoate
	1,3-Butanediol	3-Methyl-2-butyl acetate (9)
Ketones	Hydrocarbons	Phenols
2-Butanone	2-Methylbutane	Anisole
2-Propanone (1)	2-Methylpentane	
3-Methyl-2-butanone	3-Methylpentane	Ethers
3-Pentanone (5)	Hexane	Diethyl ether
4-Methyl-2-pentanone	Hexene	1,8-Cineole
1-Penten-3-one		
2-Hexanone		
2-Heptanone		

Table 1: Some of the volatile compounds identified in different categories of virgin olive oil by several researchers. Note: numbers in brackets correspond to compounds in Figure 3. (Continued)

Ketones (Cont.)	Hydrocarbons (Cont.)	Acids
6-Methyl-5-hepten-2-one	Heptane	Acetic acid (18)
2-Octanone	Octane	Butanoic acid
3-Octanone	1-Octene	Pentanoic acid
2-Nonanone	Nonane	Propanoic acid
Acetophenone	Tridecene	Hexanoic acid
	Pentene dimers	Heptanoic acid
Sulfur compounds	Methylbenzene (6)	
3-Isopropenylthiophene	Styrene	Furans
2,5-Diethylthiophene	Ethyl benzene (8)	Ethylfuran
2-Ethyl-5-hexylthiophene	Limonene	2-Propylfuran
		3-Propylfuran
		3-Methyl-2-pentylfuran
		2-Propyldihydrofuran
		3,4-Methyl-3-pentenyl furan

Table 2: Regression coefficients (R) between the concentration values of volatile compounds quantified in olive oil by SPME-GC and sensor responses. Note: R1, Regression coefficients obtained with virgin olive oils of all categories; R2, Regression coefficients obtained when only high quality oils were considered.

Volatile compound	Cr ₃ O ₂ + Ti Sensors		SnO ₂ +Pb+Pt Sensors	
	R1	R2	R1	R2
Hexanal	0.80	0.90	0.48	0.67
2-Heptanone	0.85	0.86	0.68	0.67
Ethanol	0.94	0.56	0.87	0.64
2-Methyl-3-buten-2-ol	0.92	0.83	0.85	0.44
6-Methyl-5-hepten-2-one	0.91	0.89	0.84	0.44
Octanal	0.87	0.84	0.81	0.68
1-Octanol	0.21	0.42	0.48	0.83
Acetic Acid	0.81	0.53	0.82	0.46
(E)-2-Hexenal	0.49	0.89	0.49	0.69
Octane	0.62	0.87	0.39	0.30
Ethyl Propanoate	0.20	0.83	0.24	0.57
Pentanal	0.69	0.85	0.57	0.46
1-Penten-3-ol	0.37	0.85	0.42	0.68
(Z)-3-Hexenyl acetate	0.57	0.84	0.62	0.62
1-Penten-3-one	0.14	0.82	0.27	0.76
2-Methyl-3-Buten-2-ol	0.92	0.81	0.87	0.46
(E)-3-Hexen-1-ol	0.32	0.80	0.34	0.42

Table 3: Concentration range (mg/kg), sensory characteristics, and odour threshold (mg/kg) of some of the compounds quantified in virgin olive oil (VOO).

Compound	Concentration in non-lampante VOO vs. lampante VOO ^a	Sensory Characteristic	Threshold
Acetic acid	(tr ^b) vs. (0.01-128.35)	Bitter-pungent odour	0.50
(Z)-3-Hexenyl acetate	(1.10-2.10) vs. (1.7-3.10)	Green or fruity banana	0.70
Ethanol	(2.25-6.59) vs. (3.6-57.91)	Alcohol	30.00
2-Heptanone	(0.01-0.20) vs. (0.13-0.60)	Fruity	0.40
Hexanal	(0.56-3.43) vs. (2.48-13.95)	Green, fat	0.08
(E)-2-Hexenal	(0.07-1.04) vs. (0.29-8.64)	Bitter almond	0.42
2-Methyl-3-buten-2-ol	(tr-0.03) vs. (0.04-0.29)	Herbaceous, land, oily	0.50
6-Methyl-5-hepten-2-one	(0.03-0.08) vs. (0.15-3.15)	Pungent, mushroom, gum	1.00
Octanal	(0.11-0.13) vs. (0.15-0.20)	Fat	0.32
Octane	(0.01-0.03) vs. (0.26-3.85)	Sweet, alkane	0.94
Pentanal	(0.039-0.47) vs. (0.48-1.64)	Wood, fat	0.24
Ethyl propanoate	(0.01-0.04) vs. (0.15-6.66)	Ether, intense fruity	0.10

NOTE: ^a, it includes standard of oils with defects; ^b, trace.

The results shown in Table 2 prove that the correlation study between the sensor responses and the volatile composition is greatly affected by the selection of samples. Therefore, in the case of foods with complex aroma it is difficult to reach a generalized theory about which volatile compounds are responsible for a sensor response since it depends on the kind of volatiles and their concentration in each sample analysed for the correlation study. However, this approach is valid for foods with simpler odours, or to estimate the relative significance of some volatiles (e.g. hexanal or ethanol in olive oil) contributing to the responses. In addition to this limitation, a correlation study does not consider the masking and synergy effects between compounds interacting with the sensors. Further to these drawbacks, the high regression coefficients of some compounds may not be due to their relationships with sensor responses but due to high correlation with the concentration of other compounds that do interact with the sensors (e.g. correlation between acids and esters in olive oil). As a consequence, any study to elucidate the volatile compounds contributing to sensor response should be supported by experimental approaches other than mathematical relationships.

Another mathematical approach to relate the information from volatile compounds and sensor responses is the projection of both types of information onto a PCA plot. Figure 3 shows the PCA plot of the sensor responses of 12 sensors (coded with numbers 1-12), 6 of them composed of Cr₂O₃ (doped with Ti) and the other 6 composed of SnO₂ (doped with Pd+Pt). This PCA

corresponded to 14 samples of virgin olive oil from the categories lampante (low quality) and non-lampante (extra-virgin and virgin oils). Only the compounds that showed differences in their concentration between lampante and non-lampante oils are displayed in order to facilitate interpretation; codes in brackets correspond to the compounds shown in Table 1.

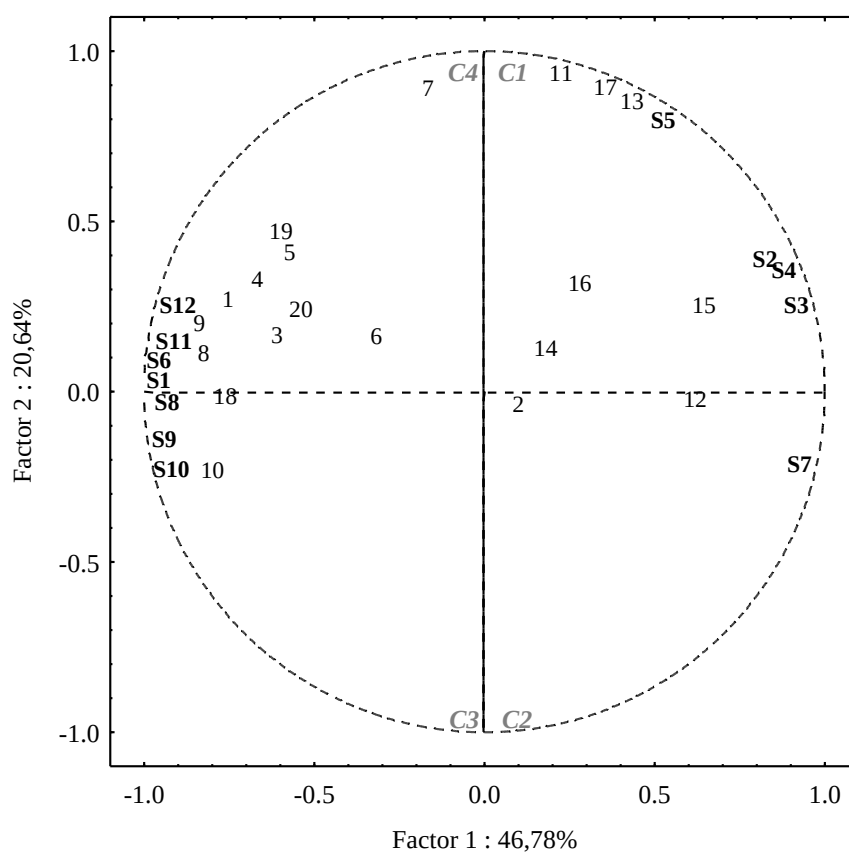


Figure 3: Projection of the information on volatile compounds onto a PCA of sensor responses.
Note: Codes correspond to volatile compounds shown in Table 1.

The first conclusion that can be inferred from Figure 3 is that there are two types of sensors: sensors 2-5 and, in another group, sensors 1, and 6-12. The responses of the first set of sensor are highly correlated with the concentration of compounds responsible for green attributes such as hexyl acetate (12), (E)-3-hexen-1-ol (15), 1-hexanol (14), (Z)-3-hexen-1-ol (16), (E)-2-hexenal (11), (E)-

2-hexen-1-ol (17), (Z)-3-hexenyl acetate (13) (Aparicio and Morales 1998). The second set of sensors (S1, S6-S12), on the other hand, is located in the same quadrants as volatile compounds that are typically quantified at high concentrations in low quality oils (lampante), such as ethyl benzene (8), 2-methyl-butyl acetate (9), acetic acid (18), 3-methyl-1-butanol (10), 2-propanone (1), ethyl acetate (3), ethyl propanoate (4), decanal (20), 1-octen-3-ol (19), 3-pentanone (5), methylbenzene (6) (Morales *et al.* 2005).

This result shows that these sensors will be likely to be able to differentiate high quality oils from the low quality ones but at the same time it suggests that there is ‘overlapping information’ that must be removed before it would be appropriate to try to develop ‘mathematical model’ to try to classify the samples. The results from PCA projection, however, should be cautiously considered as they are the results of ‘mathematical correlations’ rather than experimental data. Thus, as has been said, a high correlation does not necessarily mean a high sensitivity to a volatile compound. Therefore, these results should be further verified by somehow exposing the sensors to individual compounds.

2.4 INTERPRETATION OF THE SENSOR RESPONSES BY MEANS OF THE ANALYSES OF STANDARDS OF VOLATILE COMPOUNDS

An experimental alternative to determine the most important volatile compounds interacting with sensors and producing high responses is the systematic analysis of pure compounds diluted in an odourless vegetable oil (refined). Even though the headspace of virgin olive oil is composed of more than 100 hundred volatile compounds, a group of them can be selected based on previous experience and this subset is considered to be sufficient for this study. Thus, one strategy entails using dilutions of a set of standards at the same concentration. This experimental design makes it possible to compare the relative importance of volatile compounds contributing to the whole sensor response. Tables 4 and 5 show the relative response of a sensor array to a series of standards in refined oil at 5mg/kg. The normalization of the responses has been performed by assigning a value 1 to the volatile compound that produces the maximum response within each sensor. The results reveal that the compounds that induce higher sensor responses are ethanol, (Z)-2-penten-1-ol, 2-methyl-1-propanol, (E)-2-hexenal, hexanal, acetic acid, 3-methyl-1-butanol, 3-methyl-butanol, 3-pentanone, 2-pentanone, 1-pentanol.

Table 4: Standardized responses of Cr₃O₂ sensors –doped with Ti- (S1-S12) to 5 mg/kg of volatile compound standards in refined oil. The responses were standardized by the maximum value obtained in each sensor.

Volatile Compound	S1	S2	S3	S4	S5	S6
Octane	0.54	0.24	0.24	0.22	0.20	0.19
Ethyl acetate	0.68	0.14	0.18	0.19	0.14	0.10
3-Methylbutanal	0.91	0.84	0.85	0.88	0.69	0.79
Ethanol	0.80	0.47	0.59	0.67	0.57	0.48
Ethyl propanoate	0.73	0.17	0.25	0.25	0.19	0.14
3-Pentanone	0.98	0.63	0.96	0.89	0.69	0.55
Hexanal	0.95	0.52	0.80	0.73	0.54	0.37
2-Methyl-1-propanol	0.86	0.94	0.81	0.95	0.93	1.00
Ethylbenzene	0.55	0.24	0.23	0.23	0.20	0.23
3-Methyl-1-butanol	0.75	0.82	0.71	0.73	0.75	0.68
(E)-2-Hexenal	1.00	0.70	0.93	0.94	0.54	0.32
1-Pentanol	0.92	0.64	0.77	0.83	0.57	0.50
Hexyl acetate	0.66	0.14	0.23	0.20	0.16	0.10
(Z)-3-Hexenyl acetate	0.53	0.31	0.24	0.26	0.25	0.23
(Z)-2-Penten-1-ol	0.90	1.00	1.00	1.00	1.00	0.73
6-Methyl-5-hepten-2-one	0.94	0.29	0.43	0.41	0.28	0.18
1-Hexanol	0.86	0.40	0.57	0.58	0.40	0.31
(Z)-3-Hexen-1-ol	0.93	0.52	0.68	0.65	0.44	0.33
Acetic acid	0.82	0.41	0.49	0.52	0.51	0.46
1-Octen-3-ol	0.94	0.34	0.46	0.45	0.32	0.24
Decanal	0.78	0.45	0.47	0.45	0.34	0.32

NOTE: Values higher than 0.80 have been written in bold Italics.

Regarding the type of sensors, the Cr₃O₂ sensors are characterized by a high response to (Z)-2-pentan-1-ol, a medium response to ethanol, 6-methyl-5-hepten-2-one, (Z)-3-hexen-1-ol, among others, and a low or insignificant response to esters (ethyl acetate, isobutyl acetate, hexyl acetate). On the other hand, ethanol is by far the compound that produces highest responses in SnO₂ sensors. Unlike the Cr₃O₂ sensors, the SnO₂ sensors show a medium response to the esters included in this study, and their relative response to (E)-hexenal is slightly higher as well.

Table 5: Standardized responses of SnO₂ sensors – doped with Pb and Pt- (S7-S12) to 5 mg/kg of volatile compound standards in refined oil. The responses were standardized by the maximum value obtained in each sensor.

Volatile Compounds	S7	S8	S9	S10	S11	S12
Octane	0.52	0.46	0.66	0.44	0.16	0.09
Ethyl acetate	0.79	0.50	0.73	0.45	0.14	0.20
3-Methylbutanal	0.80	0.86	0.92	0.94	0.19	0.19
Ethanol	1.00	1.00	1.00	1.00	1.00	1.00
Ethyl propanoate	0.77	0.52	0.77	0.77	0.08	0.13
3-Pentanone	0.85	0.87	0.89	0.90	0.25	0.24
Hexanal	0.78	0.76	0.80	0.83	0.19	0.18
2-Methyl-1-propanol	0.78	0.88	0.95	0.90	0.01	0.04
Ethylbenzene	0.66	0.67	0.73	0.52	0.20	0.14
3-Methyl-1-butanol	0.76	0.74	0.89	0.83	0.11	0.07
(E)-2-Hexenal	0.79	0.76	0.79	0.89	0.11	0.12
1-Pentanol	0.80	0.77	0.87	0.85	0.12	0.15
Hexyl acetate	0.54	0.48	0.61	0.41	0.03	0.03
(Z)-3-Hexenyl acetate	0.67	0.79	0.76	0.58	0.06	0.01
(Z)-2-Penten-1-ol	0.80	0.77	0.81	0.80	0.16	0.10
6-Methyl-5-hepten-2-one	0.61	0.55	0.69	0.58	0.06	0.07
1-Hexanol	0.67	0.64	0.76	0.66	0.05	0.07
(Z)-3-Hexen-1-ol	0.66	0.70	0.70	0.61	0.08	0.08
Acetic acid	0.84	0.80	0.80	0.75	0.25	0.26
1-Octen-3-ol	0.77	0.76	0.77	0.74	0.06	0.02
Decanal	0.68	0.70	0.76	0.60	0.19	0.17

NOTE: Values higher than 0.80 have been written in bold Italics.

On the whole, the sensors were sensitive to alcohols, aldehydes and ketones, whereas they show low sensitivity to esters and hydrocarbons. Nevertheless, in this study the concentration of the compounds has been set at a fixed value (5 mg/kg) for an easier interpretation of results while the actual concentration values of these compounds in real samples cover a wide range that makes a complete study very difficult to address it.

Figure 4 shows the response of a sensor to different concentration of 6 compounds diluted in refined oil (0.2, 0.3, 0.4, 1.0, 1.5, 2.5, 3.5, 5.0 mg/kg). From the figure it may be concluded that the highest responses corresponded to

(E)-hexenal, ethanol and hexanal, while acetic acid, hexanol and 6-methyl-5-heptene-2-one induce medium responses; and finally the esters showed the lowest intensity. However, this distinction is hard to make at low concentrations at which some of the esters (e.g. hexyl acetate) may produce responses higher than other compounds. Thus, a study of responses of standards based on a single concentration, although providing valuable information and being easy to interpret, may lead to incorrect conclusions that cannot be extrapolated to real samples where the compounds are present at different concentrations; and where they interact with each other. However, this approach is useful since it is indicative of the relative importance of some volatiles in the sensors response, such as the acetic acid in oils with winey-vinegary defect (García-González and Aparicio, 2002c) or hexanal in rancid oil (García-González and Aparicio, 2002b). The analysis of standards is also useful to help classify a set of sensors into different groups and in predicting their behaviour when they are exposed to real samples or to explain retrospectively results already obtained (García-González *et al.* 2007).

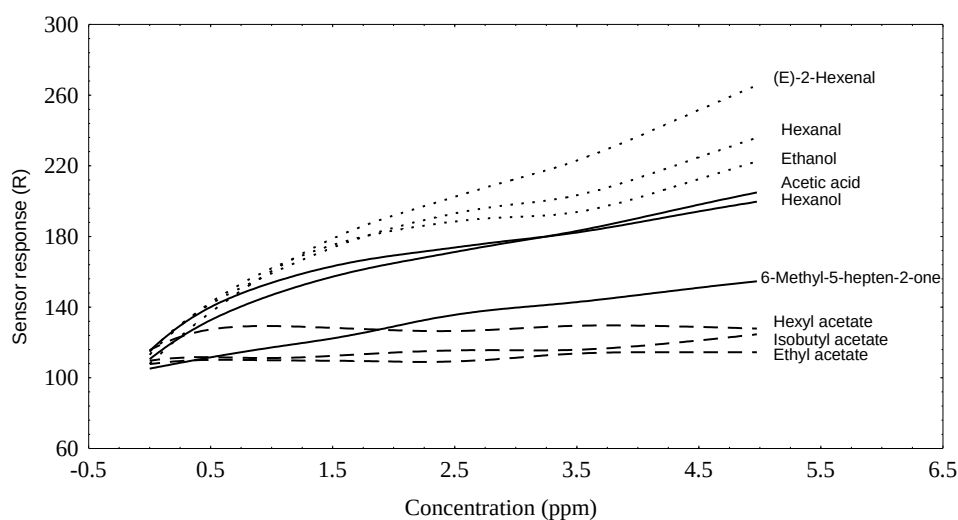


Figure 4: Responses of a Cr_2O_3 sensor (doped with Ti) to different concentration of volatile compounds that produce high (dotted lines), medium (continuous line) and low responses (dashed line). Note: Maximum response, raw data; compounds diluted in refined oil.

2.5 COUPLING GC AND ELECTRONIC NOSE AS AN EXPERIMENTAL APPROACH TO CHEMICAL EXPLANATION OF SENSOR SIGNALS.

The sensor systems or electronic noses (EN) have shown to be fine instruments for determining virgin olive oil aroma. An EN is a device composed of an array of gas sensors that have non-selective responses to volatiles but makes it possible to build a pattern recognition algorithm from the multivariate data analysis of the whole set of sensor signals. The response of MOS sensors are commonly characterised by a curve with adsorption and desorption slopes that correspond to the deposition and subsequent combustion of the volatile compounds on the hot MOS film of the sensor. This characteristic signal does not explain to which volatiles the sensor is sensitive since the response is the result of the concomitant adsorption processes of all the volatiles that occur as soon as they reach the sensor chamber. Although mathematical algorithms such as windowed time slicing (WTS) (Garcia-Gonzalez and Aparicio, 2002c), allow us to extract more information from the raw response, obtaining better classification rates, they still do not provide any kind of ‘chemical knowledge’ of the sensor signals that may be directly related with sensory attributes. Thus, a quality classification of virgin olive oils by means of sensors has as a main drawback the fact that it is impossible to determine whether the classification is based on volatiles that actually contribute to virgin olive oil aroma or not.

As has been described in a previous section, the study of the ‘correlation’ between the concentration of volatile compounds and the responses of the sensors provide certain useful information to determine the sensitivity of the sensors to some volatiles. But the relationship between sensors and volatiles observed using this method do not have any experimental basis and they are thus merely ‘mathematical correspondences’. On the other hand, an experimental verification of this relationship can be realised by the coupling of a gas chromatograph that would separate the volatile compounds of the samples prior to being analysed by the sensors. This experimental methodology would ensure that the compounds are transferred to the sensors at the concentrations that they are present in the samples.

2.5.1 Set up of a coupled system GC-EN: Experimental considerations

Figure 5 shows the schematic of the coupling of a chromatograph and a sensor system. In the resulting instrumented system the sensor array is connected to a flame ionization detector (FID) at the end of the chromatographic column. Thus, this column separates the compounds by their polarity (according to their Kovats Index) and they are sequentially released into a sensor chamber where they interact with the sensors. As a consequence, the sensors individually respond to the volatiles one at a time and with this strategy they provide information about the relative sensitivity for each compound.

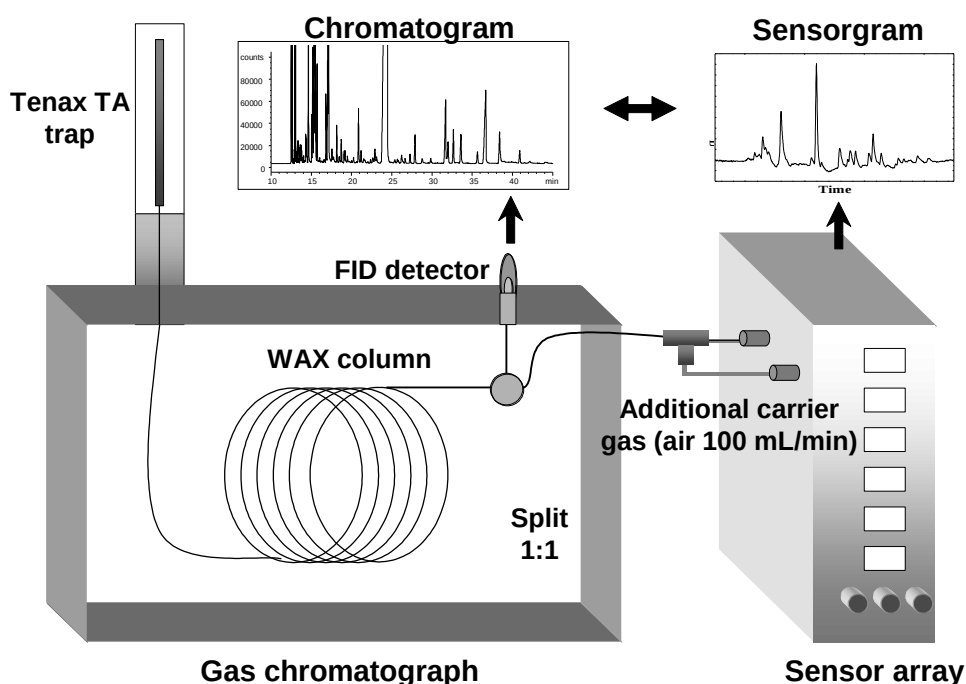


Figure 5: Coupling a gas chromatograph to a sensor array of MOS sensors.

The sensor response recorded during the experiment (Figure 6) is the results of the deposition of volatiles on the sensor grid which means a variation of the sensor resistance. Thus, the sensor response, concerning both the height and the breadth of the peak is related to the sensor sensitivity to the volatile and the

volatile concentration. Both variables, height and breadth expressed in the time domain, may be related to the peaks of the chromatogram obtained from the FID output.

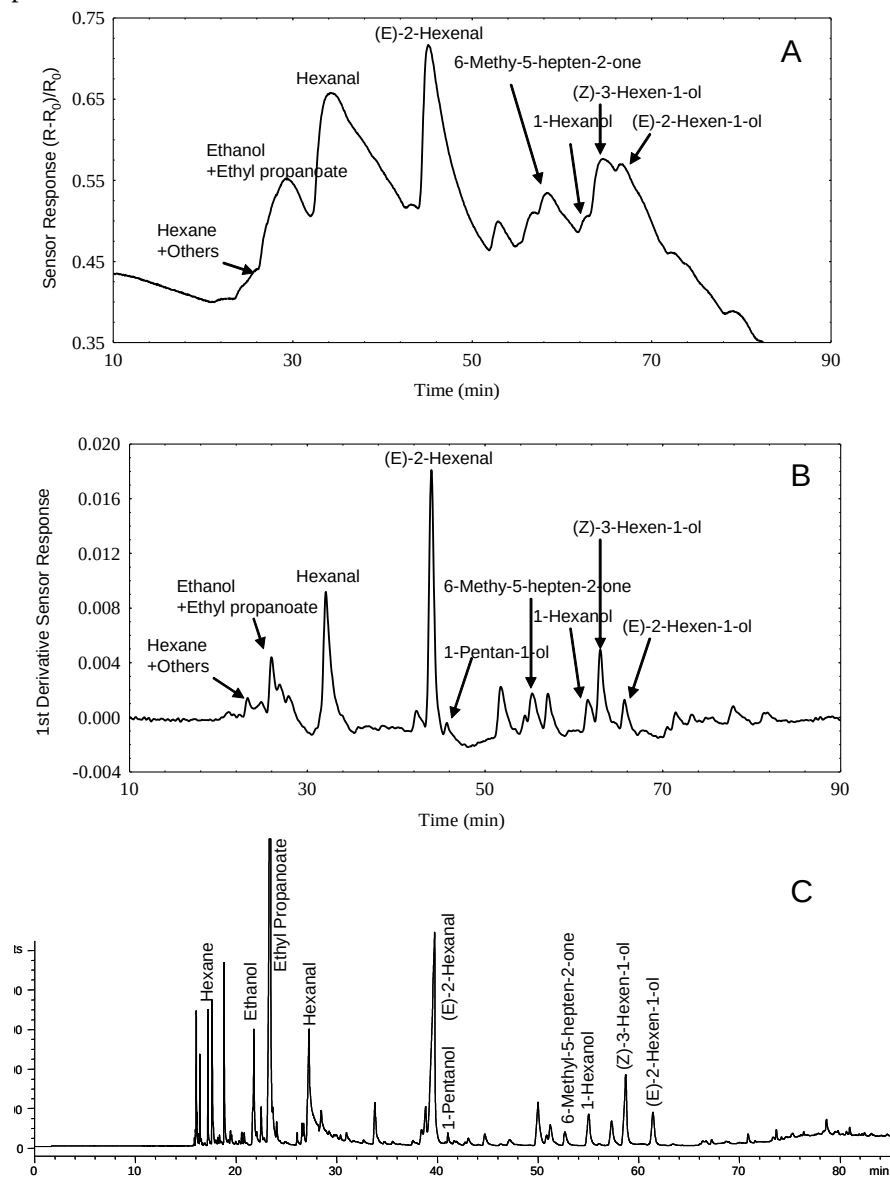


Figure 6: Sensorgram of a Cr3O2 sensor (A), its first derivative (B) and the chromatogram of volatiles (C) of an extra virgin olive oil.

Due to the similitude between the sensor response in the GC-EN and the chromatogram the former will be henceforth referred to as a sensorgram. As the sensor response is slower than the output of the GC ionization detector, the response consists of a series of adsorption and desorption slopes, some of them caused by the combined action of several volatile compounds that are released by the GC column in a short time frame. Furthermore, the sensor's relatively slow response does not allow a complete recovery of the baseline in the experiments till the end of the process. In order to individualize the contribution of each volatile - a peak can be the result of the contribution of several volatiles, as has been mentioned - the first derivative is calculated for the whole sensorgram after applying the smooth Savitsky-Golay algorithm. Thus, the resulting sensorgram is much easier to compare with its corresponding chromatogram to assign most of the peaks to the volatile compounds that are being released by the column, as shown in Figure 5. This figure shows a delay of several minutes in the response of the sensor in comparison with the GC detector as a consequence of a rapid response to the absorption of the volatiles at the sensor surface but a lengthy desorption of them. This sensor behaviour makes the sensor response to a virgin olive oil sample much more complicated to explain. Thus, Figure 6 shows the response of two sensors (Cr_3O_2 doped with Ti and SnO_2 doped with Pd and Pt) to three chemical compounds - 1-hexanol, (Z)-3-Hexen-1-ol and (E)-3-Hexen-1-ol - that are present in a virgin olive oil and are sequentially released by the GC column. The first two sensorgrams (A) correspond to the raw data from the sensors, and they show that there is apparently a broad peak per sensor response though the response of sensor S1 shows a shoulder in the peak left slope. The first derivative of the raw responses (B) reveals the information embedded in the broad peak, such as the responses to (Z)-3-hexen-1-ol and (E)-3-hexen-1-ol which are preceded by the response to 1-hexanol. According to the relative height of the signals for these compounds it can be inferred that the sensors show poor sensitivity to 1-hexanol, while the compounds responsible for this broad band are mostly (E)-3-hexen-1-ol and (Z)-3-hexen-1-ol. Beside the difference in the response intensity of the two sensors considered in Figure 7 (0.42 vs. 0.19 relative units), the speed of response also differs between these two sensors, Cr_3O_2 sensor being the faster showing changes in the signal intensity and therefore it is the most suitable for GC-EN analysis. The difference in the relative intensity of these three compounds revealed in this experiment partially agrees with those obtained with pure standards diluted in refined oil (Tables 4 and 5) in which the relative sensitivity of most of SnO_2 sensors for (Z)-3-hexen-1-ol is

slightly higher than that for 1-hexanol (e.g. relative response for Sensor 1 is 0.86 and 0.93 for 1-hexanol and (Z)-3-hexen-1ol respectively). However, this difference is more evident when it is observed in a sensorgram, where the sensor never reaches the baseline and the response for each compound is greatly affected by the rest of the volatiles. On the other hand, in a study carried out with pure compounds diluted in refined oil, each volatile is separately exposed to the sensor when it is in a basal or non-excited state so the sensors responses are more intense than in true samples where all the volatiles reach the sensor at the same time.

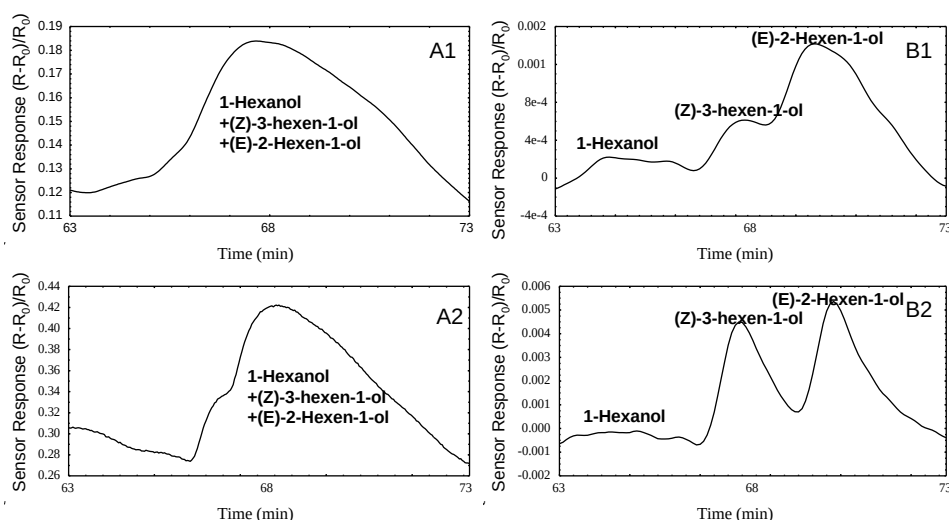


Figure 7: Sensorgrams (A) and their corresponding first derivative (B) of two sensor composed of SnO_2 doped with Pd and Pt and Cr_2O_3 doped with Ti (coded with number 1 and 2 respectively).

Alongside the procedure to interpret the signal derived from a GC-EN system, another major issue is the assurance of the repeatability in both the intensity of the peaks and the time (retention time) at which they are shown up in the sensorgram. As in the individual cases of GC and EN systems, the repeatability of the coupled instrument greatly depends, among other factors, on the performance of gas connections, valves, flow rate and the lengths of the connecting tubes between sample and detector. The optimization of all these variables involved in the coupling of both instruments should lead to a repeatability, expressed in relative standard deviation (RSD%), lower than 10%. A study of repeatability involving the repeated analysis of a sample of virgin

olive oil resulted in an average RSD% of 12%. Those peaks with RSD% higher than 10% were left out in our analysis later.

Even though the derivative of the responses permits us to assign each peak in the sensorgram to a volatile or group of them, the complex aroma of foods such as olive oil requires an additional identification by analysing simple mixtures of pure compounds. These analyses also inform of the delay of the sensor responses related to the FID detector for each volatile as a consequence of the segment of column that covers the distance between the split in the GC oven and the sensor chamber. This last segment of the column is connected to an additional inlet of carrier gas (air) to condition the GC carrier gas in temperature, flow rate and relative humidity. Thus, a column length of 15cm between the GC and the sensor system provides a delay in the sensor response of 5-10 min. This delay may be considered to be a characteristic initial time for the sensorgram.

Finally, the control of the aforementioned delay in the sensor response involves the addition of a standard to the sample that must be chosen for both its high sensor response and low interference with other compounds. 4-methyl-2-pentanol, an alcohol that is not naturally present in olive, fulfils these requirements and therefore it was selected as internal standard.

2.5.2 Analysis of virgin olive oil aroma by GC-EN: Interpretation of results

The typical sensorgram of a Cr₂O₃ sensor corresponding to the aroma of an extra virgin olive oil was already illustrated in Figure 6. The first changes in the response intensity are observed at 20 min (retention time, RT) corresponding to hexane, octane and methyl acetate. They are compounds that scarcely contribute to virgin olive oil aroma (Aparicio and Alonso, 1994) and their low responses in the sensorgram indicate that they also only make a small contribution to the sensor signal. The second broad peak, more intense than the previous one, is due to a group of compounds composed of ethyl acetate, ethanol, ethyl propanoate and 3-pentanone, all of them with a very similar retention time. Considering the results of the study carried out with pure compounds, it may be assumed that most of this response is due to ethanol rather than the other compounds (Tables 4 and 5). However, the odour threshold of this compound is very high in relation to its actual concentration in oils (Table 3). In contrast, hexanal produces a very intense peak, being one of the compounds that contribute most to the overall sensor response. Its concentration is far higher than its threshold, which proves that it has a remarkable sensory significance in olive oil aroma perception. The

hexanal odour is described as green (Aparicio and Alonso, 1994), but at higher concentration, such as in lampante virgin olive oils, it may be responsible for fatty or rancid odours.

The next compounds that are observed in the sensorgram are (E)-2-hexenal and 1-pentanol. The first is characterized by a bitter almond odour and its concentration is high in all categories of virgin olive oils. The second compound, 1-pentanol, is characterized by an intense and unpleasant glue-like aroma and it only contributes in lampante virgin olive oil aroma (Morales *et al.* 2005).

The changes of response intensity in the following time frame (RT=53-70min) is attributed to a series of compounds that are commonly responsible for pleasant fruity-green attributes (Aparicio and Alonso, 1994), and they have similar retention times in the chromatogram. These compounds are hexyl acetate, 6-methyl-5-hepten-2-one and (Z)-3-hexenyl acetate, and the alcohols 1-hexanol, (Z)-3-hexen-1-ol and (E)-2-hexen-1-ol. Depending on the relative concentration of these compounds, the response of some compounds may mask others, hence the derivative signal is necessary to elucidate the contribution of each one of them.

The sensorgram of SnO₂+Pd+Pt sensors, which are sensitive to a lower number of compounds (Table 2), shows a first zone that is assigned to hexanal, (E)-2-hexenal - and 1-pentanol if present at high concentration -, and a second zone due to three alcohols 1-hexanol, (Z)-3-hexen-1-ol, (E)-2-hexen-1-ol. However, the sensorgram profiles are very different from those described above when olive oils with sensory defects are analysed. Thus, for instance, the right side of the sensorgram is usually characterized by long chain aldehydes and acids coming from oxidation and fermentation processes and responsible for unpleasant attributes.

In general terms, the sensorgrams of low quality oils (lampante) are characterized by a higher area under the curve that points out a more intense 'neat' response. Aside from the neat intensity, the sensorgrams prove that the 'virgin olive oil defects' differ in the compounds that most contribute to sensor signal. Such differences justify the promising results of MOS sensors being used as screening devices to distinguish olive oil defects (García-González and Aparicio, 2003). Figure 8 shows the sensorgrams for standards of virgin olive oil of the 'winey-vinegary defect' types for the two kinds of sensors considered in this chapter.

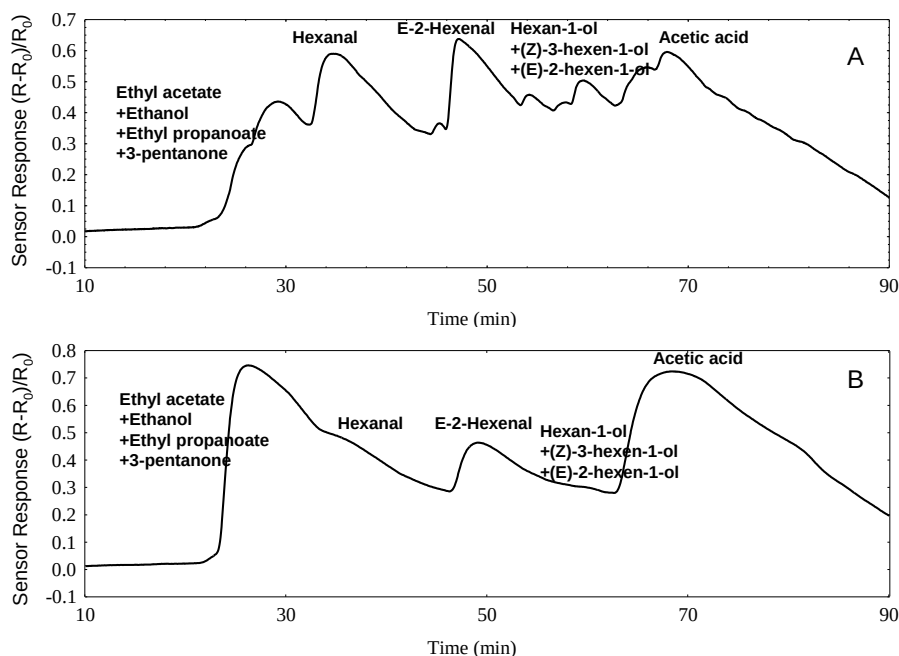


Figure 8: Sensorgrams of a Cr₂O₃+Ti sensor (A) and SnO₂+Pd+Pt sensor (B) of a virgin olive oil standard of winey-vinegary defect.

The responses of these sensors are mostly due to ethyl acetate, ethanol, hexanal, (E)-2-hexenal, hexan-1-ol, (Z)-3-hexen-1-ol, (E)-2-hexen-1-ol and acetic acid. Despite the high concentration of ethyl acetate in winey-vinegary oils (Morales *et al.* 2005), this compound produces a relatively low response in the Cr₂O₃ sensor while the response is very intense in the SnO₂ sensor. In the latter, the sensor response is mainly explained by ethyl acetate and acetic acid in the same proportion, while the rest of the compounds have an insignificant contribution. This result indicates, once again, that although this sensor shows sensitivity to a lower number of compounds, it would be the most suitable for detecting winey-vinegary defect in oils (García-González and Aparicio, 2002c).

Figure 9 shows the sensorgrams of the same sensors for a standard of olive oil with the defect musty-humidity. The sensorgrams are characterized by an intense response to hexanal and a low response to the other compounds, which are usually present at a very low concentration in oils with this kind of defect (Morales *et al.* 2005). Even though hexanal appears to be the most important

compound, minor responses are observed to other compounds, such as 2-heptanone, 2-octanol, 1-octen-3-ol, 3-methyl-1-butanol, γ (E)-2-heptenal, all of them being responsible for musty, wood, earthy and pungent attributes (Morales *et al.*, 2005).

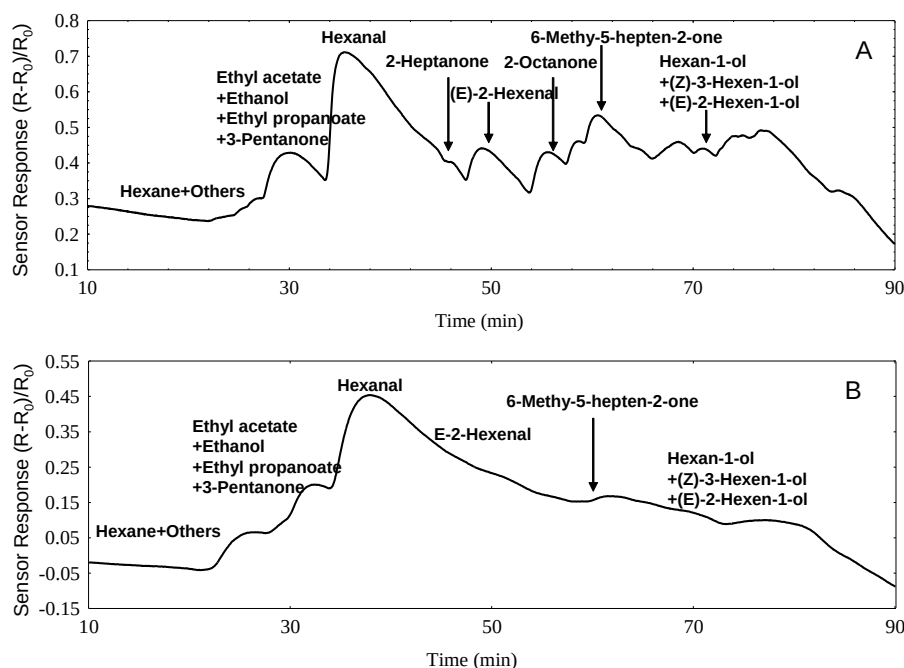


Figure 9: Sensorgrams of a $\text{Cr}_2\text{O}_3+\text{Ti}$ sensor (A) and $\text{SnO}_2+\text{Pd}+\text{Pt}$ sensor (B) of a virgin olive oil standard of musty-humidity defect.

Like the sensorgrams of the musty defect, the sensor responses for rancid oils are mainly due to hexanal (Figure 10). In particular, the response assigned to hexanal is outstanding in the sensorgram of the Cr_3O_2 sensor. However, unlike the musty-humidity defect, many other volatiles are responsible for the response, although in a varying degree of intensity. Thus, the response to numerous long chain aldehydes and acids coming from the fatty acid oxidation are observed at the end of the sensorgram resulting in a slow recovery of the baseline and a high area under the curve of the overall response.

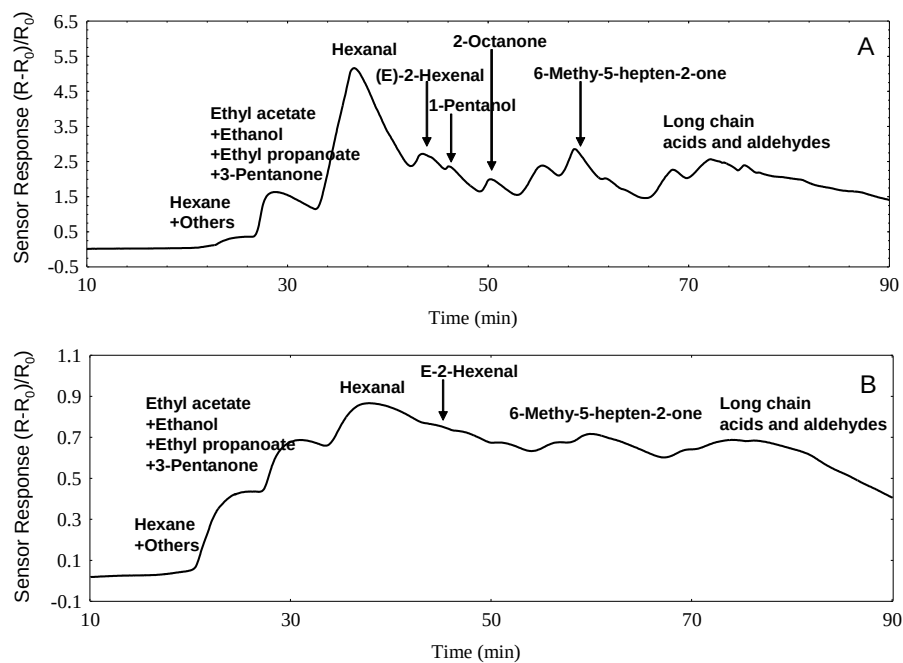


Figure 10: Sensorgrams of a Cr₂O₃+Ti sensor (A) and SnO₂+Pd+Pt sensor (B) of a virgin olive oil standard of rancid fusty.

The sensorgrams of oils with fusty defect (Figure 11) are determined by the high concentration of esters and acids. Thus, the beginning of the sensorgram (left side) is determined by the responses to ethyl acetate, butan-2-one, ethyl propanoate, butyl acetate and ethyl butanoate. Further these changes of intensity, a high response to hexanal is observed. By contrast, the response to (E)-2-hexenal is very low. However, the most characteristic feature of these sensorgrams is the later responses due to numerous acids such as acetic, butanoic, propanoic, pentanoic and hexanoic acids, most of them being produced by microorganisms (e.g. *Clostridium* bacteria) which carried out a fermentation process resulting in this sensory defect (Morales *et al.* 2005).

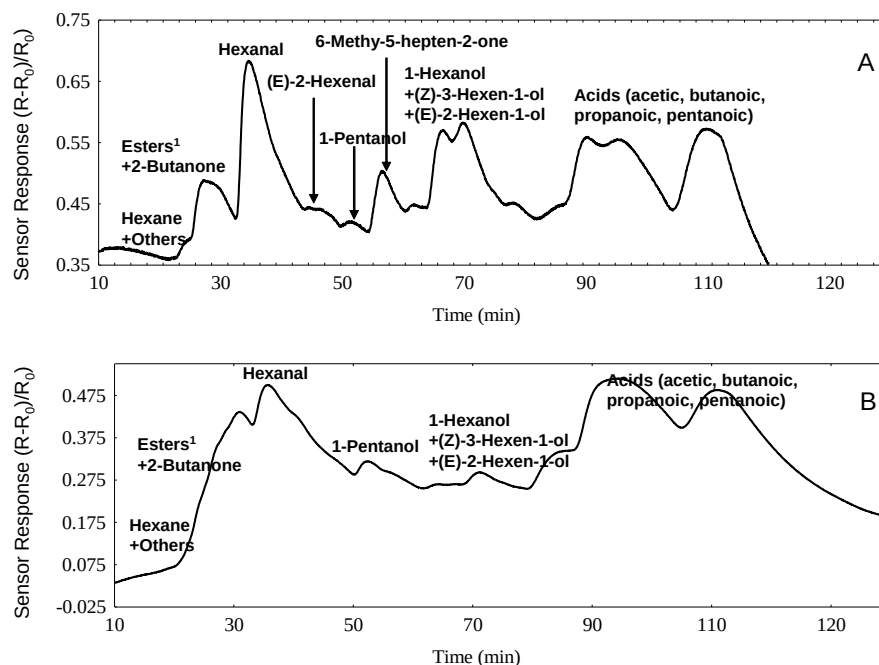


Figure 11: Sensorgrams of a Cr₂O₃+Ti sensor (A) and SnO₂+Pd+Pt sensor (B) of a virgin olive oil standard of fusty defect.

2.6 APPLICATION OF MOS SENSORS TO THE ANALYSIS OF VIRGIN OLIVE OIL AROMA. INTERPRETATION OF RESULTS ON THE BASIS OF GC-EN ANALYSIS

The virgin olive oil has been one of the matrixes traditionally chosen to check the feasibility and reliability of new sensor systems due to the commercial and economical importance of the aroma in this product. Some applications such as the classification by cultivar classification or traceability seems to be a long way from being commercially attractive propositions because they are very complex and they must deal with the large number of olive oils varieties and areas of production. However, other applications, most of them related to the assessment of the overall quality of olive oil, have proven to be reliable enough to be set up in industry once the repeatability problems are solved and the reproducibility of sensor manufacture is improved. These applications have been developed to

determine the shelf life of oils, to detect sensory defect, or to classify oil samples into different 'quality categories'. The latter application has been proposed by the authors as a rapid method to classify oils as 'lampante' and 'non-lampante' oils (García-González and Aparicio, 2003). The first category (lampante) corresponds to those oils with the lowest sensory quality and they are banned from being sold for human consumption by a European regulation (EC, 2003). The detection of this category is currently carried out by an 'assessment panel' that is in charge of the classification of samples based on sensory quality. Screening technique based on sensors would make it possible to reduce the workload of a sensory assessment (panel test), which is a costly methodology and has an important subjective component. Figure 12 shows the overlap between bell-shape distributions of the non-lampante and lampante samples qualified by the median of defects by several panel tests. Even though the panels were correctly trained and the official protocol to evaluate the samples was strictly applied, a median of defects between 3 and 4 would mean that a given sample could be classified into either lampante or non-lampante class with a probability of 60%.

Because of the complexity and fuzziness of this classification task, the maximum value of sensor responses is not enough to accomplish good classification and hence a special data processing stage was applied to extract the maximum sensor information with the smallest number of data points possible. Then, the response of the sensors was processed to produce five 'datapoints' per sensor by means of the WTS algorithm (García-González and Aparicio, 2002c). The WTS datapoints are obtained by applying 5 functions in different ranges along the response curve of the sensor. Therefore, these five datapoints represents an average value of the sensor response during the adsorption and desorption processes of the volatiles on the sensor's hot surface.

The possible use of a sensor system in the analysis of virgin olive oil aroma has been tested with two sensor arrays, each one with 6 MOS sensors (6 Cr_3O_2 sensors doped with Ti and 6 SnO_2 sensors doped with Pd and Pt). These sensors were the same as those used for the studies described above (Figure 3, Tables 4 and 5). 109 samples of olive oils, 50% of them from the lampante category, were analyzed and their sensor responses were submitted to a genetic algorithm (GA) (Goldberg, 1989) to select the sensors that allow the best classification between lampante and non-lampante oils, and later on, a multilayer perceptron (MLP) neural network (NN) was applied to obtain a 'mathematical model' (García-González and Aparicio, 2003). The samples were randomly distributed into three subsets: training, verification and validation subsets (Stasoft, 2003). A conjugate

gradient descent algorithm, applied to the training samples, was used to minimize the prediction error made by the NN. The training algorithm used the sum-squared error function to train the network and to report the root-mean-square (RMS) error. The weights and threshold were calculated by applying this algorithm to the training set. The best network was selected by means of the samples of the verification set. This set is also used to stop the training procedure to avoid that it overfits the training data. NN performance was determined by the validation set.

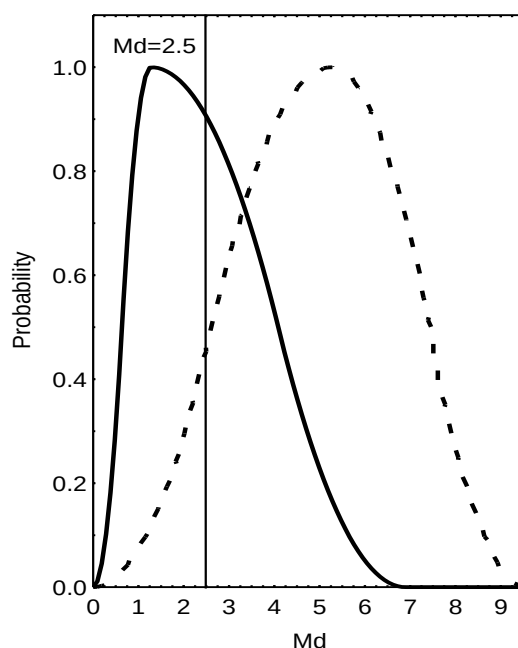


Figure 12: Bell-shape distributions of median of defects (Md) of non-lampante (continuous line) and lampante (dotted line) virgin olive oils (14 samples) evaluated by several panels. The limit between non-lampante and lampante virgin olive oil, according to the current European regulation, is marked with a vertical line. (Source: Garc ía-Gonz ález et al. 2007).

Figure 13 shows the MLP designed (MLP 11:6:1) after a training process with the conjugate gradient descent algorithm (58 epochs), and the MLP output values. The GA selected the information of 6 sensors (sensors coded with numbers 1, 5, 6, 8, 9, 11, 12) that resulted in a classification rate of 94%.

The sensors selected by the GA are those whose responses were highly correlated with compounds that are commonly concentrated in lampante oils (Figure 3), with the exception of sensor 5, which was highly correlated with the concentrations of (E)-2-hexenal, (E)-2-hexen-1-ol, (Z)-3-hexenyl acetate, responsible for green perceptions.

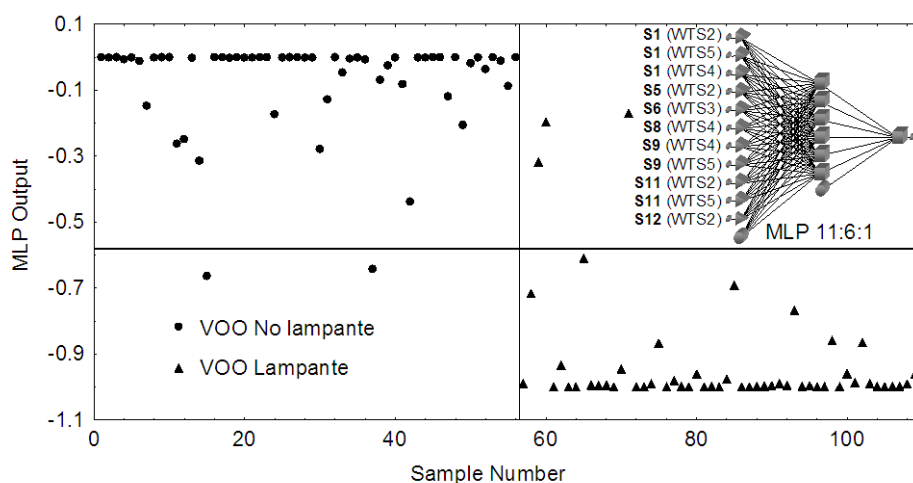


Figure 13: Results of applying multilayer perceptron (MLP) to lampante and non-lampante virgin olive oils (VOO), and schematic of the MLP designed for this purpose.

With regards to the metal composition of the sensors, around 50% of the unit inputs of the NN corresponded to Cr_2O_3 sensors while the rest of the inputs handled information from SnO_2 sensors, which points out that the selection of sensors was made on the basis of a balance between data from both kinds of sensors to reach the minimum overlapping information and maximum classification rate. The sensorgrams from the GC-EN analyses showed that both kinds of sensors provide rather different profiles that proves that the sensor signals from these sensors do not only differs in intensity, but also in their sensitivity to individual volatile compounds. The sensorgrams showed that the SnO_2 sensors are more sensitive to volatiles compounds that are characteristic of lampante oils. Thus, for instance, clear changes of response intensities were observed for hexanal, acetic acid, aldehydes in rancid oils, or acids in fusty oils. The response of Cr_3O_2 sensors, by contrast, is determined by a series of compounds that are present in both good and low quality oils such as hexanal,

(E)-2-hexenal and ethanol and they shows a medium or low response to compounds responsible for unpleasant sensory attributes (e.g. acetic acid or esters). Therefore, the good results shown in Figure 13 in distinguishing olive oils by quality can be explained by the intense responses of the sensors when some volatiles compounds that confer off-flavour are present, in contrast with the moderate responses for the volatile compounds that are commonly present in good quality oils.

2.7 CONCLUSION

Since the aroma analyzers based on gas sensors operate under a non separative principle and they are equipped with non-selective sensors, the results when classifying samples are questioned because they may be only due to a distinction of samples by total concentration of volatiles, and not to a different profile of volatiles. Therefore a true characterization of aroma by sensor systems should be supported by any experimental approach to explore the individual contribution of each volatile compound in the whole sensor response. Such a study would ensure that the results can be directly related to the quality of the samples, and it would allow prediction of the sensor responses for other samples with a different profile of volatiles, providing in this way a chemical basis to the electronic nose analyses.

Besides the ‘mathematical correlation’ and the sequential analyses of pure compounds, a new procedure based on the coupling of a sensor system to a gas chromatograph (GC-EN) adapted for samples with complex aroma provides definitive and unequivocal information about which compounds are the main ones responsible for the sensor responses. In addition to the chemical interpretation of sensor responses, the GC-EN equipment may be used as a routine analytical method for food aroma assessment characterizing the aroma of samples to a greater extent than conventional electronic nose but in a shorter time than a routine GC analyses.

Another application of this hyphenated system is to provide an experimental criterion to select sensors that are sensitive to the compounds with major odour impacts for a given application. Today there are many gas sensors that are commercially available and the presence of a dopant metal in the composition may drastically change their sensitivity, hence the analyst should select sensors based on their responses to real samples to search for an optimized sensor array.

Finally, the GC-EN and the proposed data processing based on derivative and smoothing of the signal can be an alternative non-destructive GC detector that allows the coupling to other instruments. Furthermore, a sensor array used as GC detector would provide sensorgrams that can be easily correlated with the human odour perception once the appropriate set of sensors are selected.

2.8 FUTURE RESEARCH DIRECTIONS

The implementation of a previous step of separation of compounds prior to being analysed by a sensor array and a new procedure optimized for complex odours such as food aroma open a new line to explore the possibilities of sensor systems in real applications. In this regard, the speed of the response seems to be a critical characteristic to be considered. Thus, further studies should be planned to examine the potential of other sensors such as acoustic sensors (SAW sensors) in food analysis with this methodology. Although some instruments based on SAW sensors and chromatographic separation have been commercialized, they are not optimized to be applied to food aroma. The micromachined GC columns, whose application in sensors has been recently reported, can improve the performance of these systems without significant reduction of the more advantageous properties of sensing instruments, the speedy reporting of results, low cost, and the options of portability and automation.

References

- Aparicio, R., Alonso, V. (1994). Characterization of virgin olive oils by SEXIA Expert System. *Prog. Lipid Res.*, 33, 1-2, pp. 29-38.
- Aparicio, R., Morales, M.T. (1998). Characterization of olive ripeness by green aroma compounds of virgin olive oil. *J.Agric.Food Chem.* 46, 3, pp. 1116-1122.
- Cimato, A., lo Monaco, D., Distante, C., Epifani, M., Siciliano, P., Taurino, A.M., Zuppa, M., Sani, G. (2006). Analysis of single-cultivar extra virgin olive oils by means of an Electronic Nose and HS-SPME/GC/MS methods. *Sens. Actuators B*, 114, 2, pp. 674-680.
- EC. (2003). Off. J. Eur. Communities, Regulation 1989/2003 L295, 6th November 2003.

- García-González, D. L., Aparicio, R. (2002a). Sensors: from biosensors to the electronic nose. *Grasas Aceites*, 53, 1, pp. 96-114.
- García-González, D. L.; Aparicio, R. (2003). Virgin olive oil quality classification combining neural network and MOS sensors. *J Agric Food Chem.*, 51, 12, pp. 3515-3519.
- García-González, D.L., Aparicio, R. (2002b). Detection of defective virgin olive oils by metal-oxide sensors. *Eur.Food Res.Technol.*, 215, 2, pp. 118-123.
- García-González, D.L., Aparicio, R. (2002c). Detection of vinegary defect in virgin olive oils by metal oxide sensors. *J. Agric. Food Chem.*, 50, 7, pp. 1809-1814.
- García-González, D.L., Aparicio, R. (2003). Virgin olive oil quality classification combining neural network and MOS sensors. *J. Agric. Food Chem.*, 51, 12, pp. 3515-3519.
- García-González, D.L., Tena, N., Aparicio, R. (2007). Characterization of olive paste volatiles to predict the sensory quality of virgin olive oil. *Eur.J.Lipid Sci.Technol.* 109, 7, pp. 663-672.
- Goldberg, D.E. (1989). Genetic Algorithms. Addison Wesley: Reading, MA.
- Kohl, D., Heinert, L., Bock, J., Hofmann, T., Schieberle, P. (2000). Systematic studies on responses of metal-oxide sensor surfaces to straight chain alkanes, alcohols, aldehydes, ketones, acids and esters using the SOMMSA approach. *Sens. Actuators B*, 70, 1-3, pp. 43-50.
- Masila, M.; Breimer, M., Sadik, O.A. (1999). Strategies for improving the analysis of volatile organic compounds using GC-based electronic nose. In: Electronic Noses & Sensor Array Based System, Design and Applications, pp 26-42. W.J. Hurst (Editor). Technomic Co, Inc.: Lancaster, PA.
- Morales, M.T.; Luna, G.; Aparicio, R. (2005). Comparative study of virgin olive oil sensory defects. *Food Chem.* 91, 2, pp. 293-301.
- Pardo, M., Sberveglieri, G., Gardini, S., Dalcanale, E. (2000). A hierarchical classification scheme for an Electronic Nose. *Sens. Actuators B*, 69, 3, pp. 359-365.
- Ragazzo-Sánchez, J.A., Chalier, P. Ghommidh, C. (2005) Coupling gas chromatography and electronic nose for dehydration and desalcoholization of alcoholized beverages: Application to off-flavour detection in wine. *Sens. Actuators B*, 106, 1, pp. 253-257.
- Statsoft, Statistica, User's Guide, release 5.5; Tulsa, OK, 2000.
- Williams, D., Pappas, G. Rapid identification of nerve agents sarin (GB) and soman (GD) with the use of a field-portable GC/SAW vapor detector and liquid desorption front-end device (1999). *Field Anal.Chem.Technol.* 3, 1, pp. 45-53.
- Zampolli, S., Elmi, I., Stürmann, J., Nicoletti, S., Dori, L., Cardinali, G.C. (2005) Selectivity enhancement of metal oxide gas sensors using a micromachined gas chromatographic column. *Sens. Actuators B*, 105, 2, pp. 400-406.

Chapter 3

Systematically Improving the Performance of Semiconductor Gas Sensors: Optimization of the Temperature Modulation Mode via Pseudo-random Sequences – Using Fuzzy Artmap Neural Networks

Alexander Vergara and Eduard Llobet

3.1 INTRODUCTION

In recent years, many papers have appeared in the literature concerned with investigating the analysis of single gases and multi-component mixtures using semiconductor gas sensors and intelligent systems (ISs) engineering techniques. However, some important problems associated with this approach, such as the sensor lack of selectivity (i.e., sensors respond to a wide range of species) and lack of reproducibility (i.e., changes in sensor responsiveness due to ageing or memory effects), are the reasons why this is still a field of active research (P. T. Moseley et al. 1987, 1991). To overcome these difficulties, different strategies have been developed in parallel, yet with limited success. Some of these are related to the development and improvement of materials and technologies (Sberveglieri 1995; Cosandey et al. 2000; Xu et al. 1991; Göpel 1994; Demarne and Sanjines 1992; Hohl 1990; Shimizu and Egashira 1999; Yamazoe 1991; Mukhopaghyay et al. 1998; White 1997, Martinelli et al. 1999; Martinelli and Ghiotti 1999) (e.g., use of new metal oxide material or new catalyst and dopants), or the use of sample conditioning (e.g., chromatography-based methods) (www.znose.com; www.gerstel.de; www.alpha-mos.com; www.hkr-sensor.de; www.smartnose.com; www.smartnose.com; Fenaille et al. 2003; Marcos Lorenzo et al. 2002; P é è s et al 2002; Marcel Dekker Inc. 2002; Schaller 2000, Arthur and Pawliszyn 1990; Arthur et al 1991, Chen and Pawliszyn 1995; Vinaixa et al.

2005) and pre-treating methods (e.g., techniques of selective pre-concentration) (Althainz et al. 1994; Park et al. 2002; Feng et al. 1994; Dori et al. 2000; Privat et al. 1998; Schaller et al. 2000). Other widely used techniques include taking benefit of the unavoidable partially overlapping sensitivities by using sensor arrays and pattern recognition techniques (Gardner et al. 1999, Persaud and Dodd 1982; Gardner 1991, Kohl 1990; Shurmer et al. 1990) or the use of dynamic features from the gas sensor response (e.g. A.C. operation mode or temperature modulation) (Llobet 1997; Duran et al. 2006, Clifford et al. 1981, 1982-1983, Llobet et al. 1997 and 1998). Although these techniques are usually applied separately, their simultaneous application should lead to a significant improvement in the selectivity of sensors.

In the last few years the use of signals resulting from the operation of semiconductor gas sensors in a temperature modulated mode has been, by far, the most studied dynamic method for reducing gas sensors problems (Le Vine 1975; Eicker 1977, Owen 1980; Advani et al. 1983; Sears et al. 1989 (a) and (b); Nakata et al. 1992, 1996 and 1998). When the working temperature of a gas sensor is modulated, the kinetics of the gas-sensor interaction is altered and this leads to characteristic response patterns. As the sensor response is different at different working temperatures, measuring the sensor response at n different temperatures is similar to the use of an array comprising n different sensors. This makes it possible to measure multivariate information from every single sensor and helps to keep the dimensionality of the measurement system as low as is needed to solve a specific application. Although good results have been reported, until now, selecting the frequencies for modulating the sensor operating temperature has been very often a trial and error process and it is not an accurate method to ensure that the best results are achieved for a given application. But this has changed in the last few years since several authors have developed methods to systematically design appropriate ways in which to use of sensor dynamics (Kunt et al. 1998). The principal objective of this chapter is to present one of such techniques, which permits a systematic optimization of the process of frequency selection for modulating the operating temperature of metal oxide gas sensors using maximum-length pseudorandom sequences (PRS) (either binary or multi level sequences) in order to increase their selectivity. This method is borrowed from the field of system identification and was introduced and developed in the area of gas sensors by Vergara et al. (2005 (a), 2005 (b), 2007) in the last few years. It involves the optimisation process which is conducted by selecting among the spectral components of the impulse response estimates of the

sensor-gas system (the few that better help either discriminating or quantifying the target gases for a given gas analysis application). Therefore, since spectral components are directly related to modulating frequencies, the selection of these components results in the determination of the optimal temperature modulating frequencies. The systematic studies are further validated by solving a practical application showing new results obtained from synthesizing multi-sinusoidal signals at the optimal frequencies previously identified by using these sequences.

The structure of the chapter is as follows:

Section 2 of this chapter deals with a mathematical description and definition of a general system. Furthermore, an introduction to perturbation signals for time and frequency domain system identification is given. Therefrom the ways to obtain the impulse response of a system are presented. Particularly, the generation and application of pseudorandom sequences signals (PRS) of maximum length (either binary or multilevel) is described. This is followed by an explanation of how PRS can be used to identify systems and, how the method can be extended to study temperature-modulated gas sensors systematically. Further, Section 3 describes the experimental methods and the experimental design approach that needs to be implemented in order to optimise the temperature modulation of semiconductor gas sensors.

Section 4 and 5 present the quantitative and qualitative results for gas analysis of micro-hotplate gas sensors modulated by means of PRS signals (either binary or multilevel). Particularly in Section 4, the analysis of the thermal response of the gas sensor heating element, in order to obtain the cut-off frequency of the coated membranes and there from to determine the clock frequency (f_c) and the length of the PRS signal will be detailed. Once these parameters are determined, the problem envisaged in these sections consists of the building of calibration models for the quantitative and qualitative analysis of pollutant species in order to determine in a systematic way the optimal set of modulating frequencies. In experiments an estimate of the impulse response is computed and its spectral components are studied. A few selected spectral components correspond to the frequencies that will be used in the qualitative and quantitative analysis of the gases studied. Once the optimal set of modulating frequencies has been determined a further validation of these results is envisaged by modulating the working temperature of gas sensors by means of multi-frequency sinusoidal signals. The frequencies within these signals were the ones selected during the training phase in the first part of every one of the experiments presented. The

validation process is preformed with new sets of measurements with the same type of pollutant species and using different metal oxide gas sensor micro-arrays (of the same type). This part is attended to in Section 6 of this chapter. Finally, the conclusions derived from the studies performed and an outlook on the methods and aspects towards more effective semiconductor gas sensors will be discussed in the conclusions.

3.2 GAS SENSOR/SYSTEM IDENTIFICATION

3.2.1 Description of systems: relationship between their input and output.

It is always best to begin at the beginning. Since this chapter is about how the performance of a semiconductor gas sensors is systematically improved, and the method is borrowed from the field of system identification, the first question to answer is what is a system? One way to define a system is as anything that performs a function, that is, it operates on something (e.g., any time-varying physical phenomenon which is intended to convey information) and produces something else. Another way to define a system is as anything that responds when stimulated or excited (Roberts 2004 and Openheim 1989). A block diagram of a system is shown in Figure 1. Although systems can be of many different kinds, they have some features in common: A system operates on signals at one or more inputs to produce signals at one or more outputs. In this particular case the gas-sensor pair is considered as a system where, on the one side, the temperature stimulation or excitation of the heating element and the gases to be measured are the inputs, and on the other side, the sensor operating temperature and the sensor response to the gases are the outputs of the system.

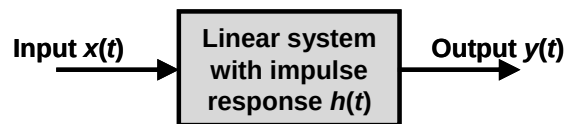


Figure: 1 Block diagram of a linear system.

There may be noise in the system and/or in the measuring device and this is commonly represented by a noise signal $n(t)$ added to the system output $y(t)$ to give the measurable output signal $z(t)$, so that

$$z(t) = y(t) + n(t) \quad (1)$$

The system is assumed to be linear, so that the principle of superposition applies. This means that if $x_1(t)$ produces a response $y_1(t)$, $x_2(t)$ produces a response $y_2(t)$ and so on, then the sum $[x_1(t) + x_2(t) + \dots]$ produces a response $[y_1(t) + y_2(t) + \dots]$.

There is a systematic method to find out how linear systems respond to excitations or inputs. This systematic method is called convolution. The convolution technique for finding the response of a linear system is based on a simple idea. The system response $y(t)$ is given by an integral of weighted inputs which have occurred in the past. The past values of input are multiplied by a function $h(t)$, called the weighting function of the system and the mathematical equation describing the relationship is:

$$y(t) = \int_0^{\infty} h(\lambda)x(t - \lambda)d\lambda \quad (2)$$

where λ is a time variable. If the characteristics of the system do not change with time, then the weighting function is also equal to the unit impulse response of the system. This is the response $y(t)$ to an input $x(t) = \delta(t)$, where $\delta(t)$, the Dirac delta distribution centred at $t = 0$, is a distribution of height which tends to infinity and duration which tends to zero such that (height $\times$ duration) = 1. This Eq.(2) is the so-called convolution.

In a practical situation, inputs which have occurred at times greater than T_s in the past have negligible effect on the present output of the system so the previous equation may be modified to:

$$y(t) = \int_0^{T_s} h(\lambda)x(t - \lambda)d\lambda \quad (3)$$

where T_s is called the setting time of the system. Combining Eq.(1) and (3), the measurable output is given by:

$$z(t) = \int_0^{T_s} h(\lambda)x(t - \lambda)d\lambda + n(t) \quad (4)$$

All equations and techniques shown until now were destined to find the response of continuous-time systems. So, on the other hand the convolution technique for finding the response of a discrete-time linear system is based on a simple idea. No matter how complicated an excitation signal is, it is simply a

sequence of discrete-time impulses. To find the response of a linear system, it is necessary to find the response of the system to one impulse at a time, all these responses then form the actual overall response. The responses to those impulses all have the same functional form except that they are shifted in time. Therefore, if a response of a linear system to a unit impulse excitation occurring at time $n = 0$ is obtained, the response to any other excitation can be easily found. Therefore, the use of the convolution technique begins with the assumption that the response of the system to a unit impulse excitation occurring at time $n = 0$ has already been found, and it is called the impulse response $h[n]$.

Therefore, since the response of the system to a single unit impulse occurring at discrete time $n = 0$ is known, by means of the convolution of the impulse response $h[n]$ of the system and the input $x[n]$, the output of the system $y[t]$ can be found by the convolution expression as in the case of continuous-time systems. So, the discrete equivalent for Eq.(3) and (4) (convolution expressions) respectively are shown as follows:

$$y[n] = \sum_{m=0}^L h[n]x[n-m] \quad (5)$$

$$z[n] = \sum_{m=0}^L h[n]x[n-m] + n[n] \quad (6)$$

where the time T_s is now denoted by L .

In theory Eq.(4) or its discrete equivalent Eq.(6), could be solved to give an estimate of the system weighting function from almost any input-output records.

3.2.2 Design and generation of PRBS signals.

One of the most useful types of periodic signals for process identification is the pseudo-random binary sequence (PRBS) (Barker 1967; Everett 1966; Godfrey 1980), which has the following properties:

- (1) The signal has two levels (e.g. 0 or 1), and it can switch from the one level to the other only at certain event points $t = 0, \Delta t, 2\Delta t$, etc. Therefore, since the signal changes level at *predetermined* event points, the PRBS is deterministic and experiments performed using this signal are repeatable.
- (2) The PRBS is periodic with period $T = L\Delta t$, where L is an odd integer. The most commonly used PRBS signals are based upon maximum-length sequences (MLS) of length $L = (2^n - 1)$ where n is the order of the sequence.

The reason why PRBS are so popular is that they can be readily generated using a feedback shift register circuit and simple XOR logic gates. The number of shift registers employed is the order, n , of the sequence (Godfrey 1993).

- (3) In any one period, there are $\frac{1}{2}(L+1)$ intervals when the signal is at one level and $\frac{1}{2}(L-1)$ intervals when it is the other.

These properties imply that a PRBS shares some properties with white noise but having the advantage of being repeatable. A PRBS of length $L = 2^n - 1$ is generated by an n -stage shift register ($n > 1$) with an OR-exclusive logic gate feedback to the first stage. The XOR gate performs a modulo 2 addition, and the logic values are given by:

$$1 \oplus 1 = 0 \oplus 0 = 0$$

$$1 \oplus 0 = 0 \oplus 1 = 1$$

Table 1: PRBS from a 3-stage shift register with feedback connections corresponding to a primitive polynomial (modulo 2).

Number of clock pulses	Shift register stage		
	1	2	3
1	0	0	1
2	1	0	0
3	1	1	0
4	1	1	1
5	0	1	1
6	1	0	1
7	0	1	0
8	0	0	1

Such a circuit goes through a set of states and eventually repeats itself after $2^n - 1$ clock pulses, where n is an integer > 1 . Figure 2 shows an example of a shift register circuit that generates a sequence of length $L=7$ (a) and a sample of a 7-length PRBS signal (b). The shift register can be started with any binary number except 0, 0, 0 (which would give a sequence of length unity). In Table 1 the register started (arbitrarily) with 0, 0, 1 and it can be seen that this number reappears after $2^3 - 1$ clock pulses (feedback are from stages 1 and 3). Each binary number from 0 0 1 to 1 1 1 appears as the register contents exactly once

during the cycle. This is a general result for all binary m -sequences: each binary number from 1 to $2^n - 1$ appears in the sequences exactly once.

A shift register with the appropriate connections to generate a PRBS has a characteristic equation in the delays, D , (introduced by each stage of the shift register) according to a primitive binary polynomial (modulo 2). A list of primitive polynomials can be found in (Watson 1962; Stahnke 1973).

If the polynomial $p(x) = p_0x^n + p_1x^{n-1} + \dots + p_{n-1}x + p_n$ is a primitive polynomial (modulo 2), then the characteristic equation in the delays, D , introduced by a shift register of length n is (Briggs et. al. 1965):

$$p_0D^n + p_1D^{n-1} + \dots + p_{n-1}D + p_n = 0, \text{ modulo } 2 \quad (7)$$

Therefore,

$$p_nX = -p_0D^nX - p_1D^{n-1}X - \dots - p_{n-1}DX, \text{ modulo } 2 \quad (8)$$

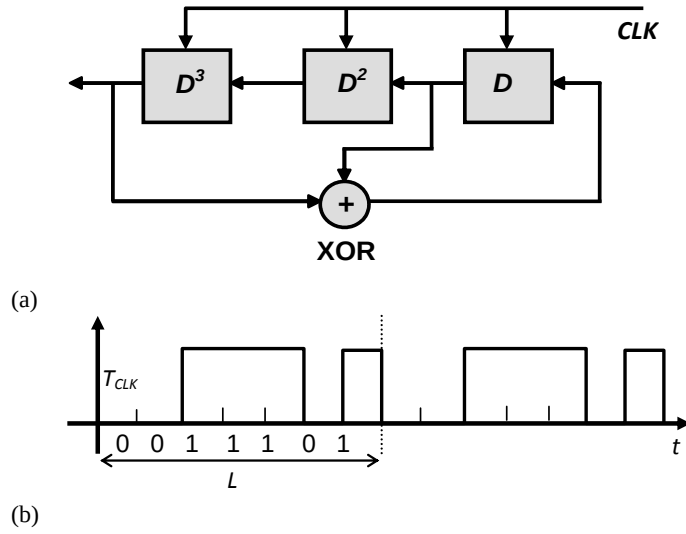


Figure 2: Shift register circuit used to generate a PRBS of length $L = 7$ (a) and 2 periods of a sequence generated (b). CLK is the clock signal and T_{CLK} is the clock period.

where X is the input sequence to the shift register and D^iX is the output at the i^{th} stage of the register. Considering that modulo 2 subtraction or addition is equivalent, Eq.(8) can be rewritten as follows:

$$p_nX = p_0D^nX + p_1D^{n-1}X + \dots + p_{n-1}DX, \text{ modulo } 2 \quad (9)$$

For maximum length binary sequences to be obtained, p_0 and p_n have to be set to unity and the remaining coefficients in Eq.(9) are either 0 or 1. For the example shown in Figure 2, the characteristic equation is:

$$D^3 + D + 1 = 0, \text{ modulo } 2 \quad (10)$$

and the corresponding primitive polynomial is $x^3 + x + 1$, modulo 2.

3.2.3 Design and generation of MLPRS signals.

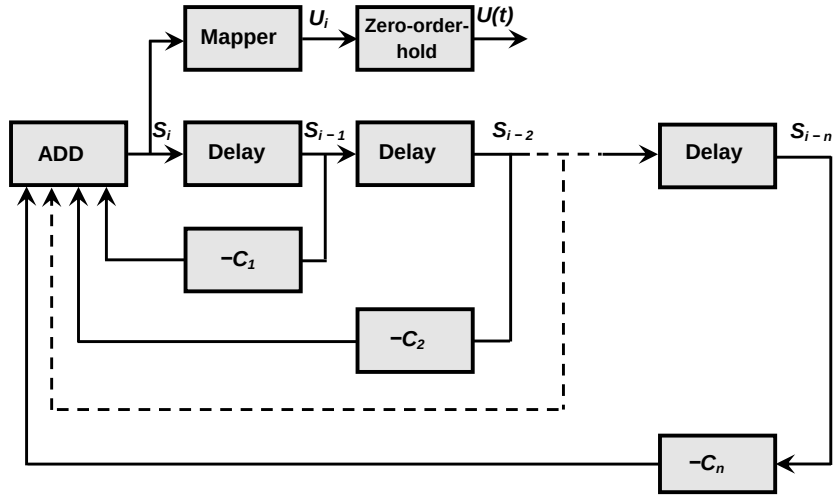
The theory behind the generation of multilevel pseudo random sequences (MLPRS) based on multilevel maximum length signals is well developed. As in binary pseudo random sequences, MLPRS are periodic, deterministic signals, and have an autocorrelation function similar to white noise (Vergara et al. 2005 (a) and (b)). MLPRS exist for the number of levels, q , equal to a prime or a power of a prime $p(>1)$, i.e. for $q = 2, 3, 4, 5, 7, 8, 9, 11, 13, \dots$ (Zierler 1959; Godfrey1993). The relevant theory behind MLPRS is based on the algebra of finite fields (Albert 1956; Barker 1986). When q (the number of levels) is a prime, the digits of the sequence are the integers $0, 1, \dots, (q - 1)$ and the sequence can be generated by a q -level, n -stage shift register with feedback to the first stage consisting of the modulo q sum of the outputs of the other stages multiplied by coefficients $a_1, \dots, a_n$ which are also the integers $0, 1, \dots, (q - 1)$. The length of such a sequence $\{x_r\}$ is $q^n - 1$, where n is an integer. After $q^n - 1$ digits, the sequence repeats itself. MLPRS signals are generated in a similar manner to that for the binary ones using a shift register and modulo addition. The generator of such a sequence and an example of a 5-level sequence (fragment) are shown in Figure 3(a) and (b), respectively. The initial state of the shift register can be any combination of length n of the values $0, 1, \dots, (q - 1)$, exception made of n zeros. Each combination of these values (except n zeros) appears as the state of the register exactly once during a period of the MLPRS. The sequences can be generated by a q -level shift register with feedback to the first stage consisting of the modulo q sum of the outputs of the other stages multiplied by coefficients $c_1, \dots, c_n$, which are integers that lie in the range $[0, q - 1]$.

For ease of reference, the important properties of q -level maximum length signals are summarized below:

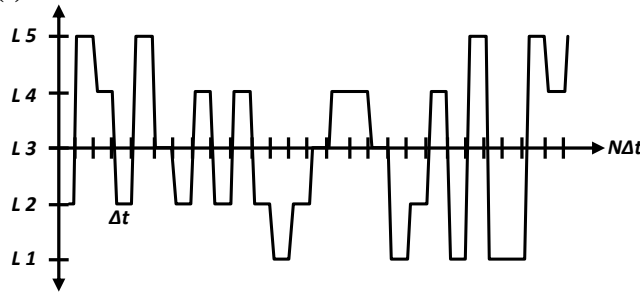
- (1) Basic signals comprise the integer elements $0, 1, 2, \dots, q - 1$.
- (2) Each cycle has $(q^n - 1)$ digits, where n is an integer ($n > 1$) corresponding to

the number of stages in the equivalent q -level feedback shift register (FSR) generator.

- (3) The number of zeros in each cycle is $(q^{n-1} - 1)$.
- (4) The number of each of the non-zero elements in a cycle is q^{n-1} .
- (5) Each cycle comprises $(q - 1)$ sequential 'blocks' of digits of length $(q^{n-1})/(q - 1)$ digits, and hence $(q - 1)$ is always a factor of (q^{n-1}) for all q and n .
- (6) From any reference point in the cycle, the block comprising the subsequent $(q^{n-1})/(q - 1)$ digits can be derived by multiplying (modulo q) all digits in the preceding block of length $(q^{n-1})/(q - 1)$ by a primitive element g of the Galois field of q elements $GF(q)$.



(a)



(b)

Figure 3: (a) A q -level pseudo-random maximum length sequence generator. (b) Fragment of a 5-level pseudo random sequence.

In a similar way to the binary case, the feedback configuration of the shift register is given by a primitive polynomial, modulo q . If $\alpha_n X^n + \alpha_{n-1} X^{n-1} + \dots + \alpha_1 X + \alpha_0$ is primitive, modulo q (i.e. the polynomial is irreducible and thus, has no factors modulo q), then the connections to the first stage are given by:

$$\alpha_0 X = -\alpha_1 DX - \dots - \alpha_{n-1} D^{n-1} X - \alpha_n D^n X, \text{ modulo } q \quad (11)$$

where X is the input sequence to the shift register, DX is the sequence at the output of the first stage of the register, so that $D^n X$ is the sequence at the output of the last stage of the n -stage register. $\alpha_0 = 1$ and the remaining coefficients $\alpha_1, \dots, \alpha_n$ have integer values in the range 0 to $q - 1$.

Eq.(11) can be rewritten as follows:

$$X = a_1 DX + \dots + a_{n-1} D^{n-1} X + a_n D^n X, \text{ modulo } q \quad (12)$$

where

$$a_i = (q - \alpha_i), \quad i = 1, 2, \dots, n \quad (13)$$

For example, the polynomial $3x^4 + 4x^3 + 4x^2$ is primitive, modulo 5, and the feedback to the input of the four-stage, five-level shift register is given by:

$$X = D^2 + D^3 X + 2D^4 X, \text{ modulo } 5 \quad (14)$$

Table 2: Feedback configuration for some q -level m-sequences of length $L = q^n - 1$.

Q	n	N	Feedback coefficients				
			a_1	a_2	a_3	a_4	a_5
5	2	24	1	3	-	-	-
5	3	124	0	1	2	-	-
5	4	624	0	1	1	2	-
5	5	3124	0	0	0	1	2
7	2	48	1	4	-	-	-
7	3	342	0	1	5	-	-
7	4	2400	0	1	1	4	-
11	3	1330	0	10	7	-	-
13	3	2196	0	12	7	-	-
17	3	4912	0	16	14	-	-
19	3	6858	0	18	15	-	-
23	3	12166	0	22	20	-	-
29	3	24388	0	28	18	-	-
31	3	29790	0	30	17	-	-

The sequence resulting from Eq.(14) has logic values 0, 1, 2, 3 and 4 and its period is $5^4 - 1 = 624$. A list of all irreducible polynomials modulo 3, 5 and 7 for

different values of n can be found in (Church). More details on the properties of q -level maximum length signals can be found elsewhere (Godfrey 1993).

Finally, the feedback connection (with modulo q) for generating q -level m -sequences are given in Table 2, for $q = 5$ and $n = 2, 3, 4$ and 5 and for $q = 7$ and $n = 2, 3$, and 4 , derived from the primitive polynomials listing of Church in 1935 (Church) and for $n = 3$ and $q = 11, 13, 19, 23, 29$, and 31 (all prime numbers), based on those given by Everett (1966).

3.2.4 Pseudo Random sequences (PRS) and system identification.

The impulse response, $h(t)$, is the main descriptor of a linear invariant system. Among the different strategies to estimate impulse response, noise methods allows the system under study to be excited for long enough time to supply it with the necessary energy to obtain a good estimate of $h(t)$. Exciting the system with white noise signals ensures a homogeneous distribution of the energy over a large frequency range. Periodic pseudo-random sequences (PRS) signals, either binary or multi-level, can potentially be employed as exciting signals for systems. Since PRS signals have a low crest factor (i.e. low peak-to-average factor) they minimize the risk of saturating the system under study. In practice, this means that the signals contain enough energy to obtain a good signal-to-noise ratio (SNR) in a wide frequency range (i.e., measurement with high dynamic range) and avoid sensor non-linearity caused by signals with high crest factors (e.g. impulsive signals). Furthermore, because the noise signal is deterministic, reproducible results would be obtained, provided that the conditions of the system under analysis remain unchanged.

The power spectrum envelope of either binary or multi level PRS of maximum length is almost flat up to a frequency equal to $0.45 \times f_c$, where f_c is the frequency of the clock signal applied to the shift register used to generate the signal. The power spectrum is discrete and the separation between spectral lines (i.e. the spectral resolution) is f_c/L , where L is the length of the PRS signal. The power spectrum envelope is similar to the power spectrum of white noise up to the -3 dB cut-off frequency: $0.45 \times f_c$. Figure 4 shows the power spectrum of a PRS signal. Therefore, pseudo-random sequences are routinely used to obtain an estimate of the impulse response of systems (Ljung 1987).

When the pseudo-random sequence is a maximum length signal, the impulse response estimate, $\hat{h}(n)$, can be obtained by computing the circular cross-correlation between the excitatory signal, $x(n)$, and the response signal, $y(n)$

(Ljung 1987). The circular cross-correlation of two sequences x and y in $\Re L$ may be defined as:

$$\hat{h}(n) = \frac{1}{L} \sum_{l=0}^{L-1} y(l+n)x(l), \quad n = 0, 1, 2, \dots, L-1. \quad (15)$$

The cross-correlation is circular since $l+n$ is interpreted modulo L , where L is the length of the sequence. In Eq.(15), L is a normalization factor that is optional. The circular cross-correlation between input and output sequences can readily be interpreted in terms of $\hat{h}(n)$ because the autocorrelation function of the PRS of maximum length signal is of approximately impulsive form.

Most non-linear systems can be described by their Volterra functional expansion (Volterra 1930):

$$\begin{aligned} y(t) = & \int_0^\infty h_1(\lambda_1)u(t-\lambda_1)d\lambda_1 + \int_0^\infty \int_0^\infty h_2(\lambda_1, \lambda_2)u(t-\lambda_1)u(t-\lambda_2)d\lambda_1d\lambda_2 + \dots \\ & + \int_0^\infty \dots \int_0^\infty h_n(\lambda_1, \lambda_2, \dots, \lambda_n)u(t-\lambda_1)u(t-\lambda_2) \dots u(t-\lambda_n)d\lambda_1d\lambda_2 \dots d\lambda_n + \dots \end{aligned} \quad (16)$$

where $h_n(\lambda_1, \lambda_2, \dots, \lambda_n)$ is called the Volterra kernel of order n (the kernel of order 1 accounts for the linear dynamics).

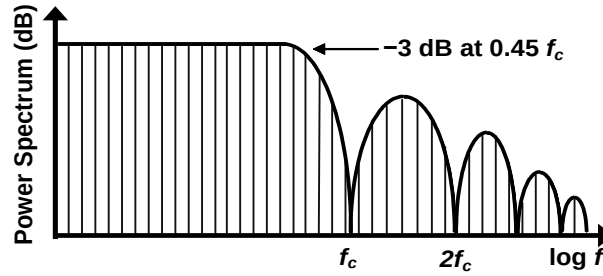


Figure 4: Discrete power spectrum of a PRS of maximum length signal. Spectral resolution is f_c/L , where f_c is the frequency of the clock signal applied to the shift register and L is the length of the sequence.

When the excitatory signal, $x(n)$, which is an PRS of maximum length signal, and the response signal, $y(n)$, are cross-correlated over an integer number of periods of the excitatory signal, contributions from even-order kernels disappear,

which leaves the linear term and terms involving kernels of order 3, 5, etc. Since in most cases, higher order kernels have a small effect on the output (this effect diminishes rapidly when the order, n , increases), a better estimate of the linear kernel is obtained by removing the effects of the second order kernel. This property is advantageous to better study the dynamics of temperature modulated metal oxide gas sensors.

In other words, signals based on multi-level maximum length pseudo random sequences (i.e. with a number of levels higher than 2) are preferred to binary signals because the former provides a better estimate than the latter signals of the linear dynamics of a system with non-linearity, which is normally the case with gas sensors.

3.2.5 Gas/sensor systems identification.

Maximum length pseudorandom sequence signals can be of help to systematically study the effects of the frequencies used to modulate the working temperature of metal oxide gas sensors. When either a binary or multi-level PRS signal voltage is applied to the heating element of a micro-hotplate gas sensor, its working temperature is modulated in a wide frequency range. If this sensor is in the presence of a gas diluted in air, the impulse response of the sensor-gas pair can be estimated from the circular cross-correlation of the PRS temperature-modulating signal $x(t)$ and the sensor response sequence $y(t)$. Because metal oxide sensors change their resistance in the presence of gases, the sensor response sequence is the resistance transient of the temperature-modulated sensor. By using spectral analysis on the impulse response estimate, $\hat{h}(t)$, important information about the modulation frequencies that are useful to discriminate between different gases and to estimate gas concentration is obtained in a systematic way. This process is illustrated in Figure 5.

PRS of maximum length signals have a characteristic that make them especially suitable for the study of metal oxide gas sensors: Since their energy is distributed over L pulses of small amplitude, such signals have a low crest factor (i.e. low peak-to-average factor). In practice, this means that the signals contain enough energy for it to be possible to obtain a good SNR in a wide frequency range (i.e., measurement with high dynamic range) and to avoid sensor non-linearities caused by signals with high crest factors (e.g. impulsive signals).

By choosing the clock frequency (f_c) used to generate the maximum length based PRS signal and the length (L) of the sequence, the frequency range of

modulating frequencies under study and the resolution are set. The range of modulating frequencies that are systematically studied is from the D.C. to $0.45 \times f_c$, which corresponds to the frequency range where the power spectrum of the PRS signals is approximately flat (e.g. white noise-like, see Figure 4). The resolution is (f_c/L) , which corresponds to the spectral resolution of the PRS power spectrum. The higher the clock frequency is, the wider the range of modulating frequencies studied is. If the clock frequency is increased, the length of the PRS signal should be increased too, for the spectral resolution not to degrade.

Once the feature extraction methods have been applied to the gas sensor transient response (because of the temperature modulation) a simple variable selection procedure needs to be implemented in order to select a short number of features (i.e., the ones more representative of the problem) among the total number of variables extracted.

A criterion is defined to rate the resolution power of each variable selected. For gas identification purposes, the measurements are grouped into as many categories as there are gas species (i.e. single species or their mixtures). In a similar way for quantification purposes, one specific model per species or gas-mixture is built. Therefore, for every quantification model, the measurements are grouped in as many categories as gas or gas mixture concentrations are measured.

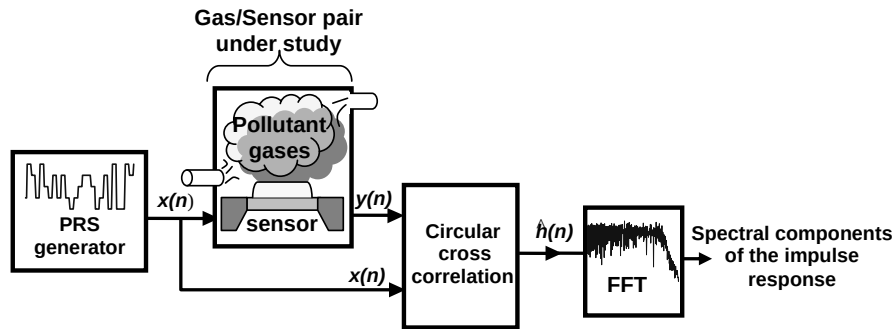


Figure 5: Study of the sensor/gas system using PRS signals. The PRS voltage signal, $x(n)$, is input to the heating element of a micro-hotplate gas sensor. The transient of the sensor conductance (i.e. the response in the presence of gases), $y(n)$, is recorded. An estimate of the impulse response, $\hat{h}(n)$, can be computed via the circular cross-correlation of $x(n)$ and $y(n)$. Finally, by performing the FFT of $\hat{h}(n)$, the spectral components of the impulse response estimate are found.

For each set of variable selected, intra-group and between-group variances are computed:

- (1) An intra-group variance is defined as the variance of the set of features or variables extracted which are considered to be within a given group of measurements. Two examples of intra-group variances are as follows: the variance of a spectral component within the measurements of a gas, no matter the concentration (case of gas identification) or the variance within the measurement of a given concentration of pollutant specie (case of quantification). The intra-group variance of a variable set j , can be defined as:

$$\sigma_{\text{intra},j}^2 = \frac{\sum_{i=1}^n (N_{ji} - \mu_j)^2}{n-1} \quad (17)$$

where n is the number of measurements within the group, N_{ji} is the value of variable set j for measurement i and μ_j is the mean of variable set j over the measurements within the group.

- (2) In a similar way, for every variable set, a between-group variance is defined as the variance within the category means *centroids*. Therefore, the between-group variance can be defined as:

$$\sigma_{\text{be},j}^2 = \frac{\sum_{i=1}^d (\mu_{ji} - \bar{\mu}_j)^2}{d-1} \quad (18)$$

where μ_{ji} is the mean of variable set j over the measurements within group i , d is the number of different groups and $\bar{\mu}_j$ is the mean over the μ_{ji} .

Finally, a figure of merit for the resolution power, RP , of a spectral component is defined as the ratio:

$$RP_j = \frac{\sigma_{\text{be},j}^2}{\sigma_{\text{intra},j}^2} \quad (19)$$

The higher the figure of merit defined in Eq.19 is, the more important is the spectral component considered to correctly discriminate or quantify gases. Therefore, a small set of spectral components, which comprises those that have the higher figure of merit, have been selected. These spectral components correspond to the temperature-modulating frequencies that lead to a better

discrimination or quantification of the gases studied and can be related to a specific temperature-modulation frequency by the following expression:

$$f_m = \frac{SC_m \times f_c}{L} \quad (20)$$

where SC_m is the spectral component number and f_c/L is the spectral resolution.

3.3 EXPERIMENTAL SETTINGS

In this section, the micro-hotplate gas sensors used to illustrate the techniques described, the measurement system layout and the experimental set-up including the pollutant gases measured are presented in detail. The experiments presented here were performed using micro-hotplate gas sensors. The sensor substrates consisted of four-element integrated micro-hotplate arrays fabricated using microsystems technology. The gas sensitive materials used were tin oxide and tungsten oxide. The active layers were deposited onto microelectronic substrates by either sputtering or screen printing techniques.

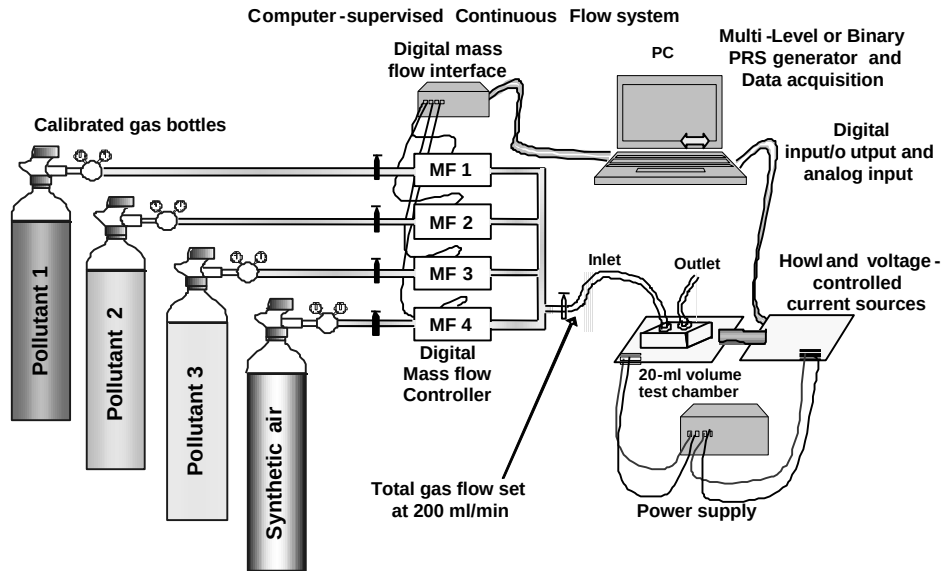


Figure 6: Experimental setup. The sensors are studied in the presence of either single or binary mixtures of three pollutant gases diluted at different concentrations in dry air.

The sensors were encapsulated and placed in a chamber where measurements were performed by means of a continuous flow system which allows us to obtain, starting from calibrated gas bottles, the desired concentrations of the different gases and gas mixtures in a highly reproducible way. The carrier was dry and the total gas flow was set to 200 ml/min (sccm) (10 % R.H. measured at $30\text{ }^{\circ}\text{C} \pm 1\text{ }^{\circ}\text{C}$) during the whole measurement process. Figure 6 shows the measurement system lay-out.

3.3.1 Micro-hotplate gas sensor fabrication

The sensor substrates consisted of four-element integrated micro-hotplate arrays fabricated using microsystems technology. Basically, the structure of any sensor within the micro-array comprised a gas-sensitive layer, interdigitated electrodes, insulating layers and a polysilicon heater, all stacked on a membrane. Figure 7 shows the architecture of a sensor micro-array encapsulated onto a TO-8 package (a) and the schematic view of a SnO_2 MHP gas sensor (b). Details of the fabrication process can be found in (Horrillo 1999; Rickerby et al. 2000; Gràcia 2001), (Llobet et al. 2003), (Ivanov et al. 2005), (Ivanov et al 2005), (Vicenzi et al. 2000), (Vicenzi et al. 2000), (Llobet et. al. 2003) and references therein.

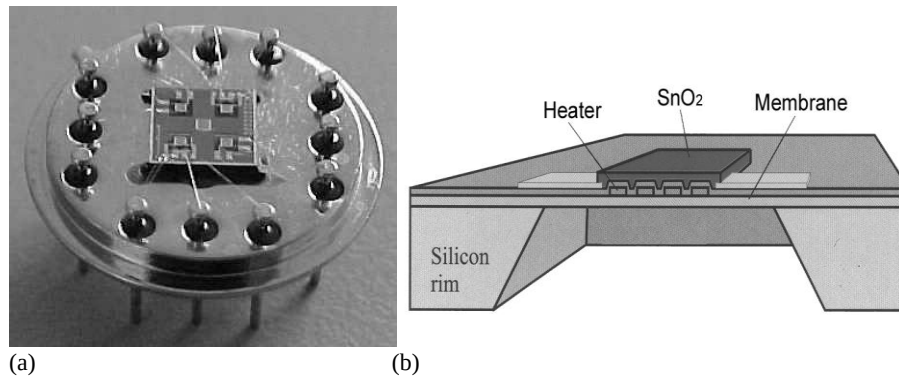


Figure 7: (a) 4-element micro-hotplate gas sensor array mounted on a standard TO-8 package and (b) structure of a micro-hotplate gas sensor.

Two different experiments were performed here. A preliminary experiment was conducted to determine the frequency range of the thermal modulation to be considered in our studies. This was done by analysing the thermal response of the

sensor membranes. Furthermore, the experiment is aimed at systematically studying the modulating frequencies of micro-hotplate gas sensors using PRBS. The discrimination among pollutant gases such as NO₂, NH₃ and their binary mixtures was envisaged. A second experiment deals with the optimisation of the modulating frequencies using this time MLPRS. Once the optimal modulating frequencies are identified and selected, a multi-sinusoidal modulating signal can be synthesized and employed to modulate the sensor's operating temperature and then validate the optimisation process.

3.3.2 Experiment 1: PRBS signals to modulate the temperature operation of micro-hotplates gas sensors.

The objectives of this experiment are twofold: On the one hand (preliminary experiment), the correct choice of the clock frequency and length of the sequence, essentials to study the appropriate frequency range with enough resolution. To determine the frequency range of the thermal modulation, the thermal response of the sensor membranes was studied. On the other hand, once the frequency range has been determined, to see whether it is possible, by studying the gas-sensor system in a systematic way using PRBS, to determine the most optimal frequencies to modulate the working temperature of the micro-hotplate gas sensors. The set-up of this experiment was then divided into the following steps:

A stepwise current signal was applied to the heating element of either thin or thick film gas sensors. It was found that all membranes behaved as first-order systems, where the ones coated by screen-printing (thick film) presented a thermal response of around 19 ms while the thermal response of the membranes coated by sputtering (thin film) was of about 13 ms. Therefore, the cut-off frequency of the coated membranes was around 52 and 77 Hz respectively. Therefrom, the clock frequency used to generate the MLS signal (f_c), was set to 250 Hz. This allowed the sensors to be tested for modulating frequencies ranging from D.C. up to 112.5 Hz (i.e., $0.45 \times f_c$). Once the f_c had been determined the length of the PRS signal can be set (e.g., 65,535) to get a spectral resolution (f_c/L) close to 4 mHz. Figure 8 shows the response of the heating element of a thin and thick film gas sensor (left to right) respectively to a stepwise voltage, where the thermal response change can be seen to be in the millisecond range. From this preliminary study, it was derived that the important modulating frequencies to

discriminate and quantify gases lay in the range between 0 and 1 Hz. This result is not surprising, since the objective of the temperature modulation is to alter the kinetics of diffusion and reaction processes at the sensor surface and these kinetics are slow (Moseley et al. 1987). This explains why low-frequency temperature-modulating signals (i.e. in the mHz range) have been used with micro-hotplate gas sensors (Cavicchi et al. 1995 and 1996; Heiling et al. 1997, Llobet et al. 2001 and 2002).

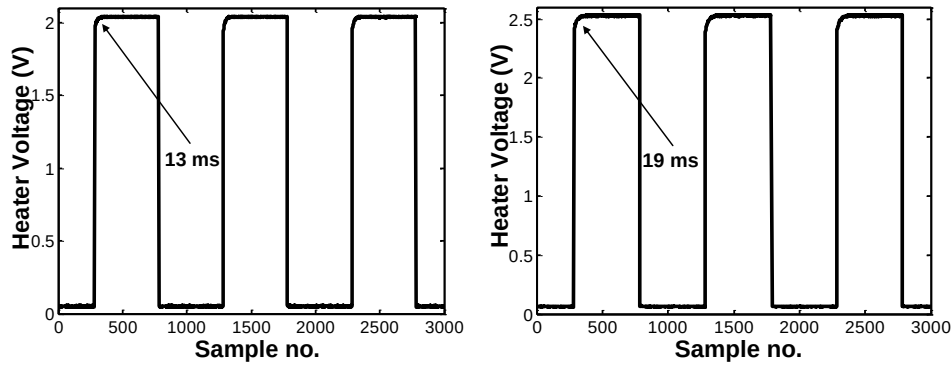
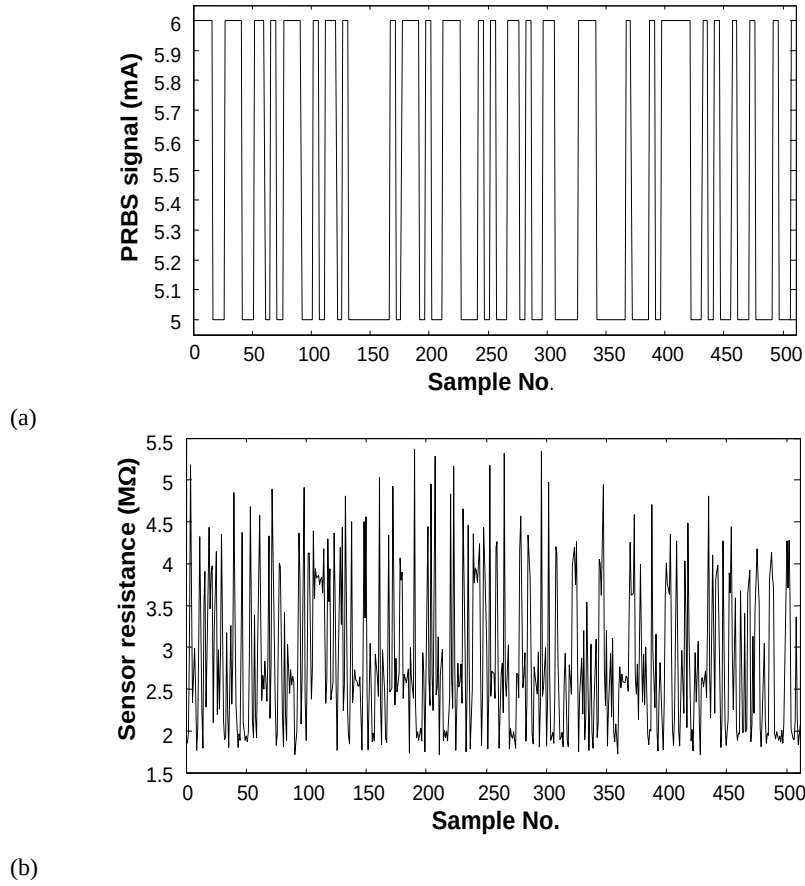


Figure 8: Response to a heater voltage stepwise applied to a micro-hotplate based gas sensor (thin and thick film (left to right)).

According to these results, the value of f_c and the length of the sequence and the resolution were readapted before the second part of the experiment was performed. Four tungsten oxide (thick film) micro-gas sensors were used to measure vapours at three different concentrations of NO_2 (0.5, 1, 2 ppm), NH_3 (100, 500, 1000 ppm) and their binary mixtures (0.5 + 100, 1 + 500, 2 + 1000 ppm of NO_2 and NH_3 , respectively) diluted in dry air (Vergara et al. 2005 (a)). Initially, measurements were performed while the operating temperature was kept constant (i.e., study of the steady-state response). In the second step, these species were measured while modulating the working temperature of the micro-hotplates. Each measurement was replicated 5 times. Every measurement took approximately 10 minutes to complete, which corresponded to the acquisition of 2.5 sequences of length 511 generated with a clock frequency of 2 Hz. The sampling frequency was set to 10 Hz. Figure 9 shows the PRBS signal used to modulate sensor temperature and a typical transient response of a micro-hotplate gas sensor in the presence of 2 ppm of NO_2 .



(b)
Figure 9: (a) Temperature-modulating PRBS sequence of length 511 applied to the sensor heating element and (b) resulting resistance transient of a tungsten-oxide micro-hotplate sensor in the presence of 2 ppm of NO_2 .

3.3.3 Experiment 2: Optimized Temperature Modulation of Metal Oxide Micro-Hotplate Gas Sensors through Multilevel Pseudo Random Sequences.

In the former experiment, we aimed at systematically selecting the modulation frequencies of micro-hotplate gas-sensors using pseudo-random binary sequences of maximum length (PRBS). In this experiment, an evolved method to modulate

the working temperature of metal oxide micro-hotplate gas sensors in a wide frequency range is presented, but this time based on the use of maximum length multilevel pseudo random sequences (MLPRS). One of the main reasons for considering multilevel signals instead of the binary ones is that the former can provide a better estimate than the latter of the linear dynamics of a process with non-linearities (Vergara et. al. 2007). And it is well known that temperature-modulated metal oxide gas sensors present non-linearity in their response.

Using this strategy, it is shown that the best temperature modulating frequencies to discriminate and quantify gases using an array of 4 metal oxide gas sensors are identified. The process is illustrated in solving a practical application: the quantitative analysis of acetaldehyde, ethylene, ammonia and their binary mixtures.

Initially (i.e., during the frequency optimization process) the response of the first micro-array was measured when the sensors' operating temperature was modulated using a 5-level PRS. The higher and lower current levels were set to 8 mA and 7 mA, respectively, which resulted in a maximum temperature variation of the micro-machined membranes equal to 200 °C. Since the number of levels of the pseudo random signal was set to 5, the current levels were 7, 7.25, 7.5, 7.75 and 8 mA. The clock frequency (f_c) used to generate the MLPRS signal was set at 2 Hz. This frequency allows us to test the gas/sensors system at temperature modulating frequencies that range from 0 up to near 1 Hz. These modulating frequencies are far below the cut-off frequency of the metal oxide coated membranes, which is the inverse of their thermal response, (i.e., $1/19 \times 10^{-3} \text{ s}^{-1} = 52 \text{ Hz}$). The length of the MLPRS signal was set to 624, which resulted in a spectral resolution, f_c/L , of 3.21 mHz (see Figure 10 (a)). During each measurement, two periods of the excitatory MLPRS signal and the corresponding sensor response signals (i.e. sensor resistance) were acquired and stored for further processing.

In the second step (i.e., during the validation process) the response of the second micro-array was measured when the sensors' operating temperature was modulated using a multi-sinusoidal signal resulting from the sum of a few sinusoids of identical amplitude. The frequencies of these signals were a subset of those that had been identified in the first step as more advantageous for gas recognition and quantification.

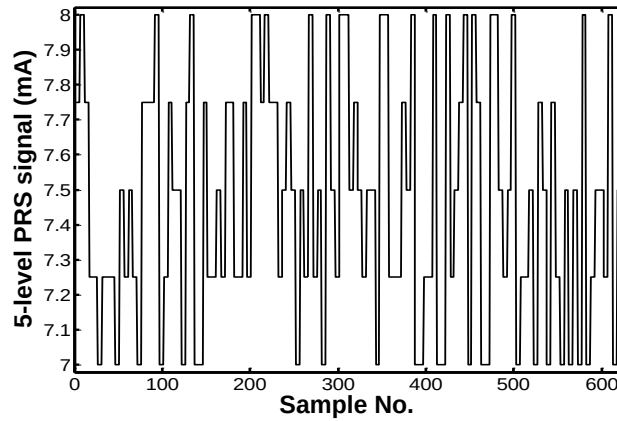
Databases: The first database was used for optimizing the temperature modulation of the sensors via a 5-level MLPRS. The following single gases and gas mixtures were measured using the first sensor micro-array:

- (1) Acetaldehyde (10; 50 and 100 ppm)
- (2) Ethylene (10; 50 and 100 ppm)
- (3) Ammonia (25; 50 and 75 ppm)
- (4) Acetaldehyde + Ethylene (10 + 10; 10 + 50; 10 + 100; 50 + 10; 50 + 50; 50 + 100; 100 + 10; 100 + 50; 100 + 100 ppm)
- (5) Ethylene + Ammonia (10 + 25; 10 + 50; 10 + 75; 50 + 25; 50 + 50; 50 + 75; 100 + 25; 100 + 50; 100 + 75 ppm)
- (6) Acetaldehyde + Ammonia (10 + 25; 10 + 50; 10 + 75; 50 + 25; 50 + 50; 50 + 75 ppm)

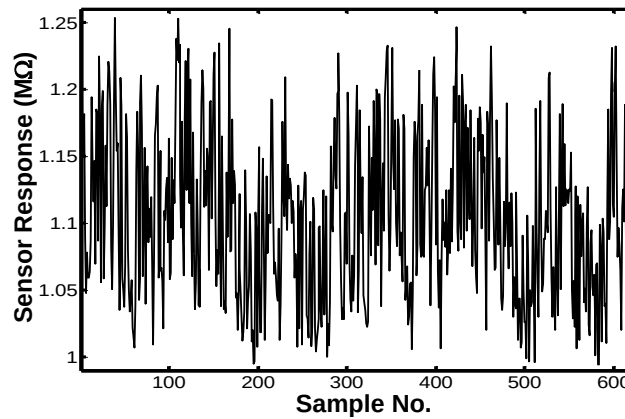
Since each measurement was replicated 5 times (performed in a disordered way), this gave a total of 165 measurements in the first database. The measurement procedure consisted of the following steps: First, measurements were performed while the operating temperature was kept constant of 450 °C. This was done to measure the steady-state sensor response. Then the sensors' operating temperature was modulated using the MLPRS signal.

After the sensors had been thermally modulated during five cycles of the MLPRS, the acquisition of 2 additional cycles started. This was done to ensure that reproducible response patterns were acquired during each measurement. The acquisition of these two cycles took 10.4 minutes to complete (i.e. 2 sequences of length 624 generated with a clock frequency of 2 Hz). The sampling frequency was set to 10 Hz. An example of the 5-level PRS signal used to modulate the temperature of the sensors and a typical transient response of a Au-doped WO₃ micro-hotplate gas sensor in the presence of Acetaldehyde 50 ppm are shown in Figure 10 (top to bottom respectively).

The second database was used for validating the frequency selection performed using the first database. A new set of measurements was taken using the second micro-sensor array. The gases and concentrations measured were the ones reported for the first database. This time, a multi-sinusoidal signal was used to modulate the sensors' operating temperature. An equivalent procedure to the one reported before was employed to perform the second databases.



(a)



(b)

Figure 10: (a) 5-level PRS applied to the sensors' heating element and (b) resulting resistance of a screen-printed WO_3 micro-hotplate gas sensor in the presence of 50 ppm of acetaldehyde.

3.4 OPTIMISATION USING PSEUDORANDOM SEQUENCES

3.4.1 Initial optimisation using pseudorandom binary sequences

In this section the data from experiment 1 is fully analyzed. The sensor used in the experiment was a tungsten oxide micro-hotplate gas sensor and vapours of

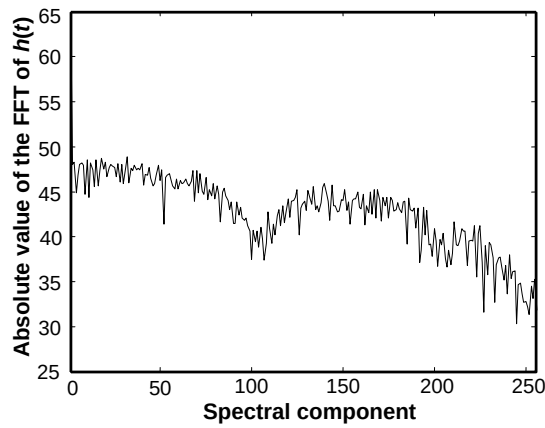
three different concentration of NO_2 , NH_3 and their binary mixtures diluted in synthetic dry air were the gases measured (Vergara et. al. 2005 (a)). As was mentioned earlier, the main aim of the first part of this experiment (preliminary study) consisted of determining the clock frequency f_c and length of the sequence L in order to study an appropriate modulation frequency range with enough resolution. To determine the thermal response of the sensor membranes a stepwise current signal was applied to the heating element of the gas sensors (i.e., sensor membranes were coated using either thin or thick film technology). Three replicate measurements were made for each one of the 4 different current impulses tested. Since the initial study revealed that the important temperature modulating frequencies lay in the range 0 to 1 Hz, the clock frequency (f_c) used to generate the PRBS signal was set to 2 Hz. This frequency allows the sensors to be tested at temperature-modulating frequencies that range from 0 up to near 1 Hz. The length of the sequence was set to 511, which resulted in a spectral resolution of 3.91 mHz. Once these parameters had been determined the PRBS was built in order to study, in the pre-established frequency range, the modulating frequencies. The process to obtain the optimal modulation frequencies for a given sensor and target gases needs two steps:

- (1) In the first step, an estimate of the impulse response of the sensor in the presence of each gas or mixture is computed. An estimate of the impulse response is obtained for every measurement and the spectral components of the impulse response are computed via the fast Fourier transform (FFT) (this step is described above).
- (2) In the second step, it is necessary to select the subset of spectral components that carry important information to discriminate and/or quantify the target gases. This subset of spectral components is the set of optimal modulation frequencies (this step has been described above).

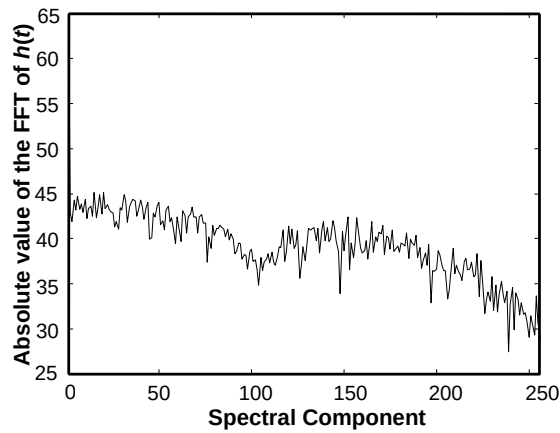
3.4.1.1 Estimating the impulse response.

The database described in experiment 1 (consisted of 45 measurements) was used for this purpose. An estimate of the impulse response, $\hat{h}(n)$, of each sensor in the presence of the gases and gas mixtures studied was computed through the circular cross-correlation between one period ($L = 511$ samples) of the PRBS signal, $x(n)$, and one period of the sensor response (i.e. resistance transient), $y(n)$.

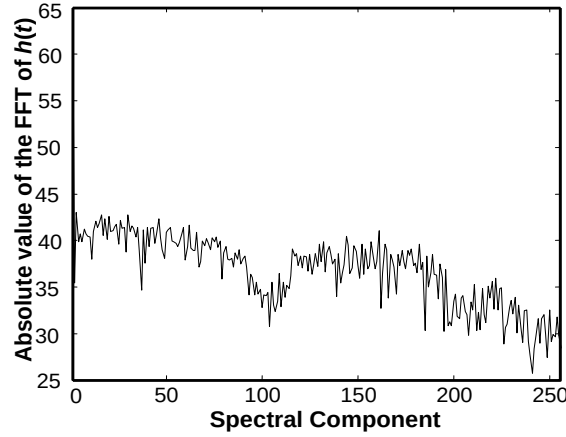
Since the clock frequency of the PRBS sequence was set to $f_c = 2$ Hz and the sampling rate of the acquisition system was set to 10 Hz, the sequences had to be decimated by 5 before the cross-correlation was computed. The module of the FFT of the impulse response estimate was calculated and, stored for further processing. This was done to study which spectral components contain important information for the identification and quantification of gases. Figure 11 shows the FFT module of the impulse responses for a tungsten oxide micro-hotplate sensor in the presence of ammonia, nitrogen dioxide and an ammonia and nitrogen dioxide mixture. This figure also shows that differences exist between these spectra when the sensor is in the presence of different gases.



(a)



(b)



(c)

Figure 11: Module of the FFT of the impulse response estimate of a tungsten oxide micro-hotplate sensor in: a) NH_3 (1000 ppm), b) NO_2 (2 ppm) and c) NH_3 (1000 ppm) + NO_2 (2 ppm).

In the next step, a variable selection method was implemented to identify those spectral components that carry important information for the discrimination and quantification of the pollutant gases studied. The number of spectral components available is 255, which corresponds to half the length of the PRBS ($L/2$). Two consecutive spectral components are separated by the spectral resolution, f_c/L (i.e. $2/511 \text{ Hz} = 3.91 \text{ mHz}$). For gas identification purposes, the measurements were grouped into 3 categories (i.e., ammonia, nitrogen dioxide and gas mixture). For quantification purposes, three specific models were built (one per species or mixture). Therefore, for every quantification model, the measurements were grouped into 3 categories (e.g. gas or gas mixture $\times$ 3 concentrations). For each spectral component, intra-group and between-group variances were computed (see Eq.(17) and (18)). A figure of merit for the resolution power, (RP), of a spectral component (i.e, the ratio between the inter-category variance and the intra-category variance) was calculated (see Eq.(19)).

Therefore, a small set of spectral components (i.e., those that have the higher figure of merit) were selected. These spectral components correspond to the

temperature-modulating frequencies that lead to a better discrimination or quantification of the gases studied (see Eq.(20)).

3.4.1.2 Qualitative and quantitative gas analysis.

One matrix per sensor was formed, which contained as number of rows the number of measurements performed (i.e. 45) and as number of columns the number of selected spectral components. Therefore, every element in these matrices corresponded to the value of a selected spectral component of the impulse response estimate for a given gas-sensor pair. These matrices were used to build and validate fuzzy ARTMAP classifiers to see whether it was possible to correctly identify the gases. A brief description of the ARTMAP network is shown as follows.

The ARTMAP comprises two ART modules (ART_a and ART_b) linked by an inter-ART associative memory referred to as ‘match tracking system’. Each ART module is formed by two major subsystems the attentional subsystem and the orienting subsystem. Two interconnected layers of neurones F1 and F2, which are fully connected both bottom-up and top-down, comprise the attentional subsystem. The application of a single input vector leads to patterns of neural activity in both layers F1 and F2. The activity in F2 nodes reinforces the activity in F1 nodes due to top-down connections. The interchange of bottom-up and top-down information leads to a resonance in neural activity. The orienting subsystem is responsible for generating a reset signal to F2 when the bottom-up input pattern and top-down template pattern mismatch at F1, according to a vigilance criterion. When a mismatch is detected, the network adapts its structure by immediately storing the novelty in additional weights. The vigilance criterion is set by the value of the vigilance parameter. A high value of the vigilance parameter means that only a slight mismatch will be tolerated before a reset signal is emitted. On the other hand, a small value (low vigilance) means that large mismatches will be tolerated. After the resonance check, if a pattern match is detected according to the vigilance parameter, the network changes the weights of the winning node. During supervised learning ART_a receives a stream of input patterns $\{a^M\}$ and ART_b also receives a stream of patterns $\{b^M\}$, where b^M is the correct prediction given a^M . When a prediction by ART_a is not confirmed by ART_b , inhibition of the inter-ART associative memory activates a match tracking process. This increases ART_a vigilance by the minimum amount needed for the system either to activate an ART_a category that matches the ART_b category or to

learn a new ART_a category. Fuzzy ARTMAP carries out supervised learning like back-propagation. But unlike back-propagation, Fuzzy ARTMAP is self-organising, self-stabilising and suitable for real-time learning. More details about the building and validation of fuzzy ARTMAP classifiers can be found in (Carpenter et al. 1992; Ionescu et al 2002) and references therein.

A cross-validation of order one (i.e. a leave-one-out cross-validation) was implemented to estimate the accuracy classification rates. Through the selection of information from up to 4 modulating frequencies, the gas identification rate using a single sensor varied between 80 and 93.33%. For instance, up to a 93.33% identification rate was reached with sensor 3 when the modulation frequencies of 168, 399, 743 and 797 mHz where used. While NH₃ samples could be perfectly discriminated, confusions occurred between samples of NO₂ and NO₂ + NH₃ mixtures. Increasing the number of modulating frequencies used did not improve these results. The gas identification rate of the fuzzy ARTMAP classifier increased up to 95.56% when information from different sensors was combined. For example, when sensors 1, 3 and 4 were combined, the modulating frequencies selected were 168, 188, 328, 399, 743, 797, 919 and 938 mHz. These results are summarised in Tables 3 and 4.

Table 3: Frequencies selected (with high figure of merit for gas identification) for the different sensors studied and success rate in gas identification using a fuzzy ARTMAP classifier.

Sensor no.	1	2	3	4
Frequencies selected (Hz)	0.919 0.938	0.188 0.403	0.168 0.399 0.743 0.797	0.188 0.328
NH ₃ identification (%)	100	100	100	100
NO ₂ identification (%)	86.67	80	93.33	73.33
NH ₃ + NO ₂ identification (%)	60	73.33	86.66	66.66
Overall success rate (%)	82.23	84.45	93.33	80

In the second step the selection of the optimal set of temperature-modulating frequencies to quantify the different gases and gas mixtures was envisaged. By an equivalent procedure to the one described above, for every micro-sensor, the spectral components that had a high figure of merit for solving the quantification problem were selected to form the data matrices. Since specific quantification

models were built (i.e. for NH_3 , NO_2 and their mixtures), for each micro-sensor 3 data matrices were formed. Once again, fuzzy ARTMAP classifiers were built and cross-validated (using the leave-one-out approach) to estimate the success rate in gas quantification. Results are summarised in table 5 which shows that the concentration of the different gases and mixtures can be identified with an accuracy of a 100% even using a single sensor.

Table 4: Success rate in gas identification when the information from different sensors is combined using a fuzzy ARTMAP classifier.

Sensor no.	1-3-4	3-4	1-4	2-4
Identification success rate (%)	95.56	93.34	91.12	91.12

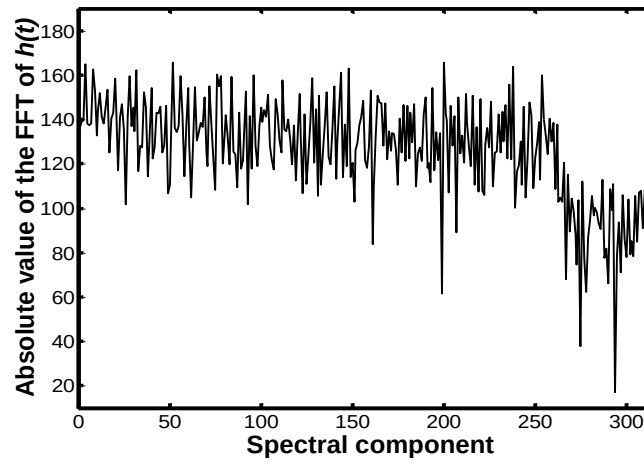
Table 5: Frequencies selected (with high figure of merit for gas quantification) for the different sensors studied and success rate in gas quantification using a fuzzy ARTMAP classifier.

Gases		1	2	3	4
NH_3	Frequencies selected (Hz)	0.950	0.070 0.188 0.234 0.789 0.950 0.993	0.188	0.789 0.950
	Quantification success rate (%)	100	100	87	100
	Frequencies selected (Hz)	0.445	0.797	0.797	0.797
	Quantification success rate (%)	93.33	100	93.33	100
	$\text{NH}_3 + \text{NO}_2$ Frequencies selected (Hz)	0.845	0.446 0.594 0.797 0.845	0.489 0.685	0.063 0.410 0.716 0.845
	Quantification success rate (%)	100	100	100	100

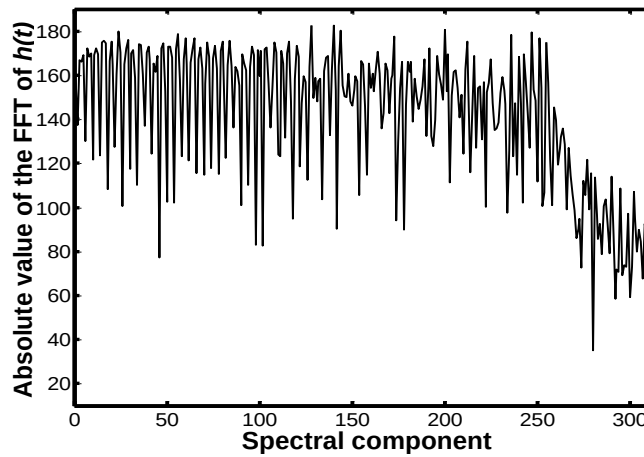
3.4.2 Optimisation and validation using multi-level sequences (MLPRS)

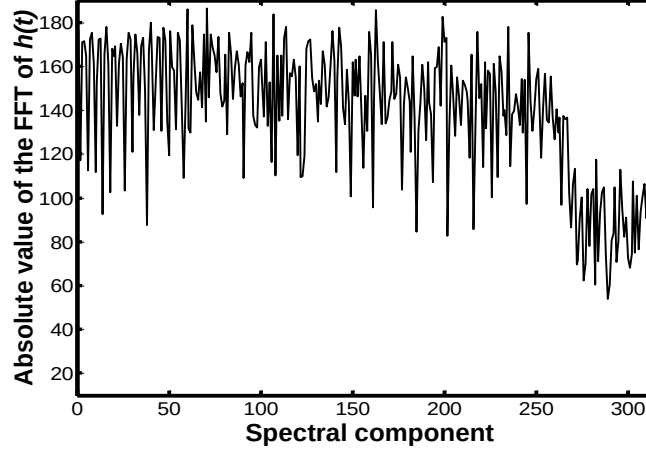
In the previous experiment, we introduced a method based on the use of MLPRS to modulate the working temperature of gas sensors. In this section an evolved method based on the use of MLPRS sequences for optimizing the choice of the operating temperature modulating frequencies of metal oxide micro-hotplate gas sensors is developed and presented. The problem envisaged to illustrate the process is the building of calibration models for the analysis of acetaldehyde, ethylene, ammonia and their binary mixtures, species related to the quality of

climacteric fruit during cold storage and the occurrence of a leak in the refrigeration system respectively. In an attempt to show the consistency of the method discussed, all the optimisation process was conducted using a sensor array and all the validation was performed using a different sensor array from the same fabrication batch.



(a)





(c)

Figure 12: Absolute value of the FFT for a temperature modulated, WO_3 micro-hotplate sensor in the presence of (a) 50 ppm acetaldehyde, (b) 50 ppm ethylene, and (c) 50 ppm of ammonia.

The first database described in section 3.3 (consisting of 165 measurements) was used for the spectral analysis purpose. For each measurement, an estimate of the impulse response, $\hat{h}(n)$, was computed by means of the circular cross-correlation between one period of the input MLPRS signal, $x(n)$, and one period of the sensor response signal (i.e. resistance transient), $y(n)$. The absolute value of the FFT of the impulse response estimate was calculated. This was done to study which spectral components contain important information for the identification and quantification of gases. Figure 12 shows the absolute value of the FFT of the impulse response estimates for an Au-doped WO_3 micro-hotplate in the presence of acetaldehyde, ethylene and ammonia (the concentration of these species was 50 ppm).

The number of spectral components available is 312, which corresponds to half the length of the MLPRS ($L/2$). Two consecutive spectral components are separated by the spectral resolution, f_c/L (i.e. $2/624 \text{ Hz} = 3.20 \text{ mHz}$).

Once again, a simple search method was implemented to find a reduced set of spectral components that better helped either in the discrimination of the gases and gas mixtures or in gas quantification. For discrimination purposes, measurements were grouped in 6 classes, three corresponding to the single species and three for gas mixtures. On the other hand for quantification model,

the measurements were grouped in 3 categories (i.e., 3 concentrations). Once again for each spectral component, intra-category and between-category variances were computed, and then the parameter defined to rank the discriminating power of a given spectral component was computed (see Eq.19).

3.4.2.1 Optimization for qualitative gas analysis.

The optimization was conducted on the first micro-sensor array. The number of measurements available in this database was 165 (5 replicate measurements for each gas, gas mixture and concentration). The objective here was to identify an optimal set of temperature-modulating frequencies to discriminate among the gases and gas mixtures, regardless their concentration.

The process of frequency selection was performed using a 5-fold validation approach. Selection the datasets was performed by using 4 out of the 5 replicates available per measurement (there were 132 measurements in each of the 5 selection datasets). Their corresponding validation datasets were formed by the replicate left out (there were 33 measurements in each of the 5 corresponding validation datasets). Therefore, each sensor within the micro-array had 5 frequency selection matrices (132 rows or measurements $\times$ 312 columns or spectral components) and 5 validation matrices (33 rows $\times$ 312 columns). The variable selection process described above was used with each frequency selection matrix in order to rank spectral components according to their usefulness to discriminate among gases and gas mixtures (6-category classification).

For each sensor and selection set, the best five spectral components were selected. Table 6 shows the frequencies corresponding to these components. Two conclusions can be derived from the frequencies selected:

- (1) Considering each micro-sensor separately: A very high consistency exists among the frequencies selected over the five selection folds. Sensor 1 shows a higher variability than sensors 2, 3 and 4. Furthermore, differences arise generally in the fourth and fifth frequencies (i.e. those among the frequencies selected that had the lower value of the parameter used to rank them).
- (2) Considering the 4 sensors studied together: There are a high number of coincidences in the values of the modulating frequencies selected. This is not surprising, since 3 out of 4 sensors are based on the same gas-sensitive material (tin oxide), even though in each sensor, a different noble metal

catalyst was used.

Table 6: Results of the 5-Fold selection and validation process for gas identification. A fuzzy ARTMAP classifier and a leave-one-out cross-validation were used to estimate performance.

Fold # ↓	Sensors→	1	2	3	4	2, 3	2, 3, 4
1		96.2	12.8	12.8	12.8		
	Frequencies selected (mHz)	121.8	25.6	25.6	38.4		
		134.6	76.9	38.4	198.7	-	-
		144.2	102.6	64.1	464.7		
		198.7	115.4	76.9	618.6		
	Identification success rate (%)	42.4	63.0	61.8	51.5	84.2	89.7
2		12.8	12.8	12.8	12.8		
	Frequencies selected (mHz)	25.6	25.6	25.6	25.6		
		38.4	115.4	38.4	38.4	-	-
		185.9	153.8	64.1	185.9		
		198.7	192.3	76.9	198.7		
	Identification success rate (%)	42.4	55.7	61.2	47.3	80.6	84.3
3		3.2	12.8	12.8	12.8		
	Frequencies selected (mHz)	12.8	25.6	25.6	25.6		
		25.6	76.9	38.4	38.4	-	-
		38.4	89.7	64.1	134.6		
		134.6	102.6	76.9	618.6		
	Identification success rate (%)	52.7	63.6	61.2	51.5	80.6	85.5
4		12.8	12.8	12.8	12.8		
	Frequencies selected (mHz)	28.8	25.6	25.6	25.6		
		121.8	76.9	38.4	38.4	-	-
		134.6	102.6	64.1	198.7		
		198.7	115.4	76.9	451.9		
	Identification success rate (%)	50.9	73.3	60.0	55.1	83.0	89.7
5		3.2	12.8	12.8	12.8		
	Frequencies selected (mHz)	22.4	25.6	25.6	25.6		
		25.6	76.9	38.4	38.4	-	-
		28.8	102.6	76.9	134.6		
		134.6	192.3	89.7	618.6		
	Identification success rate (%)	52.7	64.8	57.5	55.1	81.2	87.3

Fuzzy ARTMAP classifiers were built and validated to see whether it was possible to correctly identify the different gases and gas mixtures using the spectral components that had been selected. This process was repeated for the 5

folds available. Leave-one-out cross-validation was the approach implemented to estimate the success rate in identification. Each fold comprised 165 measurements, (132 had been used in the frequency selection step and 33 had been left out). Fuzzy ARTMAP classifiers were trained 165 times using 164 training vectors. The vector left out during the training phase (i.e. unseen by the classifier) was then used for testing. The performance of a given classifier was estimated as the averaged performance over the 165 tests.

Classifiers were built and validated using the information of 1 sensor, 2 sensors (2 and 3) and 3 sensors (2, 3 and 4). Therefore, the dimension of the vectors input to the fuzzy ARTMAP was 5, 10 and 15, respectively, since the values of 5 spectral components were used per sensor. The cross-validation results are shown in Table 6. This table shows that combining the information from sensors 2, 3 and 4 leads to success rates in gas identification that range between 84.3% and 89.7%, depending on the fold considered. The average of the success rate over the 5 folds is 87.3%. Increasing beyond 5 the number of spectral components used per sensor did not improve these results.

Table 7: Confusion matrix (leave-one-out cross-validation results) for the fuzzy ARTMAP classifiers applied to the identification of gases and gas mixtures (adding up the results of the 5 different folds). Bold face is used for correct identifications, plain face is used for misclassified samples from the frequency selection datasets and italics is used for misclassified samples from the validation datasets.

		Real category					
		1	2	3	4	5	6
Predicted as	1	72		1	2	1	
	2	1	75				
	3			74			
	4				189	23+10	17+11
	5	2			26+4	191	2
	6				4		120

1: Acetaldehyde, 2: Ethylene, 3: Ammonia, 4: 1&2, 5: 2&3, 6: 1&3.

Table 7 shows the confusion matrix that integrates the identification results for the 5 folds when the information from sensors 2, 3 and 4 was used. A clear distinction is made between measurements that had participated in the frequency selection process and those that had not. Table 8 shows that most of the samples misclassified belong to gas mixtures. The success rate in gas identification (averaged for the 5 folds), estimated using measurements that took part in the

selection of spectral components exclusively, is 89.4%. On the other hand, the success rate estimated using measurements that were left out (i.e. were not used in the selection of spectral components) is 80%. The latter success rate is similar to the former one (it decreases, but not substantially). This proves that the variable selection process is not over-trained and that it leads to the selection of those modulating frequencies that help discriminating among the species considered.

3.4.2.2 Optimization for quantitative gas analysis.

In the second step the selection of the optimal set of temperature-modulating frequencies to quantify the different gases and gas mixtures was envisaged. Once again, this optimization was conducted on the first micro-sensor array. Since specific quantification models were sought for every gas or gas mixture, the database (165 measurements, i.e. 5 replicate measurements for each gas, gas mixture and concentration) was split in 6 databases. The first 3 databases were for single gases (15 measurements in each) and the second 3 databases were for gas mixtures (45, 45 and 30 measurements for Acetaldehyde + Ethylene, Ethylene + Ammonia and Acetaldehyde + Ammonia, respectively). The process of frequency selection was performed using a 5-fold validation approach. Selection datasets were formed by using 4 out of the 5 replicates available per measurement. There were 12 measurements in the selection datasets of single gases and 36, 36 and 24 measurements in the selection datasets of gas mixtures. Their corresponding validation datasets were formed by the replicates left out.

The variable selection process was, once again, used with each frequency selection matrix in order to rank spectral components according to their usefulness to quantify the gases. Only sensors 2, 3 and 4 were studied (since sensor 1 was discarded in the qualitative analysis). For each sensor and selection set, the best five spectral components were selected. Table 8 summarizes the results of the selection process. The frequencies shown in Table 8 correspond to those that appeared selected more times over the five selection folds.

PLS calibration models that used the values of the spectral components selected from sensors 2, 3 and 4 were built and validated. Data were mean centered. PLS is a linear and supervised multivariate calibration method that attempts to find factors, which capture as much variance as possible in the predictor block X -matrix (spectral components for sensors 2, 3 and 4), under the

constraint of being correlated with the predicted block Y-matrix (gas or gas mixture concentrations) (Geladi et al. 1986).

Table 8: Frequencies selected to build specific quantitative models. The number of times a frequency is selected over the 5 folds is shown in parenthesis.

Gases/mixture models↓	Sensors→	2	3	4
Acetaldehyde	Frequencies selected (mHz)	339.7 (3)	6.4 (3)	339.7 (2)
		496.8 (4)	224.4 (5)	676.3 (3)
		612.2 (5)	339.7 (5)	698.7 (5)
		804.5 (5)	785.3 (4)	804.5 (5)
		996.8 (2)	887.8 (3)	887.8 (4)
Ethylene	Frequencies selected (mHz)	12.8 (2)	22.4 (4)	41.7 (3)
		51.3 (5)	25.6 (5)	179.5 (2)
		60.9 (4)	28.8 (5)	195.5 (3)
		73.7 (5)	67.3 (5)	208.3 (4)
		734.0 (2)	734.0 (2)	339.7 (3)
Ammonia	Frequencies selected (mHz)	256.4 (4)	450.1 (4)	185.9 (3)
		500.0 (4)	500.0 (4)	250.0 (3)
		564.1 (3)	680.0 (5)	330.1 (3)
		698.7 (3)	685.9 (3)	403.8 (4)
		705.1 (4)	894.2 (3)	685.9 (5)
Ethylene + Acetaldehyde	Frequencies selected (mHz)	12.8 (4)	3.2 (4)	3.2 (5)
		25.6 (4)	378.2 (3)	9.6 (5)
		128.2 (3)	496.8 (5)	12.8 (5)
		551.3 (3)	641.0 (5)	16.02 (5)
		910.2 (4)	846.1 (3)	38.5 (3)
Ethylene + Ammonia	Frequencies selected (mHz)	51.3 (5)	38.5 (5)	92.9 (3)
		112.2 (2)	310.9 (2)	253.2 (5)
		121.8 (2)	419.9 (5)	368.6 (3)
		298.1 (5)	682.7 (3)	419.9 (5)
		689.1 (4)	717.9 (3)	987.2 (3)
Acetaldehyde + Ammonia	Frequencies selected (mHz)	92.9 (3)	272.4 (5)	317.3 (4)
		352.6 (3)	294.9 (2)	349.4 (2)
		419.9 (4)	342.9 (3)	419.9 (5)
		557.7 (3)	403.9 (2)	544.9 (4)
		695.5 (3)	846.2 (4)	689.1 (2)

The process used to build and validate the specific PLS calibration models was as follows. The number of latent variables (LV) to be used in each model was determined using the measurements in the selection datasets. Leave-one-out

cross-validations were performed with these measurements and the root mean square error of cross validation (RMSECV) versus the number of latent variables was computed, according to the definition:

$$RMSECV = \sqrt{\frac{\sum_{i=1}^n (Y_i - y_i)^2}{n}} \quad (21)$$

where Y_i is the actual concentration value and y_i the model prediction. The number of LV selected was the value after the first sharp decrease in RMSECV.

Once the number of LV had been determined, PLS models were built (one for each gas or gas mixture) using the measurements in the selection datasets. These calibration models were validated using the measurements that had been left out, i.e. those that belonged to the validation datasets. Since 5 different selection and validation sets were available (5-fold selection), this process was repeated 5 times. Table 9 summarizes the quantification results for each gas/mixture model. These results are exclusively for validation measurements. In addition to the RMSECV, the table also shows the slopes and the correlation coefficients of the linear regressions between actual and predicted concentrations. The closer to one are slopes and correlation coefficients, the better the calibration PLS models are.

Table 9: Validation results for the specific PLS calibration models (the selected spectral components from sensors 2, 3 and 4, were used as input data). Number of LV used, slope (m) and correlation coefficient (r) of the linear regression between real and predicted concentrations and root mean square error of cross-validation (RMSECV).

Gas/gas mixture models	Validation results			
	LV#	M	R	RMSECV
Acetaldehyde	6	0.75	0.71	28.8
Ethylene	4	0.87	0.94	12.9
Ammonia	6	0.97	0.98	4.8
Ethylene + Acetaldehyde	17	0.6	0.65	31.8
		0.91	0.92	14.3
Ethylene + Ammonia	18	0.82	0.87	18.8
		0.92	0.94	7.2
Acetaldehyde + Ammonia	18	0.98	0.99	2.9
		0.98	0.97	5.3

Table 9 shows that the concentration of ammonia (either as a single gas or in mixtures) can be predicted accurately. On the other hand, the estimation of the concentrations of acetaldehyde (as a single gas) and ethylene (in ethylene + acetaldehyde mixtures) is poor. Finally, the performance of the remaining calibration models is fair.

3.4.2.3 Validation process: Use of multi-sinusoidal signals

Once the processes for determining the optimal frequencies for gas identification and quantification have been conducted, the final validation can be envisaged. This consists of selecting a reduced set of optimal frequencies, synthesize a multi-sinusoidal temperature-modulating signal, run a new set of measurements and build and validate identification and calibration models.

The six frequencies selected to synthesize the multi-sinusoidal signal were taken from the optimization results for sensors 2, 3 and 4 (from experiment 2). These frequencies are shown in Table 10. The first three frequencies selected correspond to those that are more important for gas identification (i.e. were ranked with the highest score in the selection process). The last three frequencies are representative of the ones used for quantification. Considering the fact that different frequencies are useful to quantify the different gases or gas mixtures studied (see Table 8), three frequencies were selected, which had a high score and corresponded to medium, medium-high and high frequency values, respectively. The low frequency values found in Table 8 are already represented by the identification frequencies.

Table 10: Temperature modulation frequencies selected after the qualitative and quantitative gas analysis using MLPRS signals.

Final selection of frequencies (mHz)	12.8	25.6	38.5	92.9	339.7	682.7

The objectives of this study were the following:

- (1) To show a practical application of temperature modulated sensors after the modulation process has been optimized. In other words, MLPRS are used to determine the optimal temperature-modulating frequencies. Following this optimization, a much simpler multi-sinusoidal signal (using optimal

- frequencies) would be used in a practical application.
- (2) To validate the whole optimization process by performing a new set of measurements using a different micro-sensor array. This second micro-array belongs to the same fabrication batch as the array employed to perform the optimization.
 - (3) To assess the performance of the temperature-modulated micro-sensors in qualitative and quantitative gas analysis.

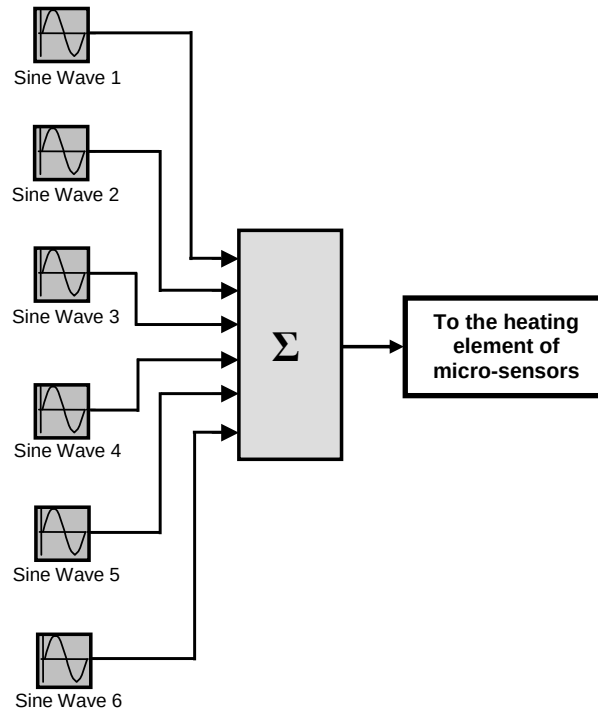
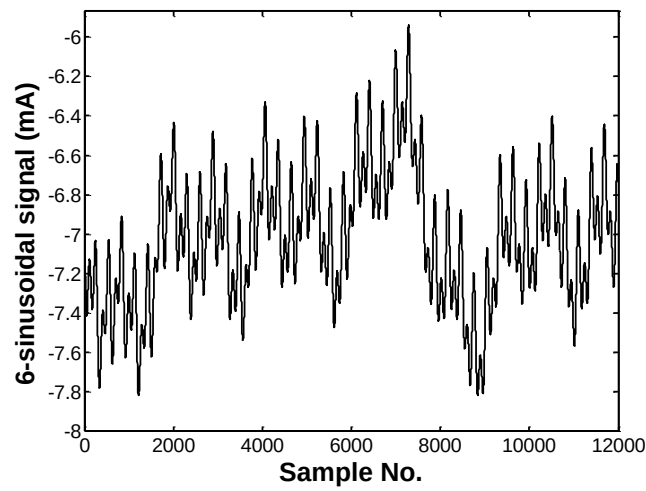


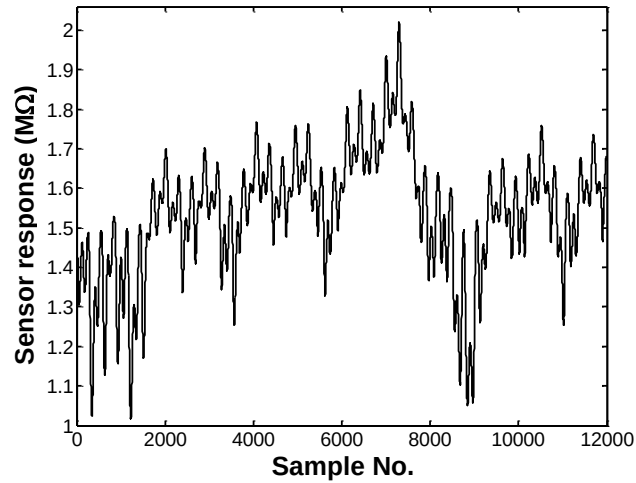
Figure: 13. Setup used to generate a multi-sinusoidal signal, which consists of the sum of 6 sinusoids of identical amplitudes and different frequencies. The signal is applied to the heating element of the micro-sensors studied.

The new measurement database consisted of 165 measurements (5 replicate measurements per gas, gas mixture and concentration) (see experiment section). The sensors had their operating temperature modulated by a signal resulting from the sum of 6 sinusoids of identical amplitude, the frequencies of which are those reported in Table 11. The amplitude of the resulting multi-sinusoidal signal (a

current signal) was equal to 2 mA. The procedure for generating this signal is illustrated in Figure 13. Figure 14 shows a segment of the multi-sinusoidal signal applied to the heater of the second micro-sensor array (a) and a typical sensor response (b).



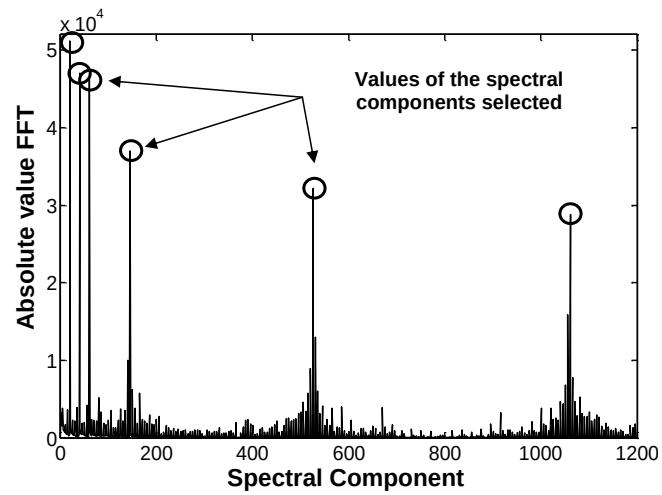
(a)



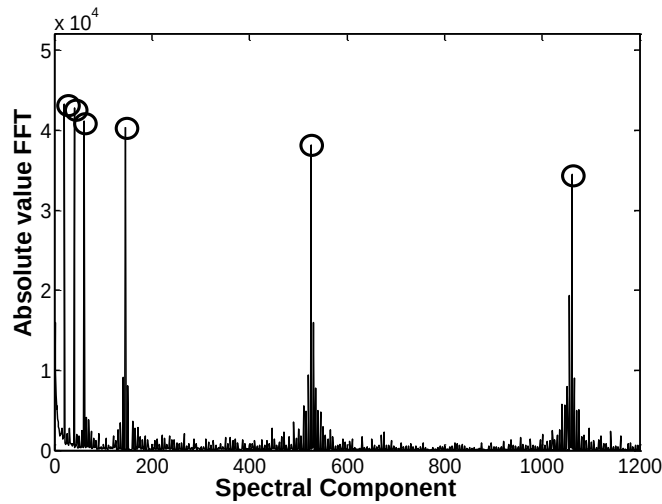
(b)

Figure 14: (a) Fragment of the multi-sinusoidal signal applied to the heating element of the micro-hotplate sensors. (b) Response of a temperature-modulated WO_3 micro-hotplate in the presence of acetaldehyde 50 ppm.

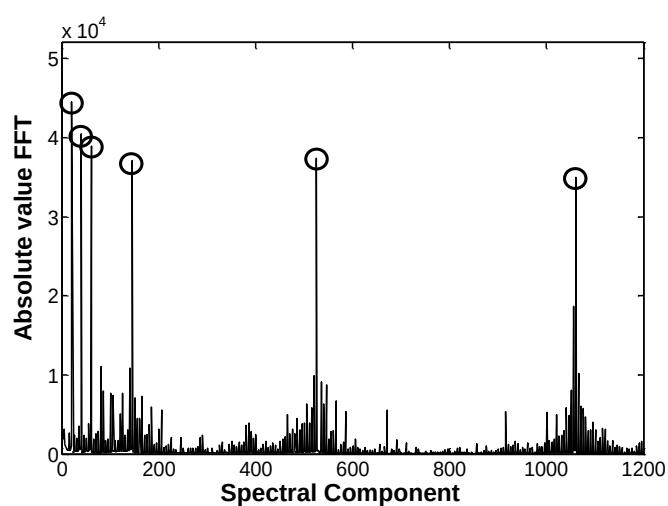
The response signals of the temperature-modulated micro-sensors in the presence of the different gases studied were obtained and processed as follows. The absolute value of the FFT was computed and the values of the 6 harmonics corresponding to the modulating frequencies were extracted. These were the 6 features used to build identification and quantification models. Figure 15 shows the FFT spectra of the transient response of a sensor in the presence of two different concentrations of acetaldehyde (50 and 100 ppm), ethylene (50 and 100 ppm) and ammonia (50 and 75 ppm). The peaks in these plots correspond to the temperature-modulating frequencies.



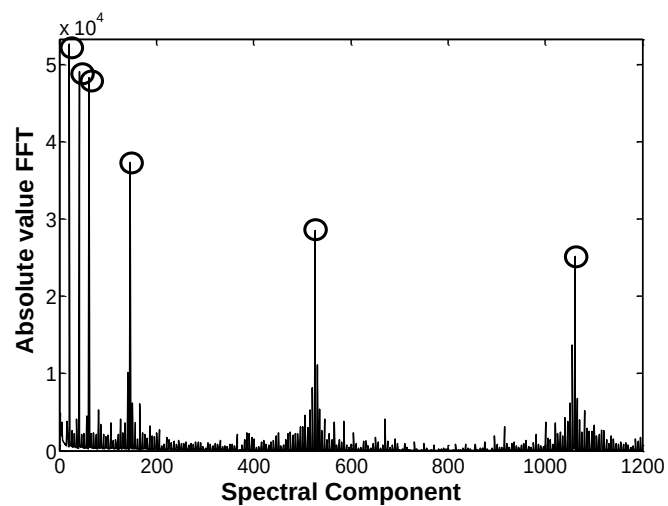
(a)



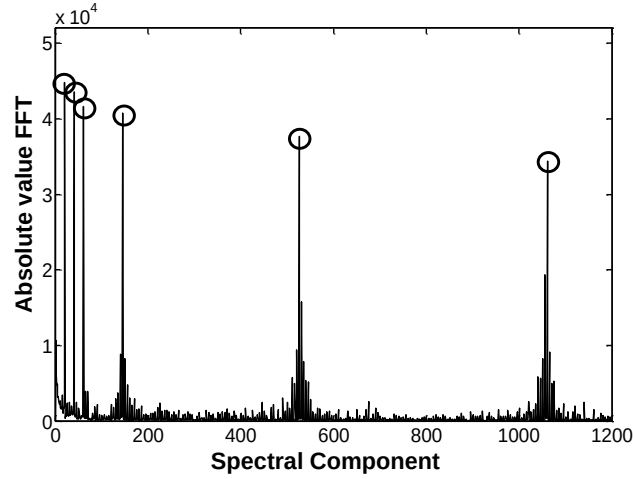
(b)



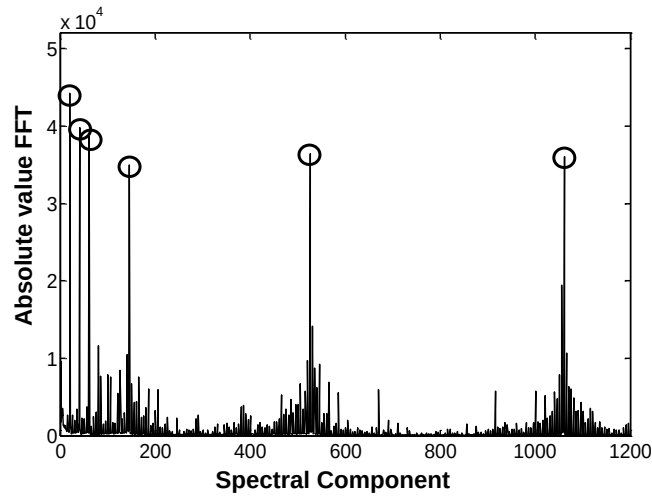
(c)



(d)



(e)



(f)

Figure: 15. FFT (absolute value) of the transient response of a temperature-modulated WO3 micro-hotplate sensor in the presence of (a) 50 ppm acetaldehyde, (b) 50 ppm ethylene, (c) 50 ppm of ammonia, (d) 100 ppm acetaldehyde, (e) 100 ppm ethylene, and (f) 75 ppm of ammonia. The temperature is modulated using a 6-frequency multi-sinusoidal signal.

In the first step the identification of the gases and gas mixtures was envisaged. A fuzzy ARTMAP classifier was built and validated (using the leave-

one-out approach) to discriminate among the different species (i.e. 6-category classification). The results of the identification process are shown in Table 11. A very high success rate in discrimination is reached, even using a single sensor. When the information from sensors 2, 3 and 4 was combined, gases and gas mixtures could be identified with a 100% success rate. This process was repeated using an additional feature, the D.C. component of the absolute value of the FFT. Adding this feature, which accounts for the mean value of transient signals (prone to be affected by drift), results in a slight degradation of the identification rate (see Table 11). These results prove that the modulating frequencies that are important for discriminating among the gases studied have been correctly identified.

Table 11: Success rate in gas identification estimated by leave-one-out cross-validation using a fuzzy ARTMAP classifier when the sensor working temperature is modulated by a sum of 6 sinusoidal signals.

Identification success rate ↓	Sensors→	2	3	4	2,3,4
Using 6 harmonics (%)		98.20	97.00	96.36	100
Using 6 harmonics + d.c. (%)		96.36	92.69	95.20	100

In the second step the building of PLS calibration models was envisaged. Like when MLPRS signals were used, specific calibration models for every gas or gas mixture were built and validated. The process employed to determine the number of latent variables to be used and the validation procedures are identical to the ones described for the MLPRS case. Table 12 summarizes the validation results for the different PLS models built. These results show that the concentration of the different gases can be accurately estimated.

To better assess the performance of the concentration estimation from the responses of the thermally-modulated sensors, an additional training and validation strategy was implemented. In this case training and validation matrices were defined as follows. Since there were three different concentrations for each species, a training matrix was formed using measurements from 2 out of the 3 concentrations available. The measurements corresponding to the concentration that had been left out integrated the validation matrix. Since with this validation approach the quantitative PLS models built are asked to predict concentrations that do not occur in the training set, this makes it possible to carefully checking

their generalisation ability. The procedure to build and validate the PLS models is analogous to the one described above.

Table 12: Validation results for the specific PLS calibration models. The harmonics of the FFT corresponding to the 6 frequencies of the multi-sinusoidal temperature-modulating signal were used as input data (information from sensors 2, 3 and 4 was used). Number of *LV* used, slope (*m*) and correlation coefficient (*r*) of the linear regression between real and predicted concentrations and root mean square error of cross-validation (RMSECV).

Gases/mixture models ↓	Validation results			
	LV#	M	r	RMSECV
Acetaldehyde	5	0.999	0.999	0.92
Ethylene	10	0.954	0.978	0.61
Ammonia	6	0.999	0.998	0.99
Ethylene + Acetaldehyde	9	0.936	0.945	12.13
		0.959	0.980	7.29
Ethylene + Ammonia	5	0.952	0.973	8.52
		0.981	0.985	3.58
Acetaldehyde + Ammonia	6	0.982	0.990	2.81
		0.968	0.985	3.57

Table 13 summarizes in detail the validation results for the different PLS models built. In addition to the RMSECV, in this table the slopes (*m*) and the correlation coefficients (*r*) of the linear regressions between actual and predicted concentrations are also shown. From this table it can be derived that an accurate estimation of gas concentration is possible using the thermally-modulated micro-hotplate sensors. This is an important result because with the validation strategy employed here, we are asking the PLS models to estimate concentrations that were not present in the training matrices. In other words, the method generalises well because it correctly interpolates gas concentration for the validation measurements.

This proves the usefulness of the optimization process conducted to identify the modulating frequencies that were important for gas quantification. Both identification and quantification results are better using the multi-sinusoidal signal than using the pseudo-random signal. This can be explained by the fact that the operating temperature sweep employed with multi-sinusoidal signals was higher than the one employed with pseudo-random signals. Summarizing, pseudo-random signals enable identification of optimal modulating frequencies

and, once these have been identified, the amplitude of the resulting multi-sinusoidal signal can help to further improve selectivity. A trade-off surely exists between the amplitude of the temperature modulating signal and the ageing of sensor membranes.

Table 13: Validation results for the specific PLS calibration models aimed at predicting gas concentration for single species and mixtures. Number of *LV* used, slope (*m*) and correlation coefficient (*r*) of the linear regression between real and predicted concentrations and root mean square error of cross-validation (RMSECV).

Gases/mixture models ↓	Validation results			
	LV#	M	r	RMSECV
Acetaldehyde	3	0.998	0.999	0.99
Ethylene	4	0.967	0.986	7.44
Ammonia	5	1.02	0.995	2.73
Ethylene in Ethylene + Acetaldehyde	13	1.003	0.979	9.42
Acetaldehyde in Ethylene + Acetaldehyde	4	0.945	0.968	7.91
Ethylene in Ethylene + Ammonia	3	0.960	0.980	8.296
Ammonia in Ethylene + Ammonia	5	0.995	0.991	2.971
Acetaldehyde in Acetaldehyde + Ammonia	7	0.991	0.987	1.038
Ammonia in Acetaldehyde + Ammonia	3	0.988	0.989	3.66

The fact that the frequency optimization process was conducted on a different micro-sensor array than the one used for validation proves that this process is consistent and robust.

3.4.2.4 Steady-state response

Finally, to better assess the improvement in gas identification and quantification obtained by an optimized modulation of the sensors' operating temperature, the

gas identification and quantification problem was re-considered using the steady-state sensor response, which is the traditional way to operate gas sensors. Identification and quantification tasks were attempted using the steady-state value of the normalized resistance change, (R/R_o) , experienced by the sensors in the presence of gases or gas mixtures. The static normalized resistance change was defined as:

$$\frac{\Delta R}{R_o} = \frac{R - R_o}{R_o} \quad (22)$$

where R_o is the baseline resistance (i.e. in the presence of dry air) and R is the steady-state resistance of the sensor in the presence of a given gas or gas mixture.

Remembering from the experimental database, that for every measurement, the steady-state response was stored before acquiring the transient response due to temperature modulation, a database with 165 steady-state measurements was available to perform this analysis.

Table 14: Comparative success rates in gas and gas mixture identification using fuzzy ARTMAP classifiers (leave-one-out cross-validation) together with temperature modulation or steady-state methods.

	MLPRS	Multi-sinusoidal	Steady-state
Identification success rate (%)	87.3	100	81

¹The first micro-sensor array was used.

²The second micro-sensor array was used.

Initially, the identification of gases was attempted. A fuzzy ARTMAP classifier, which used as inputs the steady-state response of the sensors within the array was built and validated using the leave-one-out approach. Gases and gas mixtures could be identified with an 81% success rate, which is significantly worse than the identification rate reached when transient information was used (e.g. 100% when using multi-sinusoidal temperature modulation). Table 14 shows a comparison of these results.

The quantification analysis (by building PLS calibration models) was also envisaged. As in previous cases, specific calibration models for every gas or gas mixture were built and validated. The process employed to determine the number of latent variables to be used and the validation procedures are identical to the ones described above. Table 15 summarises the validation results for the different

PLS models built. These results show that the concentration of the different gases cannot be accurately estimated when using the steady-state sensor response only.

Table 15: Validation results for the specific PLS calibration models. The steady-state sensor response was used. Number of *LV* used, slope (*m*) and correlation coefficient (*r*) of the linear regression between real and predicted concentrations and root mean square error of cross-validation (RMSECV).

Gases/mixture models ↓	Validation results			
	LV#	m	r	RMSECV
Acetaldehyde	3	0.9	0.9	8.2
Ethylene	3	0.86	0.89	16.2
Ammonia	3	0.93	0.91	6.36
Ethylene + Acetaldehyde	2	0.22	0.44	33.05
		0.86	0.92	14.0
Ethylene + Ammonia	3	0.81	0.89	16.60
		0.83	0.90	8.86
Acetaldehyde + Ammonia	3	0.82	0.88	9.17
		0.37	0.57	16.81

Therefore, these results prove that optimizing the temperature-modulation frequencies of metal oxide gas sensors is essential if quantitative gas analysis is to be performed.

3.5 CONCLUSIONS

In recent years, the lack of reproducibility, stability and selectivity have been considered as some of the major problems experienced with gas sensing systems based on metal oxide gas sensors. Thermal modulation of metal oxide gas sensors has been one of the most commonly used methods to enhance sensor selectivity and counteracting drift. Nevertheless, the selection of the frequencies used to modulate the working temperature had remained most of the times, an empirical process.

In this chapter we have discussed a systematic method to determine which are the optimal temperature modulation frequencies to solve a given gas analysis problem. This method, which is borrowed from the field of system identification

and intelligent systems engineering, has been developed and introduced in the area of gas sensors. The approach has been discussed in detail, illustrated with a practical application and fully validated in this field.

It consists of studying the sensor response to gases when their operating temperature is modulated via maximum-length pseudo-random sequences (either binary or multi-level). Such signals share some properties with white noise and, therefore, can be of help to estimate the linear response of a system with non-linearity (e.g., the impulse response of a sensor-gas system).

The optimization process is conducted by selecting among the spectral components of the impulse response estimates, the few that better help either discriminating or quantifying the target gases of a given gas analysis application. In this case in particular those spectral components were computed via the fast Fourier transform (FFT). Since spectral components are directly related to modulating frequencies, the selection of spectral components results in the determination of the optimal temperature modulating frequencies. Pattern recognition algorithms ranging from linear, (e.g., unsupervised such as Principal component analysis (PCA), or supervised such as partial least squares (PLS) and PLS-DA), to artificial neural networks such as the fuzzy ARTMAP were used in this chapter to assess the performance of the method discussed.

In the first experiment, pseudo-random binary signals (PRBS) were employed to modulate the working temperature of micro-machined metal oxide gas sensors in a frequency range from 0 up to 1 Hz. The outcome of a preliminary study was that the important modulating frequencies were in the range described. This is understandable, since the kinetics of the reaction and adsorption processes taking place at the sensor surface are slow and if these are to be altered by the thermal modulation, low frequency modulating signals need to be devised. This explains why low-frequency temperature-modulating signals (i.e. in the mHz range) have been used with micro-hotplate gas sensors, even though the thermal response of their membranes is much faster (typically, near 100 Hz).

An evolved method to determine the optimal temperature modulating frequencies for micro-hotplate gas sensors was introduced in the experiment that followed the first one, which was based on the use of maximum length multilevel pseudo-random sequences (MLPRS). Multilevel signals were considered instead of the binary ones because the former can provide a better estimate than the latter of the linear dynamics of a process with non-linearity. And it is well known that temperature-modulated metal oxide gas sensors present non-linearity in their response.

These systematic studies were fully validated by synthesizing multi-sinusoidal signals at the optimal frequencies previously identified using pseudo-random sequences. When the sensors had their operating temperatures modulated by a signal with a frequency content that corresponded to the optimal, the gases and gas mixtures considered could be perfectly discriminated and the building of accurate calibration models to predict gas concentration (after identification process) was found to be possible. In some cases, the validation process was conducted on sensors that had not been used for optimization purposes (e.g. a different sensor array from the same fabrication batch). Furthermore, the qualitative and quantitative results obtained using temperature-modulated sensors with optimized modulating frequencies compare very favourably with the results when the steady-state sensor response is used.

The main conclusion that can be drawn from the method introduced in this chapter is that an optimized thermal modulation of their working temperature can significantly increase the selectivity of metal oxide sensors. Therefore, the systematic optimization process to implement the selection of modulating frequencies discussed makes it possible to ensure a significant increase in performance for metal oxide based multi-sensor systems. We can envisage that for each gas-sensor pair, the modulating frequencies selected may be used to characterize the interaction between the metal oxide layer and the gas (e.g. film microstructure, surface diffusion and reaction kinetics). The simplicity of the methods implemented makes them suitable for the development of low-cost gas analysers and hand-held electronic noses. Finally, the method applies generally and can be used in any gas analysis problem or extended to other types of sensors (e.g. conducting polymer sensors).

References

- Advani G. N., R. Beard, L. Nanis, Gas measurement method, US Pat. 4399684, August 23 (1983).
- Albert, A.A., (1956), *Fundamental Concepts of Higher Algebra*, University of Chicago Press, Chicago, USA.
- Althainz P., A. Dahlke, J. Goschnick and H.J. Ache, (1994), "Low temperature deposition of glass membranes for gas sensors", *Thin Solid Films* 241: 344-347.
- Arthur C.L., and J. Pawliszyn, (1990), "Solid phase microextraction with thermal desorption using fused silica optical fiber", *Analytical Chemistry* 62: 2145-2148.
- Arthur C.L., L.M. Killam, S. Motlagh, M. Lim, D.W. Potter and J. Pawliszyn, (1991), "Analysis of substituted benzene compounds in groundwater using solid-phase microextraction", *Environmental Science & Technology* 26 (5): 979-983.
- Barker H.A., (1967), "Choice of pseudorandom binary signals for system identification", *Electron. Lett.* 3: 524-526.
- Barker, H. A., (1986), *Int. J. Math. Educ. Sci. Technol.* 17: 473-485.
- Briggs P. A. N., P. H. Hammond, M. T. G. Hughes, G. O. Plumb, (1965), "Correlation analysis of process dynamics using pseudo-random binary test perturbation", *Proc. Inst. Mech. Engrs.* 179, Part 3H: 37-51.
- Carpenter G., S.Grossberg, N. Markuzon, J. Reynolds, D. Rosen, (1992), "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps", *IEEE Trans. Neural Nets.* 3: 698 - 713.
- Cavicchi R. E., J. S. Suehle, K.G. Kreider, M. Gaitan, P. Chaparala, (1995), "Fast temperature programmed sensing for micro-hotplate gas sensors", *IEEE Electr. Device Letters* 16 (6): 286-288.
- Cavicchi R. E., J. S. Suehle, K.G. Kreider, M. Gaitan, P. Chaparala, (1996), "Optimized temperature-pulse sequences for the enhancement of chemically specific response patterns from micro-hotplates gas sensors", *Sensors and Actuators B: Chemical* B 33, 142-146.
- Chen J., and J. Pawliszyn, (1995), "Solid phase microextraction coupled to high performance liquid chromatography", *Analytical Chemistry* 67: 2530-2533.
- Church R., *Annals of Mathematics* 36: 198-209.
- Clifford P. K., D. T: Tuma, (1982-1983), "Characteristics of semiconductor gas sensor II, Transient response to temperature change", *Sensors and Actuators B: Chemical* B 3: 233-254.
- Clifford P. K., *Mechanisms of gas detection of metal oxide surface*, PhD thesis, Carnegie-Mellon University, 1981.
- Cosandey F., G. Skandan and A. Singhal, (2000), *Materials and processing issues in nanostructured semiconductor gas sensors*, *JOM-e*, 52, (10).
- Demarne V. and R. Sanjines, (1992), *Gas sensors principles, operation and development*, Kluwer Academic, ed. G. Sberveglieri, Dordrecht, the Netherlands: 89-116.
- Dori L., S. Nicoletti and I. Elmi, (2000), "A gas chromatographic-like system for the separation and monitoring of benzene, toluene and xylene compounds at the ppb level using solid state metal oxide gas sensors", *Sens. Mater J.* 12 (3): 163-174.
- Duran C., J. Brezmes, E. Llobet, X. Vilanova, and X. Correig, (2005), "Enhancing sensor

- selectivity through flow modulation”, book of abstracts of the IEEE sensors 2005: the 4th IEEE conference on sensors California USA: 428-431.
- Eicker H., (1977), Method and apparatus for determining the concentration of one gaseous component in a mixture of gases, US Pat. 4012692.
- Everett D., (1966), “Periodic digital sequences with pseudonoise properties”, GEC Jnl. Sci. Technol. 33: 115-126.
- Fenaille, F.; Visani, P.; Fumeaux, R.; Milo, C.; Guy, P. A., (2003), “Comparison of Mass Spectrometry-Based Electronic Nose and Solid Phase MicroExtraction Gas Chromatography-Mass Spectrometry Technique to Assess Infant Formula Oxidation”, Journal of Agricultural and Food Chemistry 51 (9): 2790-2796.
- Feng C.D., Y. Shimizu and M. Egashira, (1994), “Effect of gas diffusion process on sensing properties of SnO₂ thin film sensors in a SiO₂/SnO₂ layer-built structure fabricated by sol-gel process”, Journal of The Electrochemical Society 141 (1): 220-225.
- Gardner J. W. and P. N. Bartlett, Electronic noses principles and applications, (1999), Oxford University press.
- Gardner J.W., (1991), “Detection of vapours and odours from a multi-sensor array using pattern recognition. Part 1: Principal component and cluster analysis”, Sensors and Actuators B: Chemical B 4: 109-115.
- Geladi P., B. R. Kowalski, (1986), “Partial least squares regression: a tutorial”, Analytica Chimica Acta 185: 1-17.
- Godfrey K., (1980), Correlation methods, Automatica 16: 527-534.
- Godfrey K., (1993), Perturbation signals for system identification, Prentice Hall, UK,.
- Göpel W., (1994), “New materials and transducers for chemical sensors”, Sensors and Actuators B: Chemical B 18-19, 1-21.
- Gràcia I., J. Santander, C. Cané M. C. Horrillo, I. Sayago, J. Gutierrez, (2001), “Results on the reliability of silicon micromachined structures for semiconductor gas sensors”, Sensors and Actuators B: Chemical B 77 409-415.
- Heiling A., N. Bâsan, U. Weimer, M. Schweizer-Berberich, J.W. Gardner and W. Göpel, (1997), “Gas identification by modulating temperatures of SnO₂-based thick film sensors”, Sensors and Actuators B: Chemical B 43, 45-51.
- Hohl D. , (1990), “The role of noble metals in the chemistry of solid-state gas sensors”, Sensors and Actuators B: Chemical B 1: 158-165.
- Horrillo M. C., I. Sayago, L. Arés, J. Rodrigo, J. Gutiérrez, A. Götz, I. Gràcia, L. Fonseca, C. Cané E. Lora-Tamayo, (1999), “Detection of low NO₂ concentrations with low power micromachined tin oxide gas sensors”, Sensors and Actuators B: Chemical B 58: 325-329.
- Ionescu R., E. Llobet, X. Vilanova, J. Brezmes, J. E. Suerias, J. Calderer, X. Correig, (2002), “Quantitative analysis of NO₂ in the presence of CO using a single tungsten oxide semiconductor sensor and dynamic signal processing”, Analyst 127: 1237-1246.
- Ivanov P. , M. Stankova, E. Llobet, X. Vilanova, J. Brezmes, I. Gràcia, C. Cané J. Calderer, and X. Correig, (2005), “Nanoparticle metal oxide films for micro-hotplate-based gas sensors systems”, IEEE Sensors Journal 5: 798-809.
- Ivanov P., E. Llobet, A. Vergara, M. Stankova, X. Vilanova, J. Hubalek, I. Gracia, C. Cané X. Correig, (2005), “Towards a micro-system for monitoring ethylene in warehouses”, Sensors and Actuators B: Chemical B 111-112: 63-70.
- Kohl D., (1990), “The role of noble metals in the chemistry of solid-state gas sensors”, Sensors and Actuators B: Chemical B 1: 158-164.

- Kunt T.A., T.J. McAvoy, R.E. Cavicchi, S. Semancik, (1998), "Optimization of temperature programmed sensing for gas identification using micro-hotplate sensors", *Sensors and Actuators B: Chemical* B 53: 24-43.
- Le Vine H. D., (1975), Method and apparatus for operating a gas sensor, US Pat. 3906473.
- Ljung L., (1987), *System Identification: Theory for the User*, Englewood Cliffs, New Jersey: Prentice Hall, Chapters 6 and 8 to 11.
- Llobet E., (1997), "Selectivity enhancement of metal oxide semiconductor chemical sensors through the study of their transient response to a step-change in gas concentration", Doctoral Thesis, Universitat Politècnica de Catalunya, Barcelona.
- Llobet E., J. Brezmes, R. Ionescu, X. Vilanova, S.Al-Khalifa, J.W. Gardner, N. Barsan, X. Correig, (2002), "Wavelet transform fuzzy ARTMAP based pattern recognition for fast gas identification using a micro-hotplate gas sensor", *Sensors and Actuators B: Chemical* B 83: 238-244.
- Llobet E., J. Brezmes, X. Vilanova, J.E. Sueiras and X. Correig, (1997), "Qualitative and quantitative analysis of volatile organic compounds using transient and steady-state responses of a thick-film tin oxide gas sensor array", *Sensors and Actuators B: Chemical* B 4: 113-22.
- Llobet E., P. Ivanov, X. Vilanova, J. Brezmes, J. Hubalek, K. Malysz, I. Gracia, C. Cané X. Correig, (2003), "Screen-printed nanoparticle tin oxide films for high-yield sensor Microsystems", *Sensors and Actuators B: Chemical* B 96: 94-104.
- Llobet E., R. Ionescu, S. Al-Khalifa, J. Brezmes, X. Vilanova, X. Correig, N. Bâsan, J.W. Gardner, (2001), "Multicomponent gas mixture analysis using a single tin oxide sensor and dynamic pattern recognition", *IEEE Sensors Journal* 1: 207-213.
- Llobet E., X. Vilanova, J. Brezmes, J.E. Sueiras, R. Alcubilla and X. Correig, (1998), "Model for the steady-state and transient behaviour of thick-film tin oxide gas sensors in the presence of gas mixtures", *Journal of The Electrochemical Society* 145 (5): 1772-1779.
- Marcel Dekker, Inc., 2002, *Flavor, Fragrance and Odor Analysis*, ed Ray Marsili, New York.
- Marcos Lorenzo, I, Pérez Pavón, J.L, Fernández Laespada M.E., García Pinto, C., Moreno Cordero, B., Henriques, L. R, Peres, M.F, Simões M.P, Lopes P.S, (2002), "Application of headspace-mass spectrometry for differentiating sources of olive oil", *Anal Bional Chem.* 374: 1205-1211.
- Martinelli G., M.C. Carotta, E. Traversa and G. Ghiotti, (1999) (a) "Thick-film gas sensors based on nano-sized semiconducting oxide powders", *MRS Bulletin*: 30-35.
- Martinelli G., M.C. Carotta, M. Ferroni, Y. Sadaoka and E. Traversa, (1999) (b), "Screen-printed perovskite-type thick films as gas sensors for environmental monitoring", *Sensors and Actuators B: Chemical* B 55 (2-3): 99-110.
- Moseley P. T., B. C. Tofield, (1987), Eds., *Solid-state gas sensors*, The Adam Hilger series on sensors, Bristol, UK.
- Moseley P.T., A.M. Stoneham and D.E. Williams, (1991), *Techniques and mechanisms in gas sensing*, The Adam Hilger series on sensors, Bristol, U.K.
- Mukhopaghyay A.K., P. Mitra, A.P. Chatterjee, H.S. Maiti, (1998), "A new method of preparing tin dioxide thin film sensors", *Journal of Materials Science Letters* 17: 625-627.
- Nakata S., E. Ozaki, N. Ojima, (1998), "Gas sensing based on the dynamic nonlinear responses of a semiconductor gas sensor: dependence on the range and frequency of a cyclic temperature change", *Analytica Chimica Acta* 361: 93-100.
- Nakata S., S. Akakabe, M. Nakasuji, K. Yoshikawa, (1996), "Gas sensing based on a nonlinear response: discrimination between hydrocarbons and quantification of individual components

- in a gas mixture”, *Analytical Chemistry* 68: 2067-2072.
- Nakata S., Y. Kaneda, K. Yoshikawa, (1992), “Novel strategy to develop chemical sensors based on nonlinear dynamics-intelligent gas sensor”, *Sens. Mater.* 4: 101-110.
- Openheim A.V. and Schafer K.W., (1989), *Discrete-time signal processing*, Englewood Cliffs, Prentice Hall, N.J., U.S.A.
- Owen L. J., (1980), *Gas monitors*, US Pat. 4185491.
- Park C.O., S.A. Akbar, J. Hwang, (2002), “Selective gas detection with catalytic filter”, *Materials Chemistry and Physics* 73 (1-3): 56-60.
- Pérez Christophe, Frédéric Begnaud and Jean-Louis Berdagué, (2002), “Standard Gas Addition: A Calibration Method for Handling Temporal Drifts of Mass Spectrometry-Based Sensors”, *Analytical Chemistry* 74: 2279-2283.
- Persaud K. and G. H. Dodd, (1982), “Analysis of discrimination mechanisms of the mammalian olfactory system using a model nose”, *Nature* 299: 352-355.
- Privat E., S. Roussel, P. Grenier, V. Bellon-Maurel, (1998), *Sci. Aliments* 18 (5): 459-470.
- Rickerby D., N. Wächter, M. Horrillo, J. Gutiérrez, I. Gràcia, C. Cané, (2000), “Structural and dimensional control in micromachined integrated solid state gas sensors”, *Sensors and Actuators B: Chemical* 69: 314-319.
- Roberts M. J., (2004), *Signals and Systems Analysis using transform methods and MATLAB*, McGraw Hill, Tennessee, USA.
- Sberveglieri G., (1995), “Recent developments in semiconducting thin-film gas sensors”, *Sensors and Actuators B: Chemical* 23: 102-109.
- Schaller E., S. Zenhäusern, T. Zesiger, J.O. Bosset and F. Escher, (2000), “Use of pre-concentration techniques applied to a MS-based electronic nose”, *Analysis* 28: 743-749.
- Schaller E., S. Zenhäusern, T. Zesiger, J.O. Bosset, F. Escher, (2000), “Use of pre-concentration techniques applied to a MS-based Electronic Nose”, *Analysis* 28: 743-749.
- Sears W. M., K. Colbow, F. Consadori, (1989) (a), “General characteristics of thermally cycled tin oxide gas sensors”, *Semiconductor Science and Technology* 4: 351-59.
- Sears W.M., K. Colbow, F. Consadori, (1989) (b), “Algorithms to improve the selectivity of thermally cycled tin oxide gas sensors”, *Sensors and Actuators* 19: 333-49.
- Shimizu Y., and M. Egashira, (1999), “Basic aspects and challenges of semiconductor gas sensors”, *MRS Bulletin*: 18-24.
- Shurmer H.V., J.W. Gardner and P. Carcoran, (1990), “Intelligent vapour discrimination using a composite 12-element sensor array”, *Sensors and Actuators B: Chemical* 1: 256-260.
- Stahnke W., (1973), “Primitive polynomials”, *Math Comput.* 27: 977-980.
- Vergara A., E. Llobet, J. Brezmes, P. Ivanov, C. Cané I. Gràcia, X. Vilanova and X. Correig, (2007), “Quantitative gas mixture analysis using temperature-modulated micro-hotplate gas sensors: Selection and validation of the optimal modulating frequencies”, *Sensors and Actuators B: Chemical* 123: 1002-1016.
- Vergara A., E. Llobet, J. Brezmes, X. Vilanova, P. Ivanov, I. Gràcia, C. Cané X. Correig, (2005) (a), “Optimized temperature modulation of micro-hotplate gas sensors through pseudo random binary sequences”, *IEEE Sensors Journal* 5: 1369-1378.
- Vergara A., E. Llobet, J. Brezmes, P. Ivanov, X. Vilanova, I. Gràcia, C. Cané X. Correig, (2005) (b), “Optimised temperature modulation of metal oxide micro-hotplate gas sensors through multilevel pseudo random sequences”, *Sensors and Actuators B: Chemical* 111-112: 271-280.
- Vicenzi D., A. Butturi, V. Guidi, M. Carotta, G. Martinelli, V. Guarnieri, S. Brida, B. Margesin, F.

- Giaocemozzi, M. Zen, D. Giusti, G. Soncini, A. Vasiliev, and A. Pisliakov, (2000), "Gas-sensing device implemented on a micromachined membrane: A combination of thick-film and very large scale integrated technologies", *J. Vac. Sci. Technol. B* 18: 2441–2445.
- Vinaixa M., A. Vergara, C. Duran, E. Llobet, C. Badia, J. Brezmes, X. Vilanova, X. Correig, (2005), "Fast detection of rancidity in potato crisps using e-noses based on mass spectrometry or gas sensors", *Sensors and Actuators B: Chemical B* 106: 67-75.
- Volterra V., *Theory of Functionals and of Integro-differential equations*, New York, Dover, 1930.
- Watson E. J., (1962), "Primitive polynomials (modulo 2)", *Math Comput.* 16: 368-369.
- White N. M., J. D. Turner, (1997) "Thick-film sensors: past, present and future", *Measurement Science and Technology* 8: 1-20.
- www.alpha-mos.com/
- www.gerstel.de/
- www.hkr-sensor.de/start.
- www.smartnose.com/
- www.znose.com
- Xu C., J. Tamaki, N. Miura and N. Yamazoe, (1991), "Grain size effects on gas sensitivity of porous SnO₂-based elements", *Sensors and Actuators B: Chemical B* 3-2 147-155.
- Yamazoe N., (1991), "New approaches for improving semiconductor gas sensors", *Sensors and Actuators B: Chemical B* 5: 7-19.
- Zierler N., (1959), "Linear recurring sequences", *J. Soc. Ind. Appl. Math.* 7: 31-48.

Chapter 4

Support Vector Machines for Electronic Nose Pattern Analysis

M. Pardo and S. De Vito

Support vector machines (SVMs) are generalized linear discrimination machines developed in the early 90s in the framework of structural risk minimization. They have recently been employed in the electronic nose (EN) community for pattern analysis in both classification and regression problems. In this contribution we provide an overview of SVM theory, review the literature involving SVM in artificial olfaction and describe some case histories, regarding both classification and regression with SVM. As for any other machine learning technique, SVM require hyper parameter (HP) selection. In the applications we show how hyper parameter tuning permits exploration of the insurgence of underfitting and overfitting effects, which are the principal limitations of non parametric learning techniques.

4.1 INTRODUCTION

During 1960-1990, Vapnik and Chervonenkis conceived and developed a novel statistical learning framework, known as VC theory, which then in the 90s gave birth to support vector machines (SVMs) (Vapnik 1995; Burges 1998). Over the course of the last decade or so, SVMs became very popular as a tool for solving classification, regression and novelty detection problems in a broad range of application areas (Guyon).

SVMs are generalized linear discrimination machines that take advantage from operating in a transformed, typically higher dimensional, space in order to best discriminate classes. The basic idea behind SVMs - originally developed for

classification - is to select the optimal class separating hyperplane having maximum distance (margin) from the convex hulls of the classes. This choice can be motivated in the framework of statistical learning theory and intuitively lead to enhanced generalization properties. Subsequently, SVMs have been generalized for regression and functional approximation problems and in this formulation are also known as SVRs (support vector regressors).

Furthermore, the SVMs target problem is convex and this makes the solution of their optimization problem unique and relatively straightforward, avoiding local minima issues. It can be seen that the position of the maximum margin boundary hyperplane is determined only by a subset of the population training set, i.e. the ones contributing to the convex hulls of the classes, termed support vectors. This accounts for significant computational advantages over most classic kernel based discrimination algorithms. In this sense, SVMs are classified as a sparse solution kernel based discrimination method (Bishop 2006). As a drawback, SVMs do not provide posterior probability estimations; relevance vector machines (RVMs) based on a Bayesian approach can offer a valid alternative at the cost of a non convex optimization procedure (Tipping 2001).

Today, every new pattern recognition book has a chapter dedicated to kernel methods and SVMs. Below we summarize the main points of SVM, which are needed to understand SVM applications. Treatments of various lengths can be found in references (Burges 1998; Cristianini and Shawe-Taylor 2000; Bishop 2006).

4.1.1 Basic SVM theory for classification

In a two class classification problem let

$$y(x) = w^T \phi(x) + b \quad (1)$$

be a discrimination function with $y(x) > 0$ if x belongs to class A and $y(x) < 0$ when x belongs to class B . If classes are linearly separable in the transformed (feature) space, there will exist at least one solution w for which $t(x)y(x) > 0$ for all training data points with $t(x)$ being the ground truth function assigning $+1$ to samples belonging to class A and -1 to samples belonging to class 2 .

In case of linear separability, we can express the distance of a training point x' to a decision boundary defined by (1) as:

$$\frac{t(x')y(x')}{\|w\|} = \frac{t(x')(w^T \phi(x') + b)}{\|w\|} \quad (2)$$

So maximizing the margin can be achieved by solving:

$$\arg \max_{w,b} \left\{ \frac{1}{\|w\|} \min_x [t(x')(w^T \phi(x') + b)] \right\} \quad (3)$$

Without losing generality, we can set $t(x)(w^T \phi(x) + b) = 1$ for the point that is closest to the separating hyperplane. In this case all data points will satisfy the constraint:

$$t(x')(w^T \phi(x') + b) \geq 1 \quad (4)$$

In a two class problem by definition, there will be at least two points for which the equality for the maximized margin holds. So the optimization problem simply requires the minimization of w subject to Eq.(4) for all training points. This can be solved as a quadratic programming problem with a set of linear inequality constraints. Actually, in the SVM framework, we solve the dual problem obtained by introducing Lagrange multipliers giving the following primal Lagrangian function to be minimized:

$$\frac{1}{2} \|w\|^2 - \sum_n a_n [t(x_n)(w^T \phi(x_n) + b) - 1] \quad (5)$$

with the index n running from 1 to the cardinality of the training set. Setting w and b related derivatives to 0, we obtain the following conditions:

$$w = \sum_n a_n t(x_n) \phi(x_n) \quad (6)$$

$$0 = \sum_n a_n t(x_n) \quad (7)$$

Using (6) and (7) we obtain the following dual representation of the maximum margin problem in which we maximize

$$\sum_n a_n - \frac{1}{2} \sum_n \sum_m a_n a_m t(x_n) t(x_m) k(x_n, x_m) \quad (8)$$

with n and m running from 1 to training set cardinality, with respect to the constraints:

$$a_n \geq 0 \text{ for each } n \quad (9)$$

and

$$\sum_n a_n t(x_n) = 0 \quad (10)$$

Note that $\phi(x_n)^T \phi(x_m)$ has been denoted with $k(.,.)$, the notorious kernel function. In the dual formulation, the basis functions appears only as a dot product in the feature space and can be defined implicitly by kernel functions that satisfy Mercer's conditions (Mercer 1909; Bishop 2006). First kernels investigated in pattern recognition problems were the polynomial $k(x,y)=(xy+1)^p$, the Gaussian $k(x,y)=\exp(-\gamma \|x-x'\|^2)$ and $k(x,y)=\tanh(-Kxy-\delta)$. It is worth noting that we have changed a M variables quadratic problem, where M is the number of basis functions into a N variables quadratic programming problem with N being the cardinality of the training set. This find its convenience and significance especially when dealing with an infinite number of basis functions.

Once we have defined the optimal a_n values we can classify a sample directly in the problem domain by using

$$y(x) = \sum_n a_n t(x_n) k(x, x_n) \quad (11)$$

obtained by substituting (6) in the original $y(x)$ definition.

By using *Karhush-Kuhn-Tucker* (KKT) conditions (Vapnik 1995; Bishop 2006), that are satisfied by this kind of constrained optimization problems, we obtain that a_n is non zero valued only for a subset of training points. They turn to be the ones lying, once projected in the feature space, on the maximum margin hyperplanes determining the decision boundary position and for which $t(x_n)y(x_n)=1$.

This result is of paramount importance for the practical applicability of SVM because only the support vectors are needed to evaluate a sample and to assign it to one of the two classes. Again, by using KKT conditions, b can also be evaluated starting from (11).

Taking a step forward, the rather strong assumption of a class's linear separability in the feature space can be now removed; actually, pursuing exact separation in case of overlapping distributions can lead to overtraining errors and so poor generalization properties. To do this, the SVM framework uses slack variables $\xi_n \geq 0$ for each n . ξ is defined to be 0 for each sample being on the right side of the class boundary defined by support vectors and being equal to $|t(x_n) - y(x_n)|$ for the other points so that points lying on the decision hyperplane hold a ξ value equal to unity while points having a $\xi > 1$ have been misclassified.

Exact classification constraints are replaced with:

$$t(x_n)(w^T \phi(x_n) + b) \geq 1 - \xi_n \quad (12)$$

in which slack variables are bound to be greater than or equal to 0. We will now minimize

$$C \sum_n \xi_n + \frac{1}{2} \|w\|^2 \quad (13)$$

C is a penalty coefficient that will penalize empirical error balancing its value with model complexity term. A very high C value will eventually lead to the separable class solution. The quadratic problem becomes Eq.(8), subject to

$$0 \leq a_n \leq C \quad (14)$$

$$\sum_n a_n t(x_n) = 0 \quad (15)$$

Even in this case the solution is guaranteed to be unique by the fact that any local minima will be a global optimum. We can also see that although operative sample evaluation requires the use of support vectors alone (by the KKT conditions), training (i.e. solving Eq.(11)) requires the use of all training set samples and thus can be computationally demanding; however a set of algorithmic techniques have been developed to overcome this problem. In particular, one of the most popular approaches is sequential minimal optimization or SMO (Platt 1999).

SVM were developed as two-class classifiers. Various methods have been introduced to solve multiclass problems, e.g. by combining multiple SVMs. Typically, in a K-class problem one could train $K(K-1)/2$ separate SVMs in a one-versus-one approach. This approach can lead to heavy computational load due to the rapidly growing number of SVM you have to train even with relatively low K values. A more practicable approach for $K > 5$ is the common one-against-all in which you have to train only K different classifier. Then, $y(x)$ values can be used as a sort of a posteriori probability index, so as to assign x to the right class even when the sample is assigned to multiple classes by the SVMs array. However this technique can lead to a biased and inconsistent behavior due to the fact that $y(x)$ for different classifiers have different scales. In the 2000 work by Platt (Platt 2000), the author proposes a scheme to overcome the bias consistently

estimating class a posteriori probabilities. A number of further approaches have been developed to allow the use of SVM in the multiclass case. They range from the modification of the SVM objective function for training all SVMs simultaneously (Vapnik 1998) to generalized voting schemes. The work by Hsu (Hsu and Lin 2002) presents an interesting performance assessment of these methods. Even if application of SVMs to multiple classe problems still remain an open mathematical issue, from the practical point of view, ‘one versus the rest’ approach is generally considered a sufficiently robust methodology despite its limitations.

4.1.2 SVR Theory

In 1996, SVM has been extended to regression problems and in this formulation it is known also as SVR (Drucker, Burges et al. 1996). In the framework of regularized linear regression we try to minimize:

$$\frac{1}{2} \sum_n [y(x_n) - t(x_n)]^2 + \frac{\lambda}{2} \|w\|^2 \quad (16)$$

with lambda being the regularization parameter set to balance model complexity and empirical error. In ϵ -SVRs (Vapnik 1998), the quadratic empirical error term in Eq.(16) is replaced by a relaxed form called the ϵ -insensitive error function which is bound to be zero if the absolute prediction error is less than ϵ , with ϵ being a positive threshold value. So we have to minimize

$$C \sum_n E_\epsilon(y(x_n) - t(x_n)) + \frac{1}{2} \|w\|^2 \quad (17)$$

where $y(x)$ is given by Eq.(1) and lambda being included in the C value. This is equivalent to defining a so called ϵ -insensitive tube around $t(x)$ inside which error is neglected. Minimization of Eq.(17) is only dependant on samples whose $y(x)$ is outside the tube preserving the sparseness properties of SVMs. A typical choice for the non linear error function in Eq.(17) is to assign a linear cost to errors outside the insensitive region. We can now reformulate the problem in terms of slack variables allowing a different positive and negative slack for each training set point so to permit the predictions to lie outside the ϵ -insensitive tube.

In this case we want to minimize:

$$C \sum_n (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|w\|^2 \quad (18)$$

under the constraints:

$$t(x_n) \leq y(x_n) + \varepsilon + \xi_n \quad (19)$$

and

$$t(x_n) \geq y(x_n) - \varepsilon - \hat{\xi}_n \quad (20)$$

provided also that ξ_n and $\hat{\xi}_n$ are positive or null.

Reformulating the optimization problem using Lagrange multipliers, similarly to the classification theory, eventually leads to a regression function containing a kernel:

$$y(x) = \sum_n (a_n - \hat{a}_n) k(x, x_n) + b \quad (21)$$

KKT conditions still ensure that only data points that lie outside the boundary of the ε -insensitive tube, are support vectors and we have again a sparse solution.

4.2 SVMS IN ARTIFICIAL OLFACTION

In the last few years SVM has begun to be applied in the field of artificial olfaction, primarily in classification problems. Its enhanced generalization properties have been particularly appreciated as a partial solution to solid state sensor drift and selectivity issues.

Electronic noses (ENs) are gas mixture analysis devices whose architectures and functional principles were designed to mimic the mammals olfactory system (Persaud and Dodd 1982). They are typically based on an array of solid state sensor devices producing gas mixture specific response patterns. EN response patterns are then processed by means of pattern recognition algorithms for qualitative and quantitative applications.

(Pardo and Sberveglieri 2000) has the first presentation of VC statistical learning theory (SLT). In particular, the procedures adopted in neural network (NN) applications to avoid overfitting are explained in the general framework of SLT. The first contribution on SVM in artificial olfaction is (Pardo and Sberveglieri 2002). This conference paper appeared in an extended version in (Pardo and Sberveglieri 2005) (see later). Shortly after, Distanti et al. reported work concerned with the use of several SVMs in a “one-against-all” framework, in order to distinguish among volatiles generated by seven different VOCs

(Pentanone, Hexanal, Acetone, Water and three mixtures) by using a MOX sensor array response (Distante, Ancona et al. 2003). Model selection affected only the C parameter (they used a fixed second order polynomial kernel) and was carried out with the widely accepted leave-one-out technique on the training set. From a graph they present (their Figure. 7), it seems that their optimal C values can be found between 10 and 200, depending on the particular class related SVM. They also report the results of a simple performance comparison with traditional NN approach showing a definite SVM prominence for that particular problem. Yet, it is not shown how they dealt with NN model selection and local minima problems.

Brudzewski et al. applied SVMs to a typical food industry problem where the task was to discriminate several samples of milks, they used the ‘one-against-one’ approach. C parameter selection ($C=100$) was only carried out empirically by means of preliminary experiments (Brudzewski, Osowski et al. 2004). They assessed the performance of two kernels, linear (i.e. first order polynomial) and radial basis functions (RBF), yet they did not deal with the choice of the kernel parameter for the RBF kernel. Their classifiers reached 100% correct test classification over two different levels of difficulty datasets. PCA plots indeed showed that the distance between clusters changes for the two difficulty levels, however, in both cases the clusters are well separated on the 2D PCA plane. Interestingly, they also show that the problem’s difficulty level difference still has a role to play in the cardinality of the support vector set for the trained machines with their easier problem resulting in a smaller number of support vectors.

Pardo et al. described SVM application to two ‘two-class’ problems involving real data. In this work, the authors dealt with the model selection problem involving simultaneously the selection of optimal feature vector, C and kernel parameter, highlighting to the EN community, the complex relationships among SVM parameters, feature selection and the overall performances (Pardo and Sberveglieri 2005).

In a similar manner, Acevedo et al. dealt with kernel parameter selection using ant colony optimization (ACO) algorithm, a meta-heuristic search method based on swarm intelligence. Their work was concerned with three different two class EN problems (Acevedo, Maldonado et al. 2006). The datasets were obtained in their labs with a thermo-modulated MOX sensor array.

In 2006, Brudzewski et al., used SVMs in a ‘12 class EN based supplemented gasoline discrimination’ problem (Brudzewski, Osowski et al. 2006). This was a

computationally intensive one-against-one problem to solve which they needed to train 66 SVMs. They compared the performance of three different fixed parameter kernels (Linear, Rank 2 Polynomial and $\gamma = 28$ Gaussian) and obtained 100% classification performance with all of them. They show how the kernel choice can influence the total number of support vector needed. For this experiment the smallest number of unique support vectors (24) was needed by the 66 polynomial machines. C selection was again carried out empirically on one out of five datasubsets while performance estimation was carried out with a 5-fold cross validation (CV) scheme.

In an effort to overcome SVM limitations regarding classes a posteriori probability for novelty detection purposes, they proposed the use of Mahalanobis distance to set a rejection threshold for undetermined class samples.

Acevedo et al. applied a probabilistic approach to solve SVM multiclass problems provided by Platt in an alcohol discrimination framework (Acevedo, Maldonado et al. 2007). Platt tried to estimate class a posteriori probabilities $p(t(x_n) = 1 | y(x_n))$ as a sigmoidal function from the SVMs output (Platt 2000). The work was carried out by reserving a data subset to collect $y(x_n)$ and then applying a CV scheme. In this way, the authors were able to correctly solve the multiclass problem by using a one-against-all approach in which they compare the class a posteriori probability expressed by the eventually colliding classifiers. As a direct consequence, they minimized the memory footprint of the overall classifier; this being ultimately set by the number of support vector. Hence enhancing SVM applicability on embedded platforms. A simple performance comparison is also carried out showing the advantages SVM has over traditional NN and fuzzy ARTMAP schemes. Recently El Barbri et al. applied SVMs to another food industry problem: freshness assessment of sardines. They used the one-against-one approach with a 5 classes problem each one related to a particular aging of samples, however they proceeded by first arbitrarily fixing the kernel and then selecting, via the leave-one-out procedure, the best performing C value (El Barbri, Llobet et al. 2007)

Little effort has been directed at the application of SVMs to EN regression problems. Al-Khalifa et al. used SVMs to evaluate the concentration of CO and NO₂ in four different binary mixtures by using the wavelet coefficient of the response of a thermally modulated MOX sensor (Al-Khalifa, Maldonado et al. 2003). They reached a relative error of less than 5% on both analyte concentration estimations. In a preliminary one-against-all classification stage, they demonstrate the SVM based feature selection process by neglecting all the

features resulting orthogonal to the optimal separating hyperplane. They did not deal with model selection. More recently, De Vito et al. have applied SVMs to the dynamic analysis of a room temperature operating sensor array response to ternary mixtures, the objective being to carry out a real time estimation of the concentration of the analytes (De Vito, Castaldo et al. 2007). A tapped delay (TD) line has been coupled to the SVM in order to take into account the feature vector dynamics and the resulting TD-SVM has been optimized for C and kernel parameters; it reached an interesting squared correlation coefficient (SCC) of 0.87 for NH_3 real time quantification task. The performance comparison with conventional back propagation networks, optimized for hidden neurons and training epoch number, resulted in the SVM obtaining a slight margin over the conventional architecture for the mean absolute error to concentration range ratio.

4.3 ELECTRONIC NOSE APPLICATIONS

In this section we apply SVM to two application examples, the first for a classification and the second for a regression problem. We show how HP tuning permits one to explore the insurgence of underfitting and overfitting effects, which are the principal limitations of non parametric learning techniques.

4.3.1 Classification with SVM

In (Pardo and Sberveglieri, 2005) we analyzed two binary classification problems of different levels of difficulty with SVM. As mentioned, the binary problem is the simplest possible classification problem and constitutes the basic unit into which any N-class problem can be decomposed. The advantage of having only two classes are (1) that there is a larger number of patterns per class, which increases the significance of the results, and (2) that the results are independent of the particular decomposition scheme, which could be used to treat the original seven class problem.

The two two-class datasets have been constructed by unifying seven classes into two super-classes in two different ways. The seven classes are concerned with measurements performed on seven coffee blends with the Pico-2 EN (Pardo and Sberveglieri 2004), developed at the Sensor Lab in Brescia, which makes use of five thin film semiconductor sensors.

We investigate the performance of SVMs with regard to:

(1) The number of principal components (PC), which are given as inputs to the SVM. These range from two to five, five being the number of sensors used in the EN. By varying the number of PC we wanted to see if PCs accounting for very small variance ($\sim 1\%$) also have an influence on the classification performance as we noticed for MLP (Pardo and Sberveglieri 2002b).

(2) The type of kernel: we tried both polynomial and RBFs or Gaussian kernels. For the polynomial, we scanned the polynomial order from 1 (linear machine) to 8, for the RBF we scanned the standard deviation parameter from 0.1 to 10 (at unequal steps, concentrating on small values).

(3) The regularization parameter C of Eq.(13), which controls the trade off between low training error and large margin.

The kernel parameters and the value of the constant C determine the complexity of the SVM solution and hence its generalization ability. E.g., for the Gaussian kernel, a small standard deviation σ means that the separation surface is complex and tailored to the training data, possibly producing overfitting. The same holds for a high order of the polynomial kernel.

The performance (test error) for each parameter combination was measured by 4 fold CV.

We display the results for the polynomial kernels, where the kernel parameter is the polynomial order; for the RBF kernels, where the kernel parameter is the standard deviation of the Gaussian function, we refer the reader to (Pardo and Sberveglieri 2005). In Figure 1 and Figure 2 the misclassification rate is evaluated on a grid of two independent variables, the number of PC on which the data are projected and the value of the kernel parameter (C is set to $C=1$). The difficult (classification) problem and the easy problem are termed DP and EP respectively. We observe that:

The error surfaces show an increase for polynomial order greater than or equal to three. The minimum is obtained, for both problems, for the order two. After that, overfitting sets in, because the bigger the polynomial order the more complex the kernel function. The difficulty of the problem is reflected by the absolute values of the error. For the DP, the error surface stays above the 10% threshold, while for the EP the smallest error, attained for 5 PC and polynomial order 2, is 0%. Conversely, the worst results are circa 50% correct classification (for a two class problem this amounts to chance) for the DP, while they stay under 30% for the EP.

Moreover, for the DP, the increase of misclassification error with polynomial order is independent of the number of PC. On the other hand, for the EP, the

error increases more for a low PC number. This behaviour has been noticed for MLP also (Pardo, Sberveglieri et al. 2001), and can be explained as follows. For the EP PCs with little variance contain valuable information, i.e. - by definition - information which leads to an improved classification. Therefore these data can make use of a more flexible classification function. For the DP, on the other hand, PCs with little variance just contains noise. Therefore complex classification functions end up just modeling that noise, and hence their generalization performance is poor.

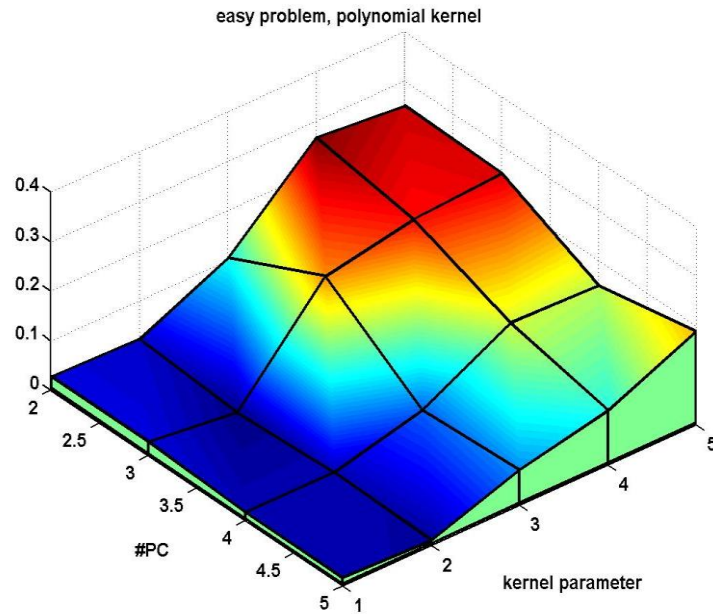


Figure 1: Error surface for the easy classification problem as a function of the number of principal components (#PC) - on which the data are projected - and the polynomial order (the kernel parameter).

For RBF (not shown), there is a difference compared with the polynomial kernel. The RBF kernel presents a degrading performance for large standard deviation parameters (which means a more smooth decision function), hinting at underfitting.

In Figure 3, we plot the error surface for a fixed number of four PCs whilst varying the complexity coefficient C , for RBF. It is evident that using $C=1$ - the standard choice and the one that had been adopted in the previous plots - the

performance is suboptimal. The minimum of the error surface is attained around $C=300$.

While this particular minimum point may well be an artefact of the relatively loose parameter grid tested, the sharp lowering of the error with respect to $C=1$ is evident. We also notice two other interesting qualitative trends:

(1) For $C=1$ there is a marked underfitting effect, the error gets bigger as the kernel parameter σ increases. This does not happen for higher C values; certainly for $C>100$. C values between 0 and 100 were not investigated.

(2) For $C>100$, the graph shows a clear, if not very intense overfitting for small σ values (the smallest σ we tried is $\sigma=1$ for this graph). The best σ seems to be $\sigma=10$ and gives an error comparable to that for the EP.

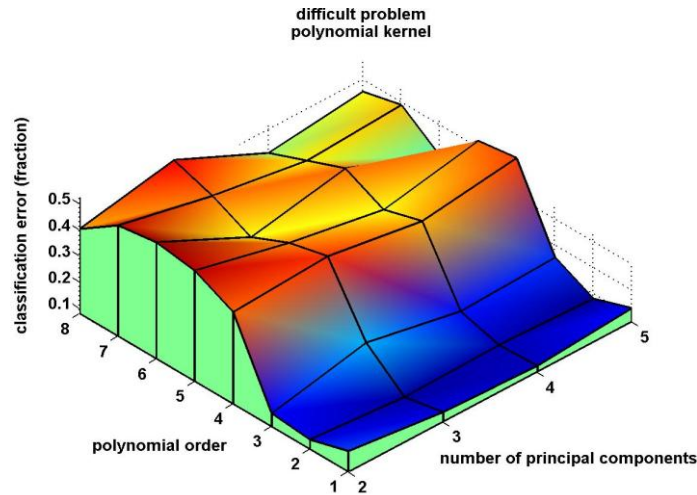


Figure 2: Error surface for the difficult classification problem as a function of the number of principal components and the polynomial order.

To determine the performance it was necessary to evaluate each HP configuration during the preceding optimization and results were calculated using four fold CV. After HP selection, the error for the SVM with the fixed HP configuration needs to be evaluated on a separate dataset, not used until this point, otherwise the results may be subject to biased error estimation; which is common practice for feature selection (Ambroise and McLachlan 2002).

Nested CV is therefore an option. Yet, CV is a time consuming process: for a given time, which is externally constrained by the computational power at hand, one can either search over a bigger HP space (or equivalently over a finer grid) or

increase the number of folds. It may therefore be interesting to investigate the dependence of the CV error on the number of CV folds, both for the internal folds used at the HP selection stage and for the external fold used at the final error estimation stage. This has been recently performed in (Pardo and Sberveglieri 2008), where we found that: (1) there is a slight dependence on the number of external CV folds, while the number of inner CV folds seems to be immaterial; (2) 2x2 nested CV is often enough for a good result. With respect to 10x10 CV, 2x2 CV requires 4% of the training time, so this result may save quite a bit of time in future computational studies.

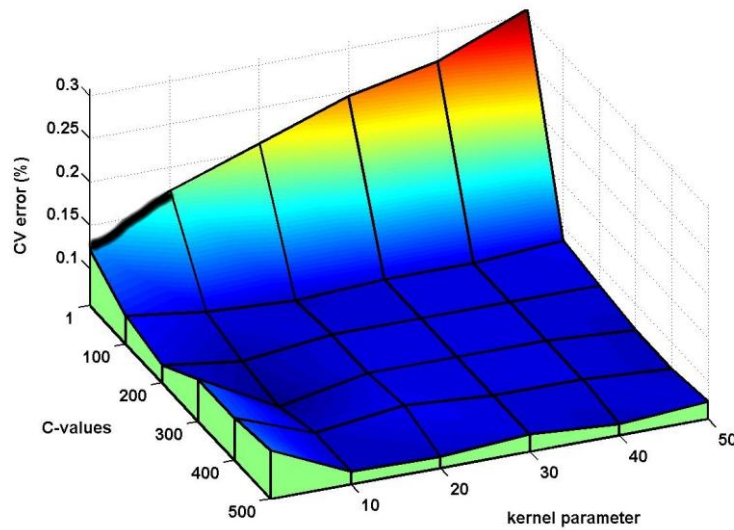


Figure 3: Error surface for the difficult classification problem as a function of the parameter controlling the complexity of the SVM (C-value) and the standard deviation in the Gaussian kernel.

In the same paper we compared the classification performance of Random Forests (RF), SVM and Nearest Shrunken Centroids (NSC). We found that RF and SVM have a similar classification performance, while NSC performed the worst. On the other hand, RF and NSC have an in-built feature selection mechanism that is very helpful for understanding the structure of the dataset and evaluating sensors.

4.3.2 Regression with SVM

In (De Vito, Castaldo et al. 2007), we applied a Gaussian RBF kernel ($K(x,x')=\exp(-\gamma \|x-x'\|^2)$) based TD ε -SVR to a ‘ternary mixture single analyte real time quantification problem’; note that here we use the notation $\gamma=1/\sigma^2$, where σ is the standard deviation, as used in the previous classification section. We addressed C and kernel parameter optimization after an empirical search for best length and density (i.e. the sampling period) of their TD line. The latter can also be considered as a constrained feature selection problem. However, the complete optimization of such a system requires the simultaneous optimization of all parameters otherwise suboptimal results are very likely to be obtained.

One way to proceed is to perform a grid search using a brute force approach that can help to shed a light on issues like the stability of SVR optimal kernel and cost parameters with respect to the feature vector composition. This can be a practical solution if the number of parameters remains low or if significant computing power is available. Alternatively, this multiple optimization problem can be tackled with different heuristic approaches (see for example Huyer and Neumaier 1999) but they often require preliminary knowledge about the relationships between the parameters.

Here, we describe the results of applying the brute force approach to the simultaneous optimization of the parameters for the setup described in (De Vito, Castaldo et al. 2007). Briefly, three RBF based TD ε -SVR are designed to continuously estimate by using a five room temperature operating chemiresistor array response, the concentration values of analytes and relative humidity (RH) levels in several NH_3 - NO_2 gas mixtures. The dataset was constructed by exposing the array to 64 different mixtures; obtained by mixing the three above mentioned species at different concentrations (see Table 1), exposure cycle duration was 60 minutes while washing time was 90 minutes. Conductivity was sampled at the sampling frequency of 2 samples/min, no correction for base resistance was performed each cycle. Estimation was meant to be carried out dynamically at each sample time (real time continuous estimation) both in exposure and washing time.

Table 1: Concentration values used to mix species in 64 different mixtures.

NH_3	NO_2	RH
[0,10,15,20] ppm	[0,300,600,900] ppb	[30,40,50,60] %

Cost parameter C and RBF γ HPs were left as free parameters to be optimised. TD sample length N and distance D between the first and last sample with which to feed the SVM are the two additional free parameters with each point in this 4-dimensional parameter space characterizing different dynamic regression models (see Figure 4). As an example, setting N and D respectively to 5 and 20 results in a feature vector comprising the sensor array response at current time t_0 and four past array responses sampled at $t_0 - kT$ where T is the sampling period and k belongs to $[5 \ 10 \ 15 \ 20]$.

Initially, we have partitioned the 64 exposure cycle dataset by reserving 51 cycles for training purposes and leaving the remaining for testing purposes. A 5-fold CV scheme was used in order to obtain reliable performance estimation. Squared Correlation Coefficient (SCC) was selected as the performance estimator for the different regression models.

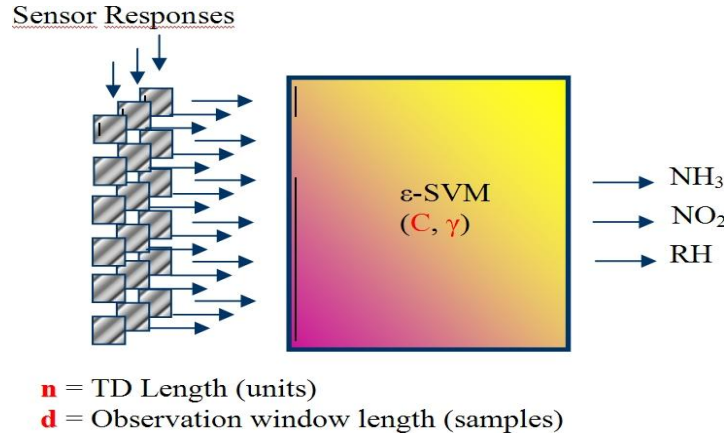


Figure 4: The proposed tapped delay based sensor fusion architecture.

In a preliminary run, we tried to optimise the (C, γ) couple and the (N, D) couple separately. We set an arbitrary value for the (N, D) couple choosing $N=10$ and $D=10$, that is letting the SVM evaluate the concentration of the three compounds by analyzing the current value of each sensor response together with the nine consecutive past values (about 5 minutes observation window length). We then optimized the value of C and γ by means of the Multilevel Coordinate Search (MCS) (Huyer and Neumaier 1999); an algorithm which uses SCC between expected and guessed value as the performance estimation index. By

using the C and γ values obtained in the latter step, we have then proceeded to evaluate the best (N,D) couple by exploring a $[2,10] \times [4,300]$ values matrix.

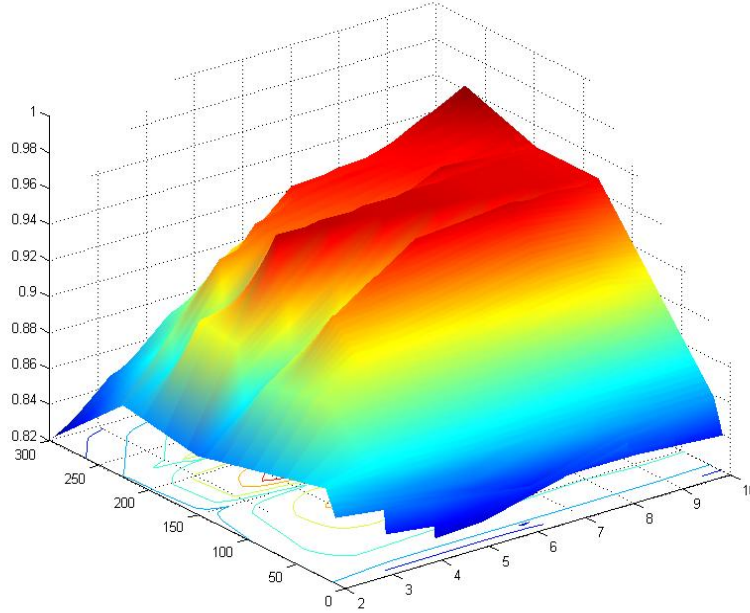


Figure 5: Best SCC values surface and pseudo-color plot obtained for each N,D couple searching through all the selected C,γ space. Highest N,D values scores best performance but interesting performance can be obtained with 60 minutes wide observation window, if adequately sampled.

The results for the NH_3 concentration estimation problem was a SCC, between estimated and ground truth values, greater than 0.92 at $N=10$, $D=180$, when choosing optimal values for (C, γ, N, D) parameters. This accounts for a significant advantage with respect to our previous 2007 work where (N,D) were fixed ($\text{SCC} = 0.87$). As expected, the SCC value grows while N and D approach their maximum. For a comparison we then tackled the brute force approach to perform the simultaneous optimization of the entire set of 4 HPs; actually, if the optimal C and γ values vary significantly with the chosen N,D couple, the optimisation procedure reported above will lead to sub-optimal overall performance. Performance estimation was conducted using a sort of ‘leave-one-cycle-out’ methodology i.e. using 63 complete exposure cycles for training, leaving the remaining cycle for testing purposes and performing a 64-fold CV run. Of course, this brute force approach was definitely computationally

expensive. In order to explore a $5 \times 5 \times 6 \times 4$ $[N, D, C, \gamma]$ (C, γ varied in $[1, 50] \times [1, 60]$) sampled parameter space for all three analytes, we had to perform the training of 115200 (3 analytes $\times$ 64 exposure cycles $\times$ 25 N, D matrix points $\times$ 24 C, γ matrix points) total SVMs. Trainings ran on a 18900 samples dataset (300 sample at a cycle) using feature vector dimensions ranging from 10 to 50. Computation was carried out using the ENEA (Italian agency for new technologies, environment and renewable energies) computing grid facilities, based primarily on AFS (Andrew File system) and the LSF job scheduler. In specific terms, a computing cluster, based on IBM p6xx AIX machines powered by 16 1.1 GHz CPUs (4 GFlops per CPU) was in use for more than one week.

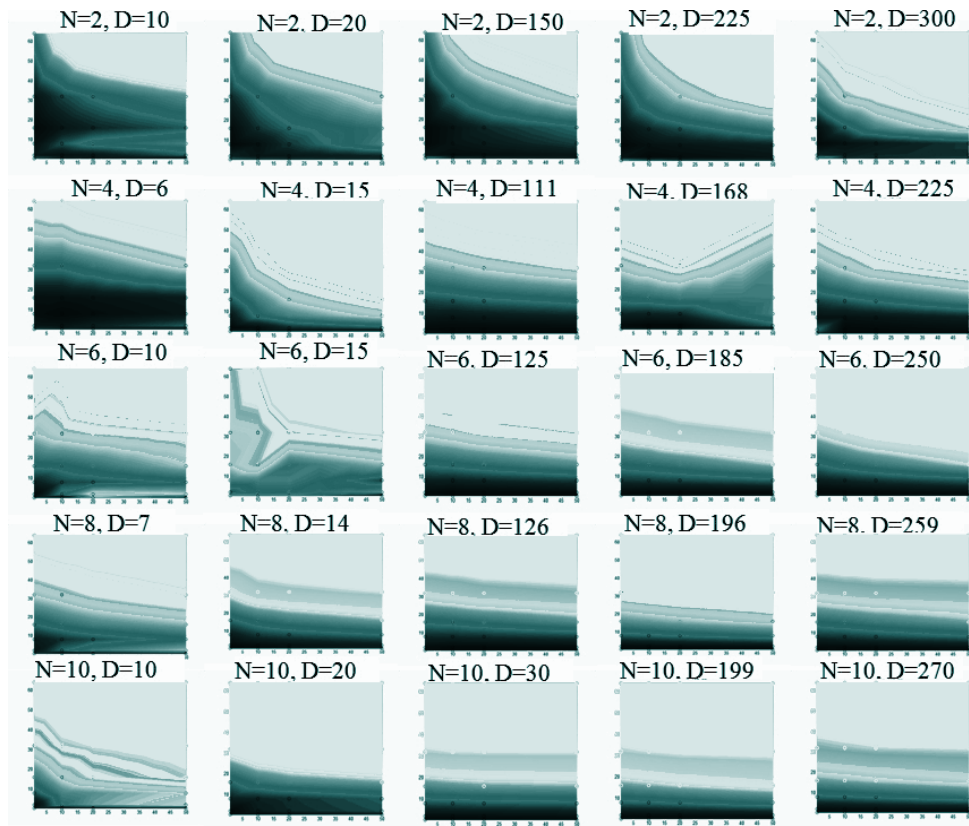


Figure 6: SCC performances of C (y-axis), γ (x-axis) dependent SVM architectures computed on the samples matrix (circles) for different N, D couples. Low C, γ values guarantee the optimal performance in most of the cases but C value became more critical at high N, D values.

Here, we report the results for the NH_3 estimation problem. In particular Figure 6 is concerned with the best SCC values obtained for all C, γ parameters values for each different TD-SVM architecture characterized by a point in the N, D sampled space. At a first glance, the SCC values reach their best while N, D both approach their maximum values in the selected grid. However, it is very interesting to note that it is possible to obtain significantly good results by limiting the observation window length to less than 60 minutes while using only 6 samples for each sensor channel.

The MAE/Range parameter (mean absolute error / concentrations range) was found to reach 3% while in (De Vito, Castaldo et al. 2007) it was 6.6%. Best SCC scores exceeded 0.96. Hence, as regard to the previously described approach, the SCC advantage is set to more than 0.04; the previously chosen N, D couple was, hence, suboptimal.

A closer look reveals also that the relationship between N and D is quite interesting, the descriptive power of the feature vector using N samples initially increase with the observation length until a maximum after which it seems to slightly fade. This agrees with a simple interpretation, i.e. a small observation window retains little information on the dynamics of sensors but a large window requires an adequate number of sampling points in order to correctly capture the optimal description power.

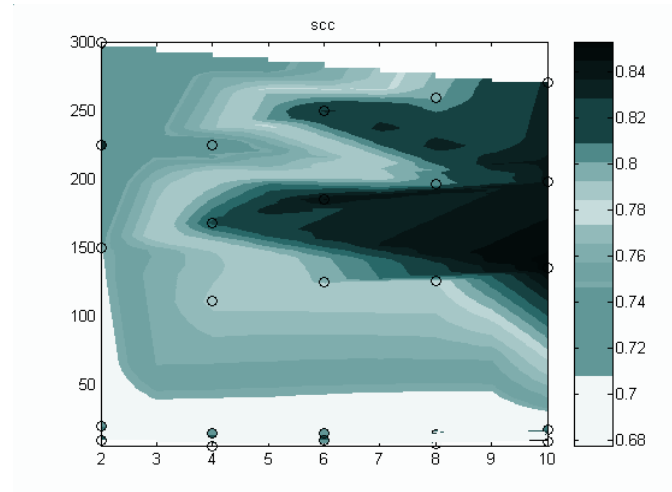


Figure 7: Best RH estimation SCC values pseudo-color plot obtained for each N, D couple searching through the C, γ space. Circles depict some of the actual performance evaluation point corresponding to different TD architectures.

In Figure 6 we depict the SCC values against C and γ values for each of the analyzed N,D couples. Optimal C and γ values varies between the different N,D couples although it is possible to recognize a region where quasi-optimal performance can be obtained whichever couple is chosen.

This performance stability in the C, γ space with respect to N,D is the cause of the closeness in the results obtained by the two optimization procedures. The problem optimization seems generally to require the use of low C and/or γ values, this accounts for using relatively simple solution structure (the lower the γ value the smoother the kernel) or setting low penalties for empirical error otherwise overfitting sets in. It is worth noting that a low C value becomes more critical at high N and D values, where best results are obtained, while γ can be chosen with increasing freedom.

Again, regarding the preliminary experiment, if we initially relied on a different $[N,D]$ couple, e.g. $N=4$ $D=6$, we would have selected a different C, γ couple to start with, this would probably have led us to a different best N,D couple, i.e. the one that reaches the best score with $C=10, \gamma=10$ choice obtaining sub optimal performances.

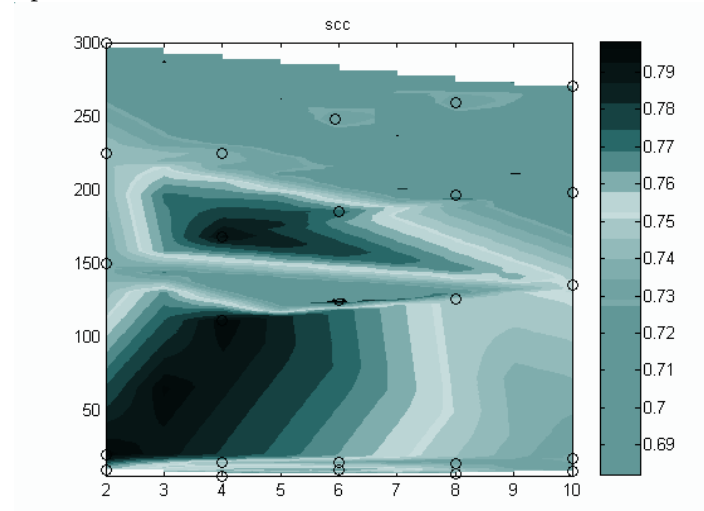


Figure 8: Best RH estimation SCC values pseudo-color plot obtained for each N,D couple searching through the C, γ space. Circles depict some of the actual performance evaluation points corresponding to different TD architectures.

Finally, we report the best values of SCC computed on the N,D space for RH and NO_2 concentration estimation problem in Figure 7 and 8. The RH estimation problem expresses a very similar behavior with respect to that for NH_3

estimation, with the best overall scores being obtained at high N,D values even if the best SCC scores are slightly worse. The NO₂ estimation problem gave significantly worse SCC figures and above all expresses a different behavior in which good SCC scores are obtained even at low N,D values. This agrees with the possibility that the most information is embedded in the short term sensor dynamics with long term dynamics giving no significant contribution to NO₂ concentration estimation. Furthermore, it is likely that the lack of an adequate number of samples describing short term dynamics for high D values causes the poor SCC scores that the proposed TD-SVR expresses in the top right region of Figure 8.

4.4 CONCLUSIONS

In this chapter we presented SVM both for classification and for regression purposes. SVM, developed more than 10 years ago, are still not very widespread in the artificial olfaction community. Yet they both have good theoretical underpinning in statistical learning theory and showed top learning performances in benchmark studies, also in artificial olfaction.

In the application section we gave the main weight to hyperparameter selection, using several different experimental datasets.

We further highlighted that SVM compares favourably with other learning machines, including neural networks and random forests. We saw that if SVM are used after feature extraction with principal component, even principal components carrying small variance; have an impact on SVM performance, just as we observed for multilayer perceptrons in the past. Moreover, the parameter controlling the trade-off between high margin and low error (in the criterion function) plays a major role in the resulting test error. In our case a value of $C \sim 300$ gives optimal results for the difficult classification problem.

For regression, we explored the relationships between model parameters for a tapped delay support vector machine showing how inadequate (separate) hyperparameter and feature selection procedures could eventually lead to suboptimal performances. In this case study, we also show the insurgence of overfitting problems at high C values. As regards to the TD line parameters, for NH₃ concentration estimation, we found that a 60 minutes wide observation window using 5 equally spaced samples plus current sensor response is sufficient in order to obtain significant performance improvements with respect to our previous work. Finally, we show that wide observation windows makes it

necessary to have adequately dense sample spacing in order to allow the TD line to correctly capture and describe the room temperature operating sensor array dynamics.

References

- Acevedo, F. J., S. Maldonado et al. (2006). Model Selection for Support Vector machines Using Ant Colony Optimization in an Electronic Nose Application. ANTS 2006, LNCS 4150, pp. 468-475. Springer-Verlag.
- Acevedo, F. J., S. Maldonado et al. (2007). Probabilistic support vector machines for multiclass alcohol identification. *Sensors and Actuators B-Chemical*, 122, pp.227-245.
- Al-Khalifa, S., S. Maldonado et al. (2003). Identification of CO and NO₂ using a thermally resistive microsensor and support vector machine, *IEE Proc. Sci. Meas. Technol.* 150(6), pp. 11-14.
- Ambrose, C. and G. J. McLachlan (2002). Selection bias in gene extraction on the basis of microarray gene-expression data,; 6562–6566 *PNAS* May 14, vol. 99 no. 10
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer.
- Brudzewski, K., S. Osowski et al. (2004). Classification of milk by means of an electronic nose and svm neural network. *Sensors and Actuators B-Chemical*, 98, pp.291-298.
- Brudzewski, K., S. Osowski et al. (2006). Classification of gasoline with supplement of bio-products by means of an electronic nose and SVM neural network. *Sensors and Actuators B-Chemical*, 113, pp.135-141.
- Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), pp. 121-167.
- Cristianini, N. and J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines*, Cambridge University Press.
- De Vito, S., A. Castaldo et al. (2007). Gas concentration estimation in ternary mixture with room temperature operating sensor array with tapped delay architectures. *Sensors and Actuators B-Chemical*, 124, pp.309-316.
- Distante, C., N. Ancona et al. (2003). Support vector machines for olfactory signals recognition. *Sensors and Actuators B-Chemical*, 88, pp.30-39.
- Drucker, H, J.C. Burges et al. (1996). Support Vector Regression Machines. *Advances in Neural Information Processing Systems*, 9, NIPS 1996, pp. 155-161.
- El Barbri, N., E. Llobet et al., (2007). Application of a portable electronic nose system to assess the freshness of Moroccan sardines, to appear in *Material Science and Engineering* :C (doi:10.1016/j.msec.2007.10.056).

- Guyon, I., <http://www.clopinet.com/isabelle/Projects/SVM/applist.html>.
- Huyer, W. and A. Neumaier (1999). Global Optimization by Multilevel Coordinate Search, *Journal of Global Optimization*, 14, pp. 331--355.
- Hsu, C. W. and C. J. Lin (2002). A comparison on methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*.
- Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations, *Philosophical Transactions of Royal Society, London, A* 209, pp. 415-446.
- Pardo, M. and G. Sberveglieri et al. (2001). Decompositional classification models for electronic noses. *Analytica Chimica Acta*, 446, pp.223-232.
- Pardo, M. and G. Sberveglieri (2002), Support vector machines for the classification of electronic nose data: 8th International Symposium on Chemometrics in Analytical Chemistry. Seattle.
- Pardo, M. and G. Sberveglieri (2002b). Coffee analysis with an electronic nose. *IEEE Transactions on Instrumentation and Measurement*, 51, pp. 1334-1339.
- Pardo, M. and G. Sberveglieri (2004). Electronic olfactory systems based on metal oxide semiconductor sensor arrays. *Sensors Bulletin*, 29, pp.703-708.
- Pardo, M. and G. Sberveglieri (2005), Classification of electronic nose data with support vector machines. *Sensors and Actuators B-Chemical*, 107, pp.730-737.
- Pardo, M. and G. Sberveglieri (2008). Random Forests and Nearest Shrunken Centroids for the Classification of Sensor Array Data. *Sensors and Actuators B*, 131/1 93-99.
- Persaud, K. C. and G. Dodd (1982). Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose, *Nature*, 299, pp.352-355.
- Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization, in *Advances in Kernel Methods – Support Vector Learning*, MIT Press, pp. 185-208.
- Platt, J. C. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods in *Advances in Large Margin Classifiers*, MIT Press, pp. 125-143.
- Tipping, M. E. (2001). Sparse Bayesian learning and relevance vector machine. *Journal of Machine Learning Research*, 1, pp. 211-214.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.
- Vapnik, V. N. (1998). *Statistical learning theory*. Wiley, New York.

Chapter 5

Dynamic Methods for Improving the Performance of Semiconductor Gas Sensors

Alexander Vergara, Xavier Vilanova and Eduard Llobet

5.1 INTRODUCTION

Semiconductor oxide gas sensors are effectively gas sensitive resistors. The sensing element generally comprises a semiconductor material with a high surface-to-bulk ratio, which is deposited onto a heated insulating substrate between two metallic electrodes. Reactions involving gas molecules can take place at the semiconductor surface to change the density of the available carriers. Thus, the atmospheric composition affects the conductance of the device (Moseley et al. 1987, 1991). Semiconductor gas sensors were first commercialised in Japan (Taguchi 1971) using tin oxide as the sensitive component but a broad range of materials is now used. Oxides are the most common materials for sensors because they make molecular chemistry reactions possible on the surface layer but the electrical consequences of which are manifest in a considerable volume of the solid. However, since any gas that can react on the sensor surface, can alter the sensor conductance, semiconductor oxide gas sensors are in general poorly-selective.

In recent years, a great deal of work has been done to overcome the problem of selectivity. In this section, new strategies that can help to improve it are reviewed.

5.1.1 Basic principles of detection

Semiconductor materials that detect small concentrations of reactive gases in air rely on changes in surface conductance and detect a change from the equilibrium condition induced by the gas. Surface conductivity semiconductor gas sensors are used to detect minor concentrations of potentially hazardous gases and are normally operated at temperatures between 300-500°C. The central surface reaction responsible for most gas responses involves changes in the concentration of surface oxygen species such as O^{2-} (Göpel 1985). The formation of such ions means that the oxygen adsorbed at the gas/solid interface abstracts electrons from the solid bulk. In the case of an *n*-type semiconductor such as SnO_2 , the electrons come from ionised donors via the conduction band, the charge carrier density at the interface is reduced and a potential barrier to charge transport, $\Delta\phi$, develops. At the junctions between the grains of the solid (see Figure 1), the depletion layer and the associated potential barrier are responsible for high resistance contacts which dominate the resistance of the sensor.

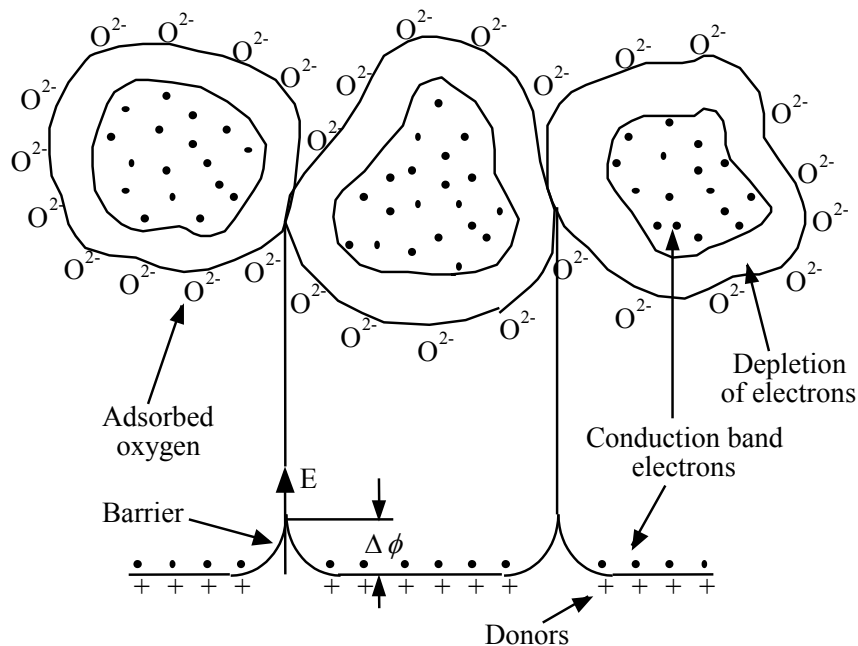


Figure 1: Barriers at intergranular contacts on a pellet. Three grains with adsorbed oxygen providing surface depletion layers. The depleted layers are responsible for a high contact resistance. For conduction, electrons must cross over the surface barriers.

Thus, at temperatures between 300-500°C, semiconductor oxides can be used as gas sensitive resistors to monitor those gases that react in such a way as to alter the quantity of charge trapped at the surface.

The surface barrier height, $\Delta\phi$, the depletion layer width, W_{sc} , the charge, Q_s , associated with oxygen surface coverage, and donor density N_D are related as follows:

$$W_{sc} = Q_s / qN_D \quad (1)$$

$$\Delta\phi = Q_s^2 / (2\epsilon_o\epsilon_r qN_D) \quad (2)$$

In the case of a p-type oxide, adsorbed oxygen acts as a surface acceptor state, abstracting electrons from the valence band and hence giving rise to an increase in the charge-carrier (holes) concentration.

5.1.2 Fabrication of semiconductor oxide gas sensors

The different fabrication technologies of semiconductor gas sensors use different processes to achieve long term stability under normal operating conditions.

- (1) Ceramic sensors: Most commercially available devices are produced by a typical ceramic fabrication process (Murakami et al. 1983, Watson et al. 1985). For long-term stability they are sintered for a long time under controlled gas and temperature conditions.
- (2) Thick-film sensors: Structured semiconductor oxide sensors may be produced in thick-film technology. The most widespread method of preparing thick films in electronic and sensor applications is screen printing. (Prudenziati 1991).
- (3) Thin film sensors: Compared with the fabrication processes described above, the fabrication of thin-film structures is based on well-established processes under clean-room conditions using different substrates such as Si wafers covered with thermal oxides or nitrides.

Of particular importance for low-power consumption devices and to reduce the thermal inertia of gas sensors is the introduction of the so called micro-hotplates. These substrates are suspended membranes to insulate the heated part of the sensor (where the active layer is placed jointly with the heating element and electrodes) from the bulk of the substrate (Möller et al. 1995, Corcoran et al. 1993).

5.1.3 The selectivity problem in semiconductor oxide gas sensors

Selectivity is the ability of a sensor to measure one measurand (e.g. a chemical component) in the presence of others. There are certain areas of sensing where the problem of cross-sensitivities is of major importance. This is especially true when sensors are used to monitor different target gases in air, or when the target gas is in the presence of an interfering species. Moreover, gas sensors are very often influenced by water vapour, so changes in the moisture content of the atmosphere monitored interfere with gas sensing.

There are several different ways of enhancing the selectivity of semiconductor oxide sensors, which can be classified into three main categories:

- (1) The search for new materials.
- (2) The use of sensor arrays and appropriate signal processing techniques.
- (3) The dynamical characterisation of the sensor response.

Important findings have been reported in this area, some of which are reviewed in the paragraphs below. Additionally, these techniques have been used not only for resistive gas sensors (e.g. metal oxides), but also for other type of devices such as gravimetric gas sensors.

5.1.3.1 The search for new materials

As tin oxide sensors have been found to suffer from a lack of selectivity, considerable efforts have been made in the search for new oxide materials with more specific responses (Moseley et al. 1991, Meixner et al. 1996). A wide range of materials has been investigated for use in the detection of minority gases in air (Moseley et al. 1992). Doping of the oxide structure with alternative metal ions also allow electronic properties to be adjusted (Williams et al. 1991).

5.1.3.2 Sensor arrays and signal processing

Sensor arrays which have sensing elements with partially overlapped sensitivities and pattern recognition techniques have been applied with some degree of success. Each sensor in the array defines an axis in a multidimensional space where gases can be represented as points positioned in this space, according to sensor responses. Gases or mixtures that produce similar responses from the sensor array will tend to cluster near one another in space. Pattern recognition

uses multivariate statistics and numerical analysis to investigate such clustering and to elucidate relationships in multidimensional data sets (Gardner et al. 1994, Di Natale et al. 1995).

A large number of interesting applications have been reported in recent years, such as the works on the detection of pollutants (Rose-Pehrsson et al. 1988, Gall 1993, Di Natale et al. 1994, Hierlemann et al. 1995, Moore et al. 1993) or those performed on aroma analysis employing the electronic nose (EN) (Shurmer et al. 1989, Gardner et al. 1992, Gardner et al. 1991, Pearce et al. 1993, Di Natale et al. 1996) and to assess quality and freshness of food (Schweizer-Berberich et al. 1994, Bourrounet et al. 1995, Barbri et al. 2007).

5.1.3.3 Dynamically characterising the sensor response

The aim of this strategy is to obtain additional information that can be combined with steady-state information in order to enhance sensor selectivity. The main approaches are the following:

- (1) Modulating the operating temperature of sensors: With properly programmed temperature changes, characteristic response patterns associated with the detected gases can be obtained (Clifford et al. 1983, Wlodet et al. 1991). Cycling the temperature, enables the whole range of gas reactions in different regions of the heating or cooling cycle to be detected, and also prevents the surface contamination that occurs after being exposed for too long at low temperatures (Sears et al. 1990). More recently, millisecond time-scale temperature changes have been used to modify the rates of adsorption, desorption and reaction on the sensing surface during sensor operation (Cavichi et al. 1995, Cavichi et al. 1996).
- (2) Measuring in A.C. mode: The plots of the real and imaginary components of the admittance versus frequency provide information on the different parameters that play a role in the sensor conduction mechanisms when metal oxides (Gutierrez et al. 1991 and 1992) or conducting polymers (Amrani et al. 1996), are used.
- (3) Modulating the concentration of the measured gases and analysing the sensor transient behaviour: This has led to new parameters being identified which are characteristic of each gas/sensor pair. Some of these parameters can be concentration independent (Vilanova et al. 1996, Llobet et al. 1995) even in gas mixtures (Llobet et al. 1997). In the last few years, the transient response of different chemical sensors (i.e. thin and thick-film metal oxide and polymer-based sensors) to a step-change in the gas concentration has been

analysed (Vaihinger et al. 1991, Samitier et al. 1994, Buttá et al. 1990, Gardner 1990, Nanto et al. 1996, Nakamura et al. 1996, Setkus et al. 2008, Nimsuk et al. 2007, Zhang et al. 2007).

5.2 NEW DEVELOPMENTS FOR EXPLOITING SENSOR DYNAMICS

In recent years a great deal of work has been done in the field of temperature modulated semiconductor gas sensors, undeniably one of the most promising approaches for taking advantage of sensor dynamics. From the late nineties onward, different methods to extract and process features from temperature modulated gas sensors have been introduced. The most successful techniques include the use of time domain methods (e.g. window time slicing), frequency domain methods (e.g. Fourier transform) and mixed methods (e.g. discrete wavelet transform or short-term Fourier transform) to extract features from the response transients. All these methods have the advantage of performing feature extraction and data compression. Furthermore, if a careful selection of the features to be input into pattern recognition engines is performed, the identification of gases with a high tolerance to the presence of drift and noise seems possible. At least, results compare very favourably with those derived from methods that rely on the static sensor response. The reader is referred to (Llobet 2006) for a review on temperature modulated gas sensors (from the early works to those performed in the mid-2000s) and to Chapter 3 of this book for an in-depth discussion about the optimisation of the temperature-modulation frequencies.

Although the dynamic characterisation of sensor response when gas concentration varies is not as widespread as temperature modulation, this method has also been the subject of research in view of ameliorating sensor performance. The usual sample delivery methods such as the static or dynamic headspace imply that, for a given measurement, the response of every sensor is a time profile and its characteristics (shape, duration, etc.) might be informative. It is only very recently that multi-way analysis methods have been introduced so it is no longer necessary to transform the measured time profiles (i.e. sensor transient response) into univariate features (at the risk of losing totally or in part important information from the time domain).

Therefore, the objective of this section is to briefly review recently introduced methods that have been shown to be useful at analysing sensor transients.

5.2.1 Phase space

The nature of the sensing mechanisms ruling of chemical sensors is still far from full comprehension. The extraction of features from the response of chemical sensors consists of the selection of some characteristics of their temporal response sequence, which result from the interaction between sensors and the compounds to be detected. The extracted features are then input to pattern recognition systems. From a general point of view, a chemical sensor can be considered as a dynamic system whose response signal temporally evolves following, with its proper dynamics, the concentration of the analytes. Therefore, any of the currently available tools usually employed to study the properties of dynamic systems may be employed with sensors (Ljung et al. 1994).

The phase space (PS) is a central concept in the analysis of dynamic systems. Given a system whose state is completely described by n scalar variables, different states correspond to different points in a n -dimensional vector space defined by an orthonormal basis where each direction corresponds to one of the scalar variables. There is an unequivocal correspondence between points in the PS and states of the system.

In a mechanical system, the observation of position as a function of time allows for obtaining new variables such as velocity (first derivative) and acceleration (second derivative) from which the phase space is constructed. If a mechanical system was observed via a single scalar quantity for a period of time, then it could be represented by a sequence of points in the phase space. This sequence, after being suitably parametrised with the time, forms a trajectory on the phase space that contains all the information about the phenomena underlying the evolution of the system. The correspondence between observed signals and the new variables forming the phase space has been demonstrated by (Takens 1980) in his embedding theorem. In accordance to this theorem, the variables that form the PS are obtained by differentiable transformations of the observed signal.

If we consider now a sensor to be a system, the information is in the temporal evolution of a single scalar quantity (i.e. in the sensor signal) and from this time dependence, a number of new variables can be generated. The minimum set of variables able to describe the state and the evolution of the sensor comprises the response signal (i.e. the observed variable) and its first derivative. Therefore, a phase space can be defined by the orthonormal basis formed by the observed signal and its first derivative. The use of the phase space in the field of gas sensors was first introduced in (Martinelli et al. 2003). The sensors employed

were quartz crystal micro-balances (QCM) with a selective adsorbent coating. QCM sensors are operated in a closed loop with an oscillator circuit and the observed signal is the shift in oscillation frequency experienced by the system when gases are adsorbed or desorbed from the selective coating. The change in the resonance frequency is directly proportional to the change of mass due to gas adsorption or desorption. Figure 2 shows a typical response of a QCM sensor to an abrupt variation of the chemical content of the air to which it is exposed. The frequency decrease corresponds to an adsorption step (i.e. the mass trapped at the surface of the selective coating increases) while the frequency increase corresponds to the desorption step (i.e. the mass trapped at the surface of the selective coating decreases because the sensor is flushed with clean air).

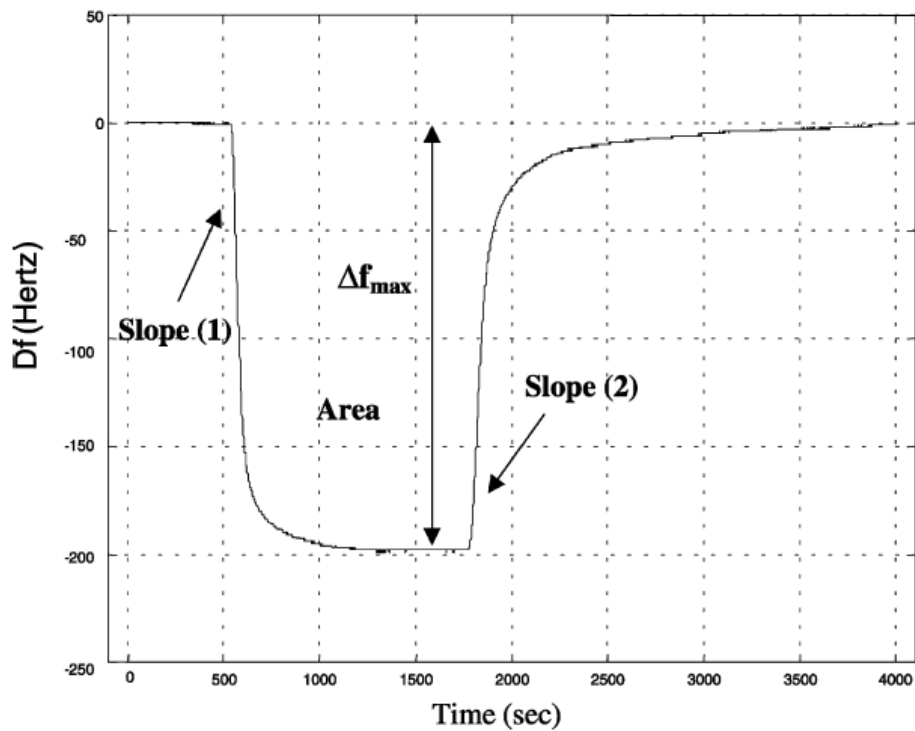


Figure 2: Typical evolution of the frequency signal of a QCM sensor during an adsorption-desorption cycle. Some typical response features such as frequency shift ($\Delta f_{\max}$) and the adsorption and desorption initial slopes are indicated. Reproduced from (Martinelli et al. 2003), Elsevier Science ©, with permission.

The same behaviour shown by Figure 2 can be represented in a PS the coordinates of which are the frequency and its first derivative. Figure 3 shows the trajectory in PS calculated employing the response signal shown in Figure 2. The shape of this trajectory is related to the adsorption-desorption processes occurring in the selective adsorbent coating. If studied in the time domain adsorption models lead to exponential or multi-exponential behaviour (the latter were introduced (Marco et al. 1995) to better fit experimental data). Multi-exponential behaviour give rise, in the PS, to trajectories characterised by several slopes each to one time constant. As a result of the exponential convergence of the observed signal in the time domain, in the phase space the speed of tracking of the trajectory decreases with time until the steady state is reached asymptotically.

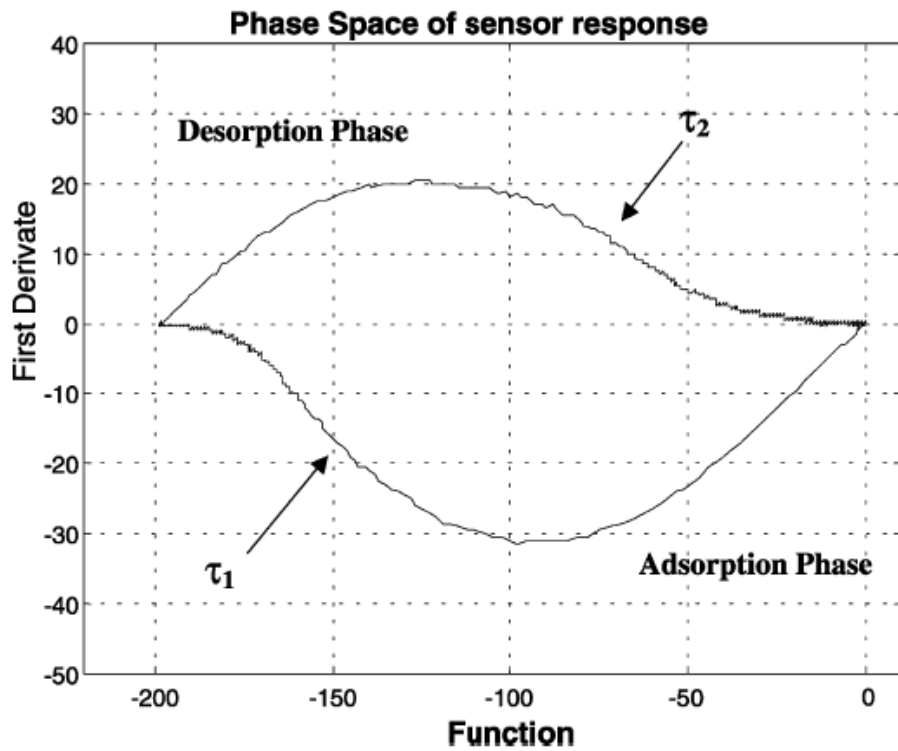


Figure 3: Trajectory in the phase space defined by the observed QCM sensor signal shown in Figure 3 (i.e. Function) and its first derivative. Reproduced from (Martinelli et al. 2003), Elsevier Science ©, with permission.

The description of the trajectory in PS may give rise to novel features related to the underlying phenomena that can be used to build more accurate gas identification or quantification models. The area spanned by the trajectory when it undergoes a phase transition (e.g. an adsorption phase) is the most obvious descriptor that can be used. This area, which corresponds to the sum, instant by instant, of the product between the signal and its first derivative considers both the magnitude of the sensor response and its velocity. However, the area is a descriptor that overlooks some characteristics of the trajectory evolution (i.e. the shape of the trajectory). This is why other descriptors, namely the dynamic moments, have been introduced.

5.2.2 Dynamic moments

If the dynamics of the interaction between a sensor and different gases differ, it can be expected that these differences should be reflected in the trajectories in PS. In other words, different gas-sensor interactions should lead to different shapes of the trajectories in PS. Under these circumstances, the area spanned by the trajectory can be too crude a descriptor since it neglects characteristics that are a ‘property’ of the sensor evolution. In (Martinelli et al. 2004) the use of morphological descriptors to better describe the trajectory characteristics associated to QCM sensors was introduced.

Morphological descriptors can easily encode the shape of the trajectories. Many different families of morphological descriptors exist and these are widely used after the segmentation step in artificial vision engines. Martinelli and co-workers considered using sets of descriptors representing the parameters analogous to the second moments of the area of a geometrical figure in a 2-D space. These quantities, which are also employed in the field of control systems are sometimes called dynamic moments (DM) (Fichera et al. 2001). Consider a generic PS defined from an observable quantity $s(t)$. By setting a time lag τ the space coordinates are:

$$[s(t) \ s(t + \tau) \dots s(t + (k - 1)\tau)] \quad (3)$$

and the dynamic moments are calculated considering both the coordinates and bisectors of the PS and defined as follows:

$$MD2 = \frac{1}{n} \sum_{i=0}^n x_i y_i \quad (4)$$

$$MD3_{PB} = \frac{\sqrt{2}}{2n} \sum_{i=1}^n (x_i^2 y_i - x_i y_i^2) \quad (5)$$

$$MD3_{SB} = \frac{\sqrt{2}}{2n} \sum_{i=1}^n [2x_i^3 + 3(x_i^2 y_i - x_i y_i^2)] \quad (6)$$

$$MD3_X = \frac{1}{2n} \sum_{i=1}^n (x_i^3 - 3x_i y_i^2) \quad (7)$$

$$MD3_Y = \frac{1}{2n} \sum_{i=1}^n (x_i^3 - 3x_i^2 y_i) \quad (8)$$

the number gives the degree of the moment, the subscript indicates the direction along which the moment is calculated: *PB* and *SB* are principal and secondary bisectors, *X* and *Y* are the PS axis. *x* and *y* are quantities related to the observable quantity *s(k)* as follows:

$$x_k = s(k); \quad y_k = s(k + \tau) \quad (9)$$

Each of the moments describes the different morphological features of the trajectory, so that the collective use of more than one moment is required for an exhaustive description. It is important to emphasize that the moment value depends on the time lag generating the PS.

In a typical measurement, a gas sensor is exposed to an adsorption-desorption experiment in which the concentration of some species is changed abruptly. A simple model for the sensor response is a double exponential model.

$$s(t) = A_1 e^{-t/\tau_1} + A_2 e^{-t/\tau_2} + c \quad (10)$$

To illustrate the resolving power of dynamic moments, let us consider that two response signals associated to the interaction of a QCM sensor with two different gas species only differ in a small variation in one of the time constants. The upper plot in Figure 4 shows these signals. The frequency change is identical for both signals, so for the usual feature extracted with QCM sensors, the two measurements are indistinguishable. The lower plot in Figure 4 shows the trajectories in PS built from the signal and its delayed value. The differences in PS between the two trajectories are slightly more evident than the signal differences in the time domain.

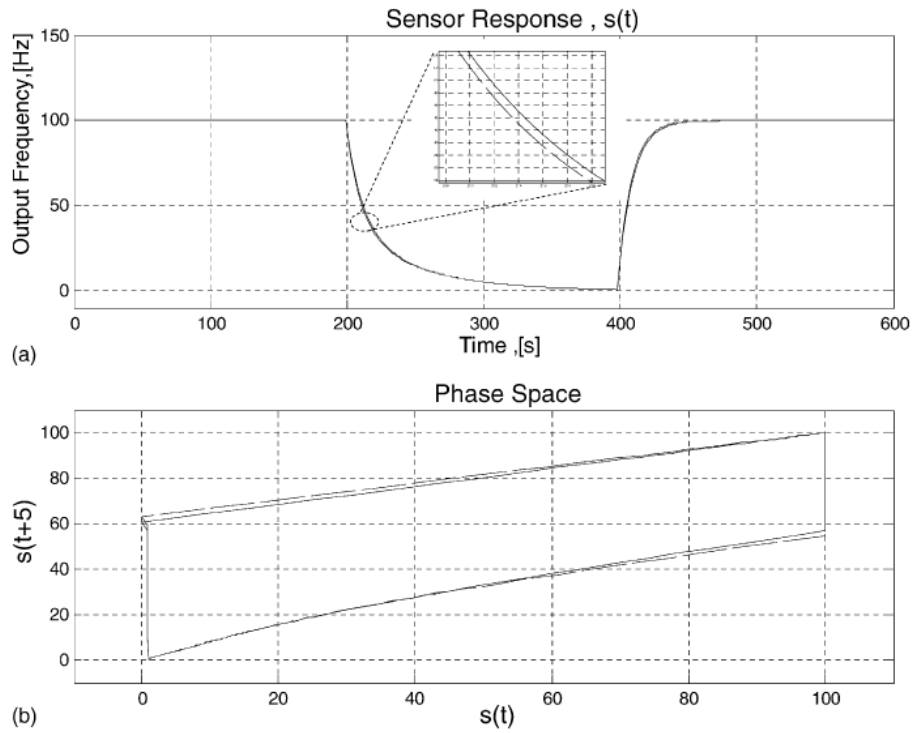


Figure 4: Signals in the time domain of two multi-exponential models for a QCM sensor exposed to a sudden variation of analyte concentration (a). Trajectories in the PS defined by $s(t)$ and $s(t+5)$ as coordinates (b). Reproduced from (Martinelli et al. 2004), Elsevier Science ©, with permission.

In Figure 5, three dynamic moments are plotted as a function of the time lag. In order to improve the recognition between the two response signals (i.e. to maximize the difference between the dynamic moments) the choice of both the dynamic moments to be used and the time lag are of paramount importance. For the case corresponding to the signals shown in Figure 4, $MD2$ shows the largest difference when the time lag is set to 1 s. For $MD3_{PB}$ and $MD3_{SB}$ the separation between the signals is maximised when the time lag is set to 4 s. This example illustrates how the dynamic moments are useful to emphasise subtle differences between sensor response signals that appear rather similar in the time domain.

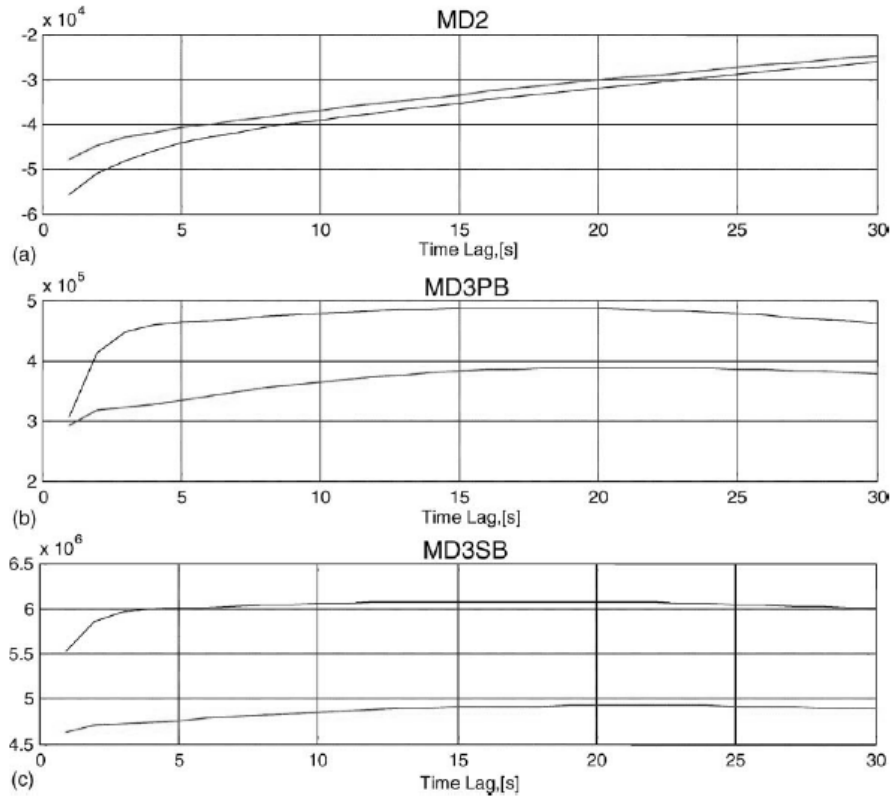


Figure 5: MD2, MD3_{PB} and MD3_{SB} plotted against the time lag calculated from the trajectories shown in Figure 5 b. Reproduced from (Martinelli et al. 2004), Elsevier Science ©, with permission.

5.2.3 Energy vector

As reviewed above, in recent years signal theory issues have been reconsidered from the point of view of the opportunity to describe the time evolving sensor signal in a different representational space where the changes of the sensor state could be more easily appreciated. It is in this context that the PS of chemical sensor signals was introduced. The concept of PS, has been shown to be useful for studying the behaviour of individual sensors.

Let us consider that gas sensor arrays are a mathematical construction where the sensor outputs are arranged as components of vectors and these vectors are built after feature extraction from the time evolution of each sensor signal. Therefore, once the vectors are mathematically formed, sensors lose their individual character and are treated as vector elements. In this way, features have been extracted from each individual sensor signal disregarding its array membership. An alternative to this conventional method is the global feature extraction, which consists of considering each sensor response always as a part of the total response of a sensor array. As stated before, the treatment of the sensor signal evolution with time can be properly treated in the frame of the theory of signals. Beside the concept of PS, very useful to study the behaviour of individual sensors, the theory of signals provides an excellent tool, the mutual energy, suitable to study the relationships between sensors belonging to the same array. The sensors of an array simultaneously interact with the same chemical pattern at a time and, as a consequence, the sensor signals share a certain degree of correlation. The amount of correlation may change according to the quality and quantity of the gas species to which the array is exposed.

A useful way to quantify the relationship between sensors of an array is provided by the mutual energy (Papoulis 1984). Given two generic signals $x(t)$ and $y(t)$, the mutual energy in the interval (t_0, t_1) is defined by the following equation:

$$\xi_{xy} = \int_{t_0}^{t_1} x(t)y(t)dt \quad (11)$$

In the trivial case when $x(t) = y(t)$ Eq.(11) calculates the energy of $x(t)$. During a measurement, sensors interact with the gases present in the environment to which they are exposed and their signals change. This variation is considered as the signal portion utilized for the energy feature extraction procedure as introduced in (Martinelli 2003). The value of the energy may depend on the affinity between the gas species present and the sensor active layer. A measurement is usually completed when the sensor has reached a dynamic equilibrium with the environment and, as a consequence, it reaches a steady value of energy. The difference between the two energy levels, before and after the exposure, can be considered as a measure of the sensor reaction and it is strongly correlated with the transient response signal. In data analysis, feature extraction provides synthetic descriptors for each individual sensor and only after

their extraction are they used as input for data analysis. Nevertheless, it is worth noting that the sensors of the array interact with the same chemical pattern at the same time, and that the m responses (i.e., number of measurements) are connected in some way between themselves. These correlations may represent further information about the gas species under test and their relation with the global sensor array.

It is well clear how the energy features take into account not only the absolute value of the sensor signals but also their dynamics. Actually, the time constants of the two signals and the relation between them play an important role in determining the energy. It is also important to remark that the ξ_{xy} depends on the size of the signals, namely its time length. This last aspect can influence the recognition power of the method because differences in sensor signal portions can hide the information content. Starting from the previous considerations about the correlation between the sensor signal evolution and the mutual energy, any sensor array measurement may be represented by an n -column vector defined as follows:

$$EV = \begin{bmatrix} \xi_{1,1} & \xi_{1,2} & \dots & \xi_{1,n} & \xi_{2,1} & \xi_{2,2} & \dots & \xi_{2,n} & \dots & \xi_{n-1,n-1} & \dots & \xi_{n,n} \end{bmatrix} \quad (12)$$

where n is the number of sensors within the array. This vector, formed by the application of Eq.(11) to the sensor array responses, results in a vector of energies called the energy vector (EV). The EV contains each sensor energy and all the mutual energies.

5.2.4 Multi-way analysis

Data from sensor array based analytical instruments can be arranged in three dimensions as sample $\times$ time $\times$ sensor. Due to software limitations but more often to general practice, the time –dimension of the data is reduced so traditional two-way chemometric methods such as principal component analysis or linear discrimination analysis can be run to explore data. As a result of this dimensionality reduction process, the internal relationship between the different modes (i.e., samples, time and sensors) is destroyed and the result can be an incomplete data analysis leading to a suboptimal final interpretation. Ideally, a better strategy would not require transforming the informative measured time profile into univariate features before meaningful data analysis can be performed.

The method discussed here, which is based on multi-way chemometric models, is a global method (i.e. it considers each sensor response always as a part of the total response of a sensor array) and allows for processing of the time information. It explores the total time profiles such that the potentially useful information is not discarded by feature extraction before the actual data analysis step is performed (Skov et al. 2005).

Since the most commonly employed model for performing two-way data analysis is principal component analysis (PCA), a short description of this method is given. This will help making a comparison with multi-way techniques.

5.2.4.1 Principal component analysis

According to (Esbensen 2002), PCA constitutes the most basic ‘work horse’ of all multivariate data analysis. PCA is a linear combinatorial method which reduces the complexity of the dataset, from the initial n -dimensional sensor space to a few dimensions. The inherent structure of the dataset is preserved while its resulting variance is maximised. In other words, data points will be scaled along new dimensions, linear combinations of the initial dimensions. The magnitudes of the coefficients, in the resulting linear combinations, give an indication of the relative importance of the initial dimensions in the data structure. PCA is performed with no information on the classification of samples. It is based solely on the variance of the dataset. Thus it is an unsupervised method.

$$\begin{array}{c} J \\ \boxed{\mathbf{X}} \\ I \end{array} = \begin{array}{c} R \\ \boxed{\mathbf{T}} \\ I \end{array} \begin{array}{c} J \\ \boxed{\mathbf{P}^T} \\ I \end{array} + \begin{array}{c} J \\ \boxed{\mathbf{E}} \\ I \end{array}$$

Figure 6: Representation of an R component PCA model of the data matrix X.

The underlying premise in PCA is that the raw data in the response matrix X can be decomposed into eigenvectors and associated eigenvalues. Of the several methods available to decompose X, one of the most common is through singular value decomposition (SVD). As shown in Figure 6, a PCA model makes a bilinear decomposition of the X matrix into a score matrix (T) (eigenvectors) and

a loading matrix (**P**) (eigenvalues), which describe the original data in a more condensed way. The residuals (i.e., the difference between the original and the reconstructed matrix with the calculated model) are collected in the E matrix. Thus, in PCA each component is the outer product of two vectors (scores pertaining to samples and loadings pertaining to variables). The model is linear in the scores for fixed loadings and vice versa and therefore, bilinear. If a given data set can be modelled by an R-component PCA model, then the $I \times J$ matrix **X** can be approximated as $\mathbf{TP}^T$ where T is the score matrix ($I \times R$) and **P** the loading matrix ($J \times R$).

As already mentioned, the aim of PCA is to retain the main information contained in the original variables in a smaller number of variables, called principal components (PC), which describe the main variations in the data. These PCs are linear combinations of the original variables. Some properties of PC are that they are orthogonal (i.e., uncorrelated with each other), hierarchical (i.e., the first PC retains the maximum information of the data, the second PC retains the maximum information that is not included in the first one, and so on), and are calculated sequentially (i.e., the F-1 component model is a subset of the F component model).

Mathematically, PCA decompositions can be written as follows:

$$X_{ij} = \sum_{r=1}^R a_{ir} b_{jr} + e_{jk} \quad (13)$$

where elements a_{ir} , b_{jr} refer to the scores and loading matrices **A** and **B** and where i, j refer to the dimensions of the matrix and R is the number of factors of the PCA model.

5.2.4.2 Parallel factor analysis

Parallel factor analysis (PARAFAC) is a decomposition method for multi-way data that works directly on the multi-way structure. This decomposition is made into triads or trilinear components. Each component consists of three loading vectors that are treated mathematically in the same way. However, the vector associated with the sample mode is often named the score vector. A PARAFAC model of a three-way array is therefore given by three loading matrices, **A** (samples), **B** (time profile) and **C** (sensors), with elements a_{if} , b_{jf} and c_{kf} , where the i, j and k indices refer to the dimensions of the array and f is the number of factors of the model. Figure 7 is a graphical representation of the PARAFAC

decomposition. The cube E contains the residuals. Mathematically, the PARAFAC model for R components is expressed as follows:

$$X_{ijk} = \sum_{r=1}^R a_{ir} b_{jr} c_{kr} + e_{ijk} \quad (14)$$

Two of the most important properties of PARAFAC are uniqueness and the fact that it is not a nested model. Unlike bilinear methods such as PCA, PARAFAC provides unique solutions. This means that, apart from trivial variations of scale and column order, no restrictions are needed to estimate the model. Another difference with respect to PCA is that PARAFAC is a non-nested model (i.e., it cannot be calculated sequentially). This means that the F-1 factor model is not a subset of the F factor model. There are several techniques for determining the number of factors in a PARAFAC model, like residual analysis, visual appearance of loadings, number of iterations of the algorithm, core consistency, etc. For a more in-depth discussion of the PARAFAC model the reader is referred to the tutorial (Bro 1997).

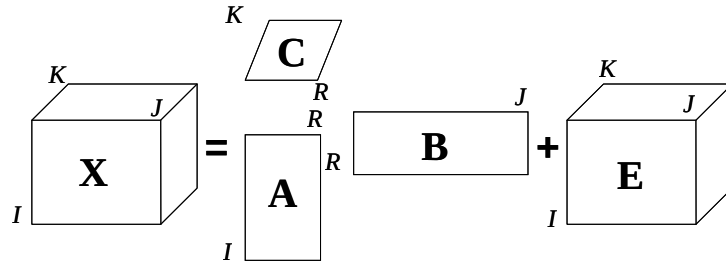


Figure 7: Representation of an R component PARAFAC model of the data array $\underline{X}$.

Sometimes the data profiles are characterised by showing shifted peaks or more generally varying shapes. PARAFAC2, which is a development of the original PARAFAC method is more efficient at handling such a situation. In PARAFAC2 scores (i.e. the matrix A) and sensor loadings (i.e. the matrix C) are of similar structure as in PARAFAC. However, the time loadings are not in a single loading matrix B as in PARAFAC. Instead, for each sensor l an individual time loading matrix B_l is given. In other words, unlike in PCA or in PARAFAC, the PARAFAC2 model allows every sensor to have its own set of time loadings. For the model to be valid and retain uniqueness, all the cross-product matrices of the B_l matrices should be constant for all l :

$$B_l^T B_l = B^T B, \quad \text{for } l = 1, \dots, L \quad (15)$$

The reader is referred to (Bro et al. 1999) and (Kiers et al. 1999) for a deeper insight into this method.

5.3 EXPLOITING SENSOR DYNAMICS

This section illustrates how the different techniques described above have been used to analyse the dynamic response of sensors either operated in a temperature modulation mode or exposed to a sudden change of gas concentration.

5.3.1 Transient response analysis in the phase space

Phase space methods were first applied to the response of quartz crystal microbalance sensors (QCM) (Martinelli et al. 2003). The QCM devices were coated with a molecular film of metalloporphyrins, which provided selective adsorption towards some target gases. The sensors were exposed to abrupt variations of the chemical content of the air (e.g. from clean air to a gas mixture diluted in air and back to clean air), which allowed for recording both the adsorption and desorption transients of the QCM sensors (see Figure 2). These transients are related not only to the adsorption/desorption rates between the sensor chemistry and sensed analytes but also to the characteristics of sample delivery such as flow, volume of the sensor chamber and tubing. Therefore, only a portion of the sensor signal carries important information.

As described above, the PS was defined by two canonical variables: the response signal and its first derivative. A typical trajectory in the phase space is shown in Figure 3. The feature extracted from a trajectory in the PS was the area spanned by the trajectory during either an adsorption or a desorption phase. The performance of the phase space method was assessed in two different EN experiments. The experiments consisted of the classification of different quality of apples and in the estimation of the concentration of androstenone in pork fat matrices.

5.3.1.1 Apple classification

The objective of this experiment was to build and validate a classifier for apples according to their degree of mealliness (internal defect to post-harvest over

ripening) and skin damage (bruises and cuts). Apples with no mealiness and with two levels of mealiness were employed. Additionally a set of apples with skin cuts were also used. Each fruit was closed in a sealed glass jar and left at room temperature for 20 minutes to allow a stable headspace to develop. The headspace was then allowed to flow inside the sensor chamber while the sensor signals were acquired.

The area spanned by the trajectory in the PS (without any additional pre-processing) was used as the input feature to build a partial least squares discrimination analysis model (PLS-DA) aimed at classifying apples. To better assess the performance of the PS method in apple classification, the usually employed steady-state feature extracted from QCM sensors (i.e., frequency shift) was also used to build a different PLS-DA model for apple classification. Both models were cross-validated using a leave-one out technique.

For the PLS-DA model built on the steady-state frequency shift, its first latent variable carried about 99% of the total variance in the data. Since almost the total variance is explained by the first latent variable it can be derived that sensors respond in the same way to class variation. The model cross-validated with a leave one out technique achieved an 80% correct classification in 4 categories (i.e., good apples, apples with 2 different levels of mealiness and skin-cut apples). While good and skin-cut apples could be easily identified, the model had problems in associating the correct level of mealiness to spoiled apples.

For the PLS-DA model built on the area of the trajectories in PS, its first two latent variables accounted for 69.28% and 30.46% of data variance. The information accounted for by the area seems not to be so cross-correlated as the frequency shift information. Indeed, the cross-validated model achieved a 100% correct apple classification in the 4 categories.

5.3.1.2 Androstenone concentration estimation

Different samples of pure pork fat were artificially contaminated with known amounts of androstenone in the range 0.7 $\mu\text{g/g}$ up to 10 $\mu\text{g/g}$. Samples were prepared in sealed vials held at 35 $^{\circ}\text{C}$ in a bath with temperature control for 30 min. A portion of the headspace was extracted and allowed to flow into the sensor chamber. PLS models aimed at estimating the concentration of androstenone in the different samples were built and cross-validated. Similarly to the apple experiment, the leave one out cross-validation technique was employed and the performance of the models that used either the area or the frequency shift

was compared. When the frequency shift was used, the minimum of the root mean squared error in prediction was 4 $\mu\text{g/g}$ when three latent variables were selected. On the other hand, when the area spanned by the trajectory on the PS was employed, the minimum of the root mean squared error in prediction was 1.4 $\mu\text{g/g}$. Therefore, the use of the area as sensor response feature results in a reduction of about a factor of 3 in the prediction error.

The concept of area in the PS is connected to both the transient and the stationary conditions of sensor response fusing them into a unique parameter. Since it is well-known that the dynamic sensor response contains important information to perform qualitative and quantitative gas analysis, it is not surprising that by using the area as an input feature results in models with improved classification and concentration estimation ability.

5.3.2 Transient response analysis using dynamic moments

As discussed above, depending on the nature of the phenomena explaining the transient response of a sensor, trajectories on the PS can assume a large variety of shapes. Since the shapes can be expected to be associated to the properties of the underlying physical phenomena, it would be interesting to use not only the area spanned by these trajectories but also morphological descriptors for describing (i.e., encoding) shapes.

The ‘dynamic moments’ strategy was first applied to the response of quartz crystal microbalance sensors by the same research group that introduced the PS (Martinelli et al. 2004). The performance of dynamic moments was compared to the one obtained using the standard frequency shift in two typical EN applications. The first experiment considered the classification of kiwi-fruits of the same cultivars but farmed using either conventional or organic agriculture. The second experiment envisaged the classification of peaches according to their ripening stage in three different categories according to three different storage times. Measurements were performed while enclosing individual fruits in glass jars. Jars were sealed and held at a constant temperature for 20 min to allow the headspace to stabilise. Finally, the headspace was allowed to flow into the sensor chamber. The sensor responses consisted of a sequence of frequencies sampled at constant intervals of 10 s. Time lags multiple of 10 s were employed to build the PS.

5.3.2.1 Kiwi-fruit database

In this experiment the dynamic features used were MD2 with a time lag equal to 20 and 30 s, and MD3_{PB} with a time lag equal to 20 s. PLS-DA models were built and validated using the leave one out cross-validation approach. When 4 latent variables were employed, the classification rate peaked at 97% with only one organic sample being misclassified as industrial. On the other hand, when the steady-state frequency shift was the only feature employed to train and validate the PLS-DA model, the success rate in classification degraded to 82%.

5.3.2.2 Peaches database

For this database the dynamic features used were MD2 with a time lag equal to 10 and 20 s. Once again PLS-DA models were built and validated using the leave one out cross-validation approach. This time a three-category classification was envisaged, which corresponded to pre-ripe, ripe and over-ripe peaches. Feeding the classifier with the dynamic moments resulted in 89% of the samples being correctly classified. Classification errors corresponded to ripe peach samples being classified as pre-ripe or over-ripe. The highest correct classification rate was obtained when 4 latent variables were employed. When the frequency shift was employed to build and validate the PLS-DA based classifier, the classification rate degraded down to 73%.

5.3.2.3 Rancidity in potato crisps

The dynamic moments strategy has been also applied to the transient response of metal oxide gas sensors (Vergara et al. 2007a). This experiment, which consisted of the classification of the rancidity in potato crisps using dynamic headspace analysis, was performed with a 12-element TGS-type metal oxide gas sensor array. Four boxes (labeled A, B, C and D) with 200 g packs of crisps (12 packs per category) were prepared. Crisps in box A were stored during 18 days in a dry and dark conservation chamber, where their temperature was kept around 20 °C. Crisps B, C and D were kept for 6, 12 and 18 days, respectively in a rancidity accelerating chamber at around 40 °C.

The sample delivery system consisted of two temperature-controlled stainless-steel vessels (a sampling vessel and a reference vessel). These chambers were identical and kept heated at 70 °C. The measurement procedure was as follows: In the first step (concentration phase), the crisps were heated at 70 °C for

30 minutes inside a sampling vessel in order to concentrate the headspace. In the second step (measurement phase), the airflow was re-circulated, which caused dynamic sampling of the crisps' headspace. During this phase, the resistance of the sensors was acquired and stored for later processing.

The responses of the 12 commercially available metal oxide gas sensors were used to classify the rancidity of potato crisps. Initially, the feature extracted from each sensor response was the steady-state conductance change. A fuzzy ARTMAP network was used to classify the samples according to their rancidity stage. The reader is referred to Chapter 3, where the basics of the fuzzy ARTMAP neural network are described. The success rate reached in rancidity classification was 68.75% (estimated by the leave-one-out approach).

Additionally, a PLS-DA model was built to discriminate among crisp samples. The success rate of this method was 68.75%, identical to that obtained by the fuzzy ARTMAP classifier. The samples misclassified belonged to categories B, C and D. In other words, although the system could discriminate fresh crisps from rancid ones, errors occurred in the discrimination of rancid samples with different rancidity states. The dynamic moments method was implemented with different time lags in order to select the optimal time lag to be used. For this database, MD2 with time lags equal to 2 and 25 s were considered for each sensor within the array. A PLS-DA model was built and its performance evaluated using the leave-one-out approach. The success rate in potato crisp classification was 85.4%. This compares favourably with the result obtained when the steady-state sensor response was used. The cross-validated PLS-DA models were formed with 10 latent variables.

5.3.2.4 Gas mixture analysis

This experiment consisted in the identification of nitrogen dioxide, ammonia and their mixtures using a micro-array of thermally-modulated metal oxide gas sensors (Vergara et al. 2007a). Four tungsten oxide gas microsensors were placed in a 20-ml volume test chamber and were used to measure vapours of three different concentrations of NH_3 (100, 500, 1000 ppm), NO_2 (0.5, 1, 2 ppm), and their binary mixtures diluted in dry air. Each measurement was replicated 5 times. The total gas flow was set to 200 ml/min and kept constant. The working temperature of the micro-hotplate gas sensors was modulated with an optimised multi-sinusoidal signal. The selected frequencies were 39.0 mHz, 141.0 mHz, 429.0 mHz 480.7 mHz and 743.0 mHz (Vergara et al. 2005). Every measurement

took approximately 6 minutes to complete and the sampling frequency was set, once again, to 10 Hz. Figure 8 shows a fragment of the response signal of a temperature-modulated tungsten oxide sensor in the presence of ammonia.

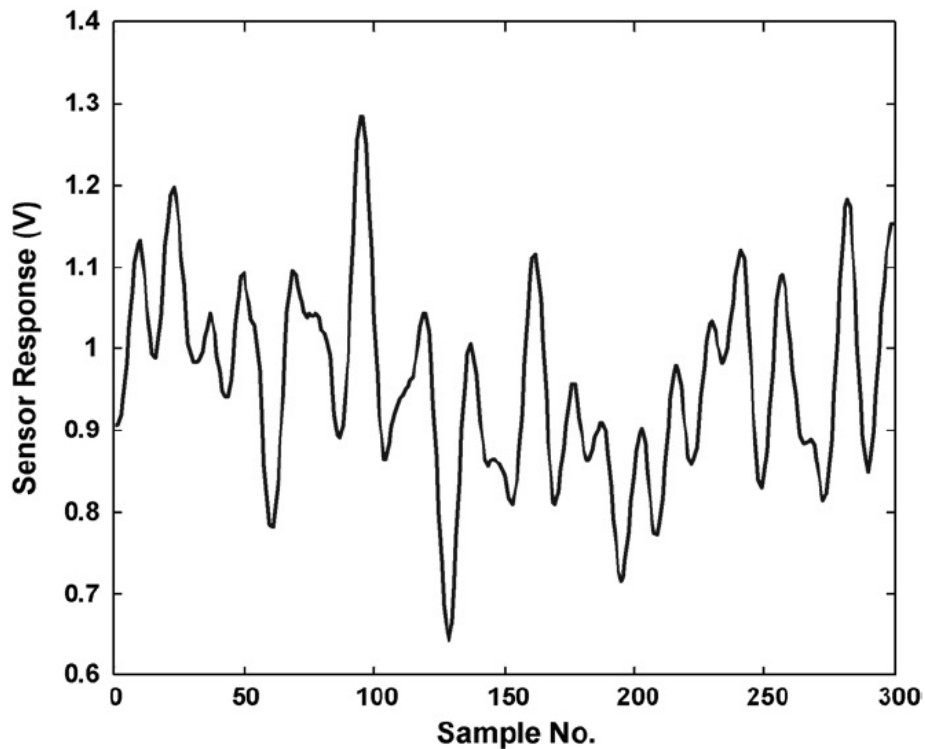


Figure 8: Response of a WO_3 microhotplate sensor in the presence of ammonia (1000 ppm) when its operating temperature is modulated using the sum of 5 sinusoidal signals. Reproduced from (Vergara et al. 2007a), Elsevier Science ©, with permission.

In the first step, the response signals of the temperature-modulated microsensors were processed as follows. The absolute value of the fast Fourier transform (FFT) was computed and the values of the D.C. component and the 5 harmonics corresponding to the modulating frequencies were extracted. Therefore, from each measurement, 6 features were extracted and used to build a PLS-DA classifier aimed at discriminating between the different species

measured. The classifiers were cross-validated using a leave-one-out approach. The success rates in gas/gas mixture identification was 91.0%.

In the second step, the dynamic moments method was employed to extract features from the sensor responses. The method was implemented at different time lags and the use of response fragments with different lengths was considered. Figure 9 shows the typical trajectories in the PS domain for a tungsten oxide sensor in the presence of the gases studied.

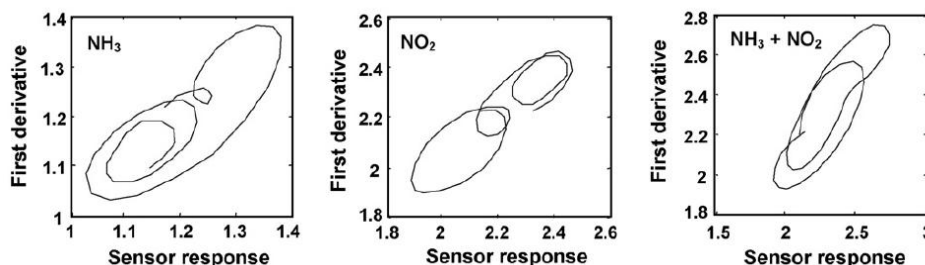


Figure 9: Typical response of a tungsten oxide micro-hotplate sensor to ammonia (500 ppm), nitrogen dioxide (1 ppm) and ammonia + nitrogen dioxide (500 + 1 ppm) in the PS domain. The first derivative of the sensor response for a time lag equal to 3 ms (y-axis) is plotted vs. the sensor response (x-axis). Reproduced from (Vergara et al. 2007a), Elsevier Science ©, with permission.

Finally, a small fragment of the initial sensor response signal was selected (i.e. 2.5 s of the response, which is about 1.8 % of its total length). The dynamic moments used to build PLS-DA classifiers were MD_2 and $MD_{3\gamma}$ at the time lags equal to 1 and 3 ms. The success rate in gas and gas mixture identification was 95.6% (estimated by leave-one-out cross-validation). PLS-DA models used 5 latent variables. While NH_3 samples could be perfectly discriminated, confusion occurred between samples of NO_2 and $NO_2 + NH_3$ mixtures. Increasing the length of the response signal fragment used to calculate the dynamic moments did not improve these results.

When the results obtained using the more standard features extracted by the FFT are compared to those obtained using the dynamic moments, it can be derived that the latter obtains a slightly better gas identification performance while dramatically reducing measurement time.

Considering the different datasets analysed, it clearly appears that the selection of the best dynamic moments is not straightforward. The relationship between the information content of the dynamic moments and the recognition power does not follow an analytical law. As a result, the performance of the classifier is not correlated with the order of the moment and, in particular, the amount of correct classifications does not reach a convergence value while increasing the order of the dynamic moments. This situation is rather similar to the optimisation of the number of principal components in Principal Component Regression (PCR), where the order of the principal components, proportional to the correspondent eigenvalue, is not correlated with their predictability of the regressed variable. Also in this case the optimal choice of principal components is heuristically carried out.

5.3.3 The energy vector approach

As stated in sub-section 5.2.3, the energy vector approach is a *global* feature extraction technique in the sense that it provides a tool suitable for studying the relationships between sensors belonging to the same array. Indeed, the sensors of an array simultaneously interact with the same chemical pattern at a time and, as a consequence, the sensor signals share a certain degree of correlation. The amount of correlation may change according to the quality and quantity of the gas species to which the array is exposed. The method was applied recently to the identification of volatile compounds using an array of thermally-modulated metal oxide gas sensors (Vergara et al. 2007b).

An array of four micro-hotplate gas-sensors was placed in a 20 ml volume test chamber to measure vapours, at three different concentrations, of acetaldehyde, ethylene and ammonia. This chamber was connected to a continuous flow system that allowed us to obtain different concentrations of pollutant gases diluted in dry air.

The total gas flow was set to 200 ml/min and kept constant. The response of the micro-gas sensor array was measured when the operating temperature of sensors was modulated using a multi-sinusoidal signal. A set of six optimal modulating frequencies (frequencies previously selected in a systematic way) was selected and used to synthesise the temperature-modulating multi-sinusoidal signal. The same multi-sinusoidal signal was applied to the heating element of every sensor within the array. The selected frequencies were 12.8, 25.6, 38.5, 92.9, 339.7 and 682.7 mHz. The procedure for generating this signal and how the

frequencies have been selected in a systematic way is illustrated in (Vergara et al. 2005).

Every measurement took approximately 6 min to complete. Figure 10 shows a fragment (split in segments of 1 s) of a typical transient response of a gas sensor exposed to 50 ppm of ammonia used for the energy vector analysis.

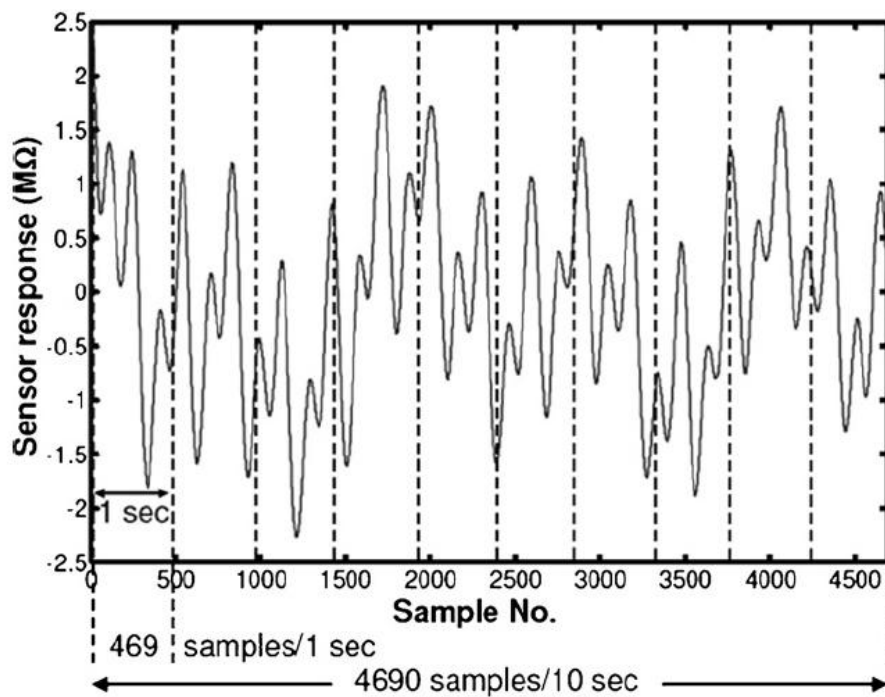


Figure 10: Fragment of the gas sensor response to a six-sinusoidal modulating signal applied to the heater element in the presence of 50 ppm of ammonia. Signal is divided into 1 s sub-segments. Reproduced from (Vergara et al. 2007b), Elsevier Science ©, with permission.

The energy vector was calculated by applying Eq.(11), where different segments of the gas sensor array response signal were considered. One energy vector was formed for each measurement where the sensor energy and the mutual energy of all possible couples of sensors in the array (see Eq.(12)) were considered. Therefore, each energy vector was the feature extracted from the gas sensor array transient response and input to a PLS-DA classifier. The PLS-DA models were validated with the leave-one-out cross-validation procedure in order

to obtain the success rate in gas discrimination. A 100% success rate in gas identification (regardless of their concentration) was obtained, even when a short segment of the response signal (e.g., 1 s) was employed.

Finally, to better assess the energy vector approach combined with the temperature modulation, the method was compared against the standard FFT feature extraction method. This method is one of the most commonly used when thermal modulation strategy is employed in gas sensors. The success rate in gas identification of the PLS-DA model built using as input the absolute values of the six harmonics corresponding to the temperature modulating frequencies peaked at 99.4%, with confusion occurring between ammonia and acetaldehyde samples. These results are obtained provided that the FFT is computed over a long period of the response signal (no shorter than 100 s). If shorter response signals are considered, performance significantly degrades since not even a period of all temperature-modulating signals is present.

Therefore, the energy vector method not only ameliorates slightly the identification performance but also dramatically reduces the length of the response signal needed for processing, so measurement time can be reduced too.

5.3.4 The use of multi-way methods

Much of the early work on multi-way methods applied to the analysis of chemical species was performed with the use of a combination of some type of chromatography with a multivariate detector, often spectroscopy (Geladi 1989). Here we will review the work done in the field of gas sensors.

5.3.4.1 Classification of liquorice samples

Skov and co-workers (Skov et al. 2005) investigated the usefulness of multi-way methods for classifying liquorice samples using an α -FOX-3000 EN manufactured by ALPHA M.O.S. (Toulouse, France), which comprised an array of 12 metal oxide gas sensors. Samples from a commercial liquorice produced in Denmark were evaluated. These could be classified into good, bad (deviating liquorices) and fabricated bad (by drying the good liquorices). Figure 11 shows the pre-processed (baseline-corrected) response signals of the sensor array to a liquorice sample. For further processing these response signals, 2-way methods such as unfold PCA or 3-way methods such as PARAFAC can be envisaged. ‘Unfold PCA’ could be appropriate if the time profiles were completely unrelated

across sensors or samples. Unfold PCA is simply conventional PCA performed on the unfolded pattern vectors. On the other hand, PARAFAC would be suitable if time profiles were related across sensors and samples (i.e. time profiles do not change in shape but only in magnitude). PARAFAC2 is an intermediate model since it allows time profiles to vary in shape while being somewhat related. The fact that the response peaks appear shifted in Figure 11 (i.e. their maximum does not occur simultaneously) makes the PARAFAC2 technique a good candidate for processing such data.

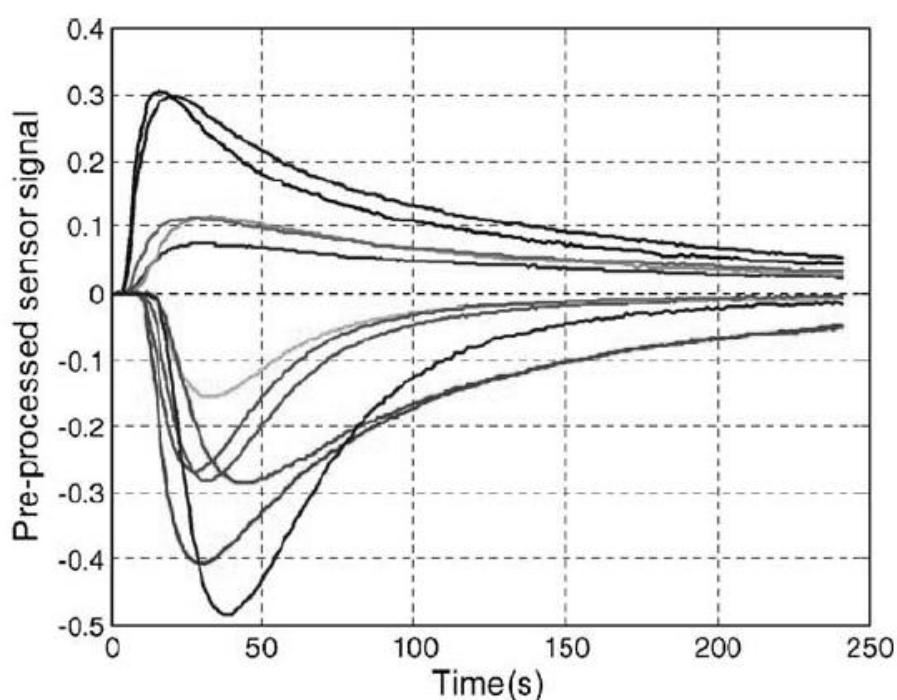


Figure 11: Baseline-corrected sensors signals of the metal oxide sensors within the FOX-3000 electronic nose employed to analyse liquorice samples. Adapted from (Skov et al. 2005), Elsevier Science ©, with permission.

Three different classification models were built using the above-mentioned processing techniques coupled to a nearest neighbour classifier. The worse results were obtained when the standard PARAFAC method was employed (with an important overlap between the bad and fabricated bad samples). The best

results were obtained when the PARAFAC2 method was used and the success rate in classification reached 98.33%. Finally, when the unfold PCA method was implemented, the success rate in classification was 95%.

The relatively poor classification of the samples using the standard PARAFAC model indicates that this data is not strictly tri-linear, but rather that the time profile of each sensor should be modeled independently, which is the case when the PARAFAC2 model is employed. The relatively good results obtained from the unfold PCA model also suggest that most information about the different samples is in the maximum of the signal. However, ignoring the three-way structure of the original data tends to give a lower separation of the samples and suggests that important information also can be found in the rest of the time profile curve.

5.3.4.2 Estimating the content of flavouring added to crisps

M. Padilla and co-workers (Padilla et al. 2006) studied the usefulness of PARAFAC as a data compression technique for an EN in food analysis. Sealed vials of potato crisps with different amounts of flavouring agents added were prepared (0 to 80 mg of flavouring). The headspace of these vials was measured using a Nordic Sensor Technologies System, NST3320 (Linköping, Sweden). The instruments comprised 13 metal oxide gas sensors. A baseline-injection-recovery-flush-stabilisation cycle of 15-10-10-120-110 s was implemented and sensor signals were acquired at a sampling rate of 1 Hz. A three-way matrix was formed (samples $\times$ sensors $\times$ time) containing the transient response signals and the trilinearity of the data was checked.

The use of a direct multi-way calibration (e.g. a multi-way PLS) was discarded because differences and lack of correlation existed between the PARAFAC factors and the flavouring concentration. Instead, an inverse least squares model was built, which took the PARAFAC scores matrix as the predictor variable matrix and the concentration, expressed in grams of added flavouring agents, as the predicted variable. It was concluded that the amount of flavouring added could be predicted with an absolute error of 10 mg.

5.3.4.3 Packaging emission on food

T. Heine and co-workers (Heine et al. 2007) investigated the usefulness of multi-way methods and an array of metal oxide gas sensors to identify 'off-odours' from packaging materials used by the food industry. Materials were cut

into pieces of equal area and introduced into vials and gas-tight sealed. The headspace of these vials was injected into a packed capillary column coated with a polar phase. Organic solvents and water were roughly separated by the packed column and detected with an array of 4 Pd-doped tin oxide gas sensors. Data were organised in a three-way matrix (samples $\times$ sensors $\times$ time) containing the transient response signals. The objective was to predict the total amount of organic solvents in the sample so regression models were built and validated. Multi-way regression methods (such as the multi way PLS) performed similarly to their 2-way counterparts but being more robust (e.g. using fewer latent variables and, therefore, being less prone to over fitting).

5.4 CONCLUSIONS AND OUTLOOK

A great deal of work has been done in the last fifteen years on the use of dynamic information for ameliorating the performance of gas sensors. Transient signals are the result of either abrupt variations of the chemical content of the air at which sensors are exposed or changes in their operating temperature (e.g. pulsed, sinusoidal or multi-frequency). The techniques which has been employed the most for extracting features from these response transients has been the fast Fourier transform, the discrete wavelet transform, the short-time Fourier transform and, to a lesser extent window time slicing. All these methods make it possible to extract a few parameters (e.g. a sub-set of wavelet coefficients), which are then used to build pattern recognition engines. These methods have proved to work well for increasing the discrimination ability and drift resilience of gas sensor arrays.

In this chapter a few recently introduced methods for exploiting sensor dynamics have been reviewed.

In the first place phase-space methods have been discussed. As an analogy to mechanical systems, the observed signal (i.e. the response transient) and its first derivative (or alternatively, a lagged version of the response transient sequence) are considered as the variables from which a phase space is defined. The typical adsorption-desorption experiments (e.g. caused by abrupt changes in gas composition) or temperature modulation experiments turn out to give closed trajectories in the phase space, which makes evident the reversibility of sensor response. The area spanned by the trajectory in the phase space has been shown to carry important information since it is connected to both static and dynamic

conditions. Since not only the area but also the shape of the trajectory can characterise adsorption-desorption processes and reaction kinetics in gas sensors, the use of a set of morphological descriptors has been introduced. These descriptors (dynamic moments) describe the properties of symmetry of the curves towards a number of fixed directions such as the coordinate axis and their bisectors. The use of phase space methods has been shown to be useful for improving the selectivity of gas sensors and also for reducing the measurement time. However, the trajectories in the phase space are to some extent sensitive to the experimental conditions (e.g. flow fluctuations) and, the optimal choice of the actual dynamic moments and time lags that should be used can very much depend on the specific application, as revealed by the different examples discussed. In conclusion, once the usefulness of these methods have reached an adequate level of acceptability, further studies are needed to assess their robustness.

Additionally, processing methods that consider the whole sensor array instead of each sensor individually have been discussed, too. Given the fact that the sensors within an array are simultaneously interacting with the same chemical pattern, their response signals should have a certain degree of correlation. Global methods assume that the amount of correlation changes according to the nature and the concentration of the species present in the chemical pattern.

The energy vector was the first approach that falls into this category to be discussed in this chapter. The method has been shown to work well at extracting important information from very short transient response signals, which may help to dramatically reduce measurement time. Significantly reducing measurement time is of interest because power consumption due to sensor operation can be reduced too. This is important if gas sensors are to become widespread in systems employing wireless communications and RF power harvest.

The second type of global methods discussed was the so called multi-way methods. Multi-way methods are interesting in the sense that unlike in standard two-way methods such as PCA or PLS, it is not necessary to transform the measured time profile into univariate features before data analysis can be performed. Multi-way classification methods such as PARAFAC or regression methods such as multi-way PLS have been shown to perform well in different gas sensor applications. Multi-way models tend to be more parsimonious (e.g. use fewer factors) than their 2-way counterparts and are, therefore, less prone to over-fitting. However, a remarkable property of multi-way methods is that under certain conditions they provide the ability to estimate the concentration of one

analyte in the presence of unknown interference. This property, which has been verified to hold for analytical instruments (e.g. chromatography linked to spectroscopy detection) still needs to be explored in the context of gas sensor arrays. Undeniably, it could bring a major breakthrough in the field of multi-sensor gas analysers and electronic noses. In conclusion, further work with more and more 'realworld' data that includes for example the presence of interfering species at high concentrations and response drift, is needed to better assess the potential of multi-way methods.

References

- Abe H., T.Y. Yoshimura, S. Kanaya, Y. Takahashi, Y. Miyashita and S. Sasaki, (1987), "Automated odor-sensing system based on plural semiconductor gas sensors and computerised pattern recognition techniques", *Analytica Chimica Acta*, 194, 1-9.
- Aigner R., F. Auerbach, P. Huber, R. Müller, G. Scheller, (1994), "Sinusoidal temperature modulation of the Si-planar-pellistor", *Sensors and Actuators B: Chemical B* 18-19: 143 - 147.
- Aishima T., (1991), "Aroma discrimination by pattern recognition analysis of responses from semiconductor gas sensor arrays", *Journal of Agriculture and Food Chemistry*, 39: 752-56.
- Amrani H., P. Payne, K. Persaud, (1996), "Multi-frequency measurements of organic conducting polymers for sensing of gases and vapours", *Sensors and Actuators B: Chemical B* 33: 137-141.
- Barbri El, N., Amari, A., Vinaixa, M., Bouchikhi, B., Correig, X., Llobet, E., (2007), "Building of a metal oxide gas sensor-based electronic nose to assess the freshness of sardines under cold storage", *Sensors and Actuators B: Chemical B* 128: 235-244
- Bourrounet B., T. Talou, A. Gaset, (1995), "Application of a multi-gas-sensor device in the meat industry for boar-taint detection", *Sensors and Actuators B: Chemical B* 26-27: 250-254.
- Bro R., H.A.L. Kiers, C.A. Andersson, (1999), "PARAFAC2 part II. Modelling chromatographic data with retention time shifts", *Journal of chemometrics* 13: 295-309.
- Bro R., (1997), "PARAFAC. Tutorial and applications", *Chemom. Intell. Lab. Syst.* 38: 149-171.
- Buttá N., M. Melli and S. Pizzini, (1990), "Influence of surface parameters and doping on the sensitivity and on the response times of tin oxide resistive sensors", *Sensors and Actuators B: Chemical B*, 2: 151-61.

- Cavichi R.E., J.S. Suehle, K.G. Kreider, M.Gaitan and P. Chaparala, (1995), "Fast temperature programmed sensing for micro-hotplate gas sensors", IEEE, Electronics Device Letters., Vol. 16, No. 6, 286-88.
- Cavichi R.E., J.S. Suehle, K.G. Kreider, M.Gaitan and P. Chaparala, (1996), "Optimized temperature-pulse sequences for the enhancement of chemically specific response patterns from micro-hotplate gas sensors", Sensors and Actuators B: Chemical B 33: 142-46.
- Clifford P.K. and D.T. Tuma, (1983), "Characteristics of semiconductor gas sensors II. Transient response to temperature change", Sensors and Actuators, 3: 255-281
- Corcoran P., H.V. Shurmer and J. W. Gardner, (1993), "Sensitivity enhancement for gas sensing and electronic nose applications", Sensors and Actuators B: Chemical B, 15-16: 32-37.
- Di Natale C., A. D'Amico, F. Davide, G. Faglia, P. Nelli, G. Sberveglieri, (1994), "Performance evaluation of an SnO₂-based sensor array for the quantitative measurement of mixtures of H₂S and NO₂", Sensors and Actuators B: Chemical B 20: 217-224.
- Di Natale C., F. Davide and A d'Amico, (1995), "Pattern recognition in gas sensing: well-stated techniques and advances", Sensors and Actuators, B 23: 111-118.
- Di Natale C., F. Davide, A. D'Amico, P. Nelli, S. Groppelli, G. Sberveglieri, (1996), "An electronic nose for the recognition of the vineyard of a red wine", Sensors and Actuators B: Chemical B 33: 83-88.
- Esbensen K. H., (2002), "Multivariate data analysis in practice-An Introduction to Multivariate Data Analysis and Experimental Design", 4th ed. Oslo, Norway, CAMO ASA.
- Fichera A., C. Losenno, A. Pagano, (2001), "Clustering of chaotic dynamics of a lean gas-turbine combustor", Appl. Energy, 69: 101-117.
- Gall M., (1993), "The Si-planar-pellistor array, a detection unit for combustible gases", Sensors and Actuators B: Chemical B 15-16: 260-264.
- Gardner J.W and P.N. Barlett, (1994), "A brief history of electronic noses", Sensors and Actuators, B 18-19: 211-220.
- Gardner J.W., (1990), "A Non-linear Diffusion-reaction Model of Electrical Conduction in Semiconductor Gas Sensors", Sensors and Actuators B: Chemical B 1: 166-170.
- Gardner J.W., (1991), "Detection of vapours and odours from a multisensor array using pattern-recognition Part 1. Principal Component and Cluster Analysis", Sensors Actuators B, 4: 109-115.
- Gardner J.W., H.V. Shurmer, T.T. Tan, (1992), "Application of an electronic nose to the discrimination of coffees", Sensors and Actuators B: Chemical B 6: 71-75.
- Geladi P., (1989), "Analysis of Multi-Way (Multi-Mode) Data", Chemometrics and Intelligent Laboratory Systems 7: 11-30.
- Göpel W., (1985), "Chemisorption and charge transfer at ionic semiconductor surfaces: Implications in designing gas sensors", Prog. Surf. Sci., Vol 20, 9.
- Gutierrez F.J., L. Arés, J.I. Robla, M.C. Horrillo, I. Sayago and J.A. Agapito, (1992), "Properties of polycrystalline gas sensors based on d.c. and a.c. measurements", Sensors and Actuators B: Chemical B , 8: 231-35.
- Gutierrez F.J., L. Arés, M.C. Horrillo, I. Sayago, J.A. Agapito, L. López, (1991), "Use of complex impedance spectroscopy in chemical sensor characterisation", Sensors and Actuators B: Chemical B 4: 359-363.
- Heine T., F. Röck, U. Weimar, (2007), "Analysis of food packaging emissions: A multi way approach for evaluating gas sensor array data", Proceedings of the ISOEN 2007, St Petersburg, Russia, 3-5 May 2007, 81-82.
- Hierlemann A., W. Göpel, (1995), "Polymer-based sensor arrays and multicomponent analysis for

- the detection of hazardous organic vapours in the environment", *Sensors and Actuators B: Chemical* B 26-27: 126-134.
- Kiers H.A.L., J.M.F. ten Berge, R. Bro, (1999), "PARAFAC2 part I. A direct fitting algorithm for the PARAFAC2 model", *Journal of Chemometrics* 13: 275-294.
- Ljung L., T. Glad, (2006), "Modelling of dynamic systems", Prentice Hall, Englewood Cliffs, NJ, 1994.
- Llobet E., Temperature-modulated semiconductor gas sensors in "Encyclopedia of Sensors", Vol. 10, American Scientific Publishers: 131-152.
- Llobet E., X. Vilanova and X. Correig, (1996), "Novel technique to identify hazardous gases/vapours based on transient response measurements of tin oxide gas sensors, Int. Conf. on Environmental Monitors and Hazardous Waste Site Remediation", *Proceedings SPIE* 0361-0748, Iss. 2504 (1995) 559-566, München (Germany), June 1996.
- Llobet E., X. Vilanova, J. Brezmes, J.E. Sueiras and X. Correig, (1997), "Transient response of thick-film tin oxide gas sensors to multicomponent gas mixtures", *Proceedings of the 11th European Conference on Solid-State Transducers, Eurosensors XI*, Warsaw, Poland, 1389-1392.
- Marco S., J. Samitier, J.R. Morante, (1995), "A novel time domain method to analyze multi-component exponential transients", *Meas. Sci. Technol.*, 6: 135-142.
- Martinelli E., C. Falconi, A. D'Amico, C. Di Natale, (2003), "Feature extraction of chemical sensors in phase space", *Sensors and Actuators B: Chemical* B 95: 132-139.
- Martinelli E., (2003) "Investigation on alternative approaches to chemical sensors data treatment", Ph.D. Thesis, University of Rome "Tor Vergata".
- Meixner H., U. Lampe, (1996), "Metal oxide sensors", *Sensors and Actuators B: Chemical* B 33: 198-202.
- Möller S., J. Lin and E. Obermeier, (1995), "Material and design considerations for low-power microheater modules for gas-sensor applications", *Sensors and Actuators B: Chemical* B 24-25: 343-346.
- Moore S.W., J.W. Gardner, E.L. Hines, W. Göpel and U. Weimar, (1993), "A modified multilayer perceptron model for gas mixture analysis", *Sensors and Actuators B: Chemical* B , 15-16: 344-348.
- Moseley P. T., in M. Prudenziati (Ed.), (1994), *Thick-film sensors*, Elsevier Science, Amsterdam, 261-288.
- Moseley P.T. and B.C. Tofield, (1971), *Solid-state gas sensors*, Adam Hilger, Bristol, 1987, 1-16.
- Moseley P.T., A.M. Stoneham and D.E. Williams, *Techniques and Mechanisms in Gas Sensing*, Adam Hilger, Bristol, UK, 1991.
- Moseley P.T., (1992), "Materials selection for semiconductor gas sensors", *Sensors and Actuators B: Chemical* B , 6: 149-156.
- Murakami N., S. Yasunage and K. Ihokura, (1983), "Sensitivity and sintering temperature of SnO₂ gas sensors", *Anal. Chem. Symp. Ser.*, 17: 18.
- Nakamura M., I. Sugimoto, H. Kuwano, (1996), "Application of plasma-polymer-film-coated sensors to gas identification using linear filters", *Sensors and Actuators B: Chemical* B 33: 122-27.
- Nimsuk N., T. Nakamoto, (2007), "Improvement of capability for classifying odors in dynamically changing concentration using QCM sensor array and short-time Fourier transform", *Sensors and Actuators B: Chemical* B 127: 491-496
- Padilla M., I. Montoliu, A. Pardo, A. Perera, S. Marco, (2006), "Feature extraction on three way

- enose signals", *Sensors and Actuators B: Chemical* B 116: 145-150.
- Papoulis A., (1984), "Signal Analysis", McGraw Hill International Editions, Montreal.
- Pearce T.C., J.W. Gardner, S. Friel, P.N. Bartlett, N. Blair, (1993), "Electronic nose for monitoring the flavour of beers", *Analyst*, 118: 371-377.
- Prudenziati M., (1991), "Thick-film technology", *Sensors and Actuators A*, 25-26: 227.
- Rose-Pehrsson S.L., J.W. Grate, D.S. Ballantine and P.C. Jurs, (1988), "Detection of hazardous vapours including mixtures using pattern recognition analysis of responses from SAW devices", *Anal. Chem.*, 60: 2801-2811.
- Roth M., R. Hartinger, R. Faul, H. E. Endres, (1996), *Sensors and Actuators B: Chemical* B 35-36: 358.
- Rout, C. S., Ganesh, K., Govindaraj, A., Rao, C. N. R., (2006), "Sensors for the nitrogen oxides", NO₂, NO and N₂O, based on In₂O₃ and WO₃ nanowires, *Applied Physics A-Materials Science & Processing*, 85 (3): 241-246
- Samitier J., J.M. López-Villegas, S. Marco, L. Cámara, A. Pardo, O. Ruiz and J.R. Morante, (1994), "A new method to analyse signal transients in chemical sensors", *Sensors and Actuators B: Chemical* B , 18-19 308-312.
- Schweizer-Berberich P.M., S. Vaihinger, W. Göpel, (1994), "Characterisation of food freshness with sensor arrays", *Sensors and Actuators B: Chemical* B 18-19: 282-290.
- Sears W.M., K. Colbow, R. Slamka, F. Consadori, (1990), "Selective thermally cycled gas sensing using fast fourier-transform techniques", *Sensors and Actuators B: Chemical* B 2: 283-89.
- Setkus, A., Kancleris, Z., Olekas, A., Rimdeika, R., Senuliene, D., Strazdiene, V., (2008), "Qualitative and quantitative characterization of living bacteria by dynamic response parameters of gas sensor array", *Sensors and Actuators B*, 130 (1): 351-358
- Shurmer H.V., J.W. Gardner, H.T. Chan, (1989), "The application of discrimination techniques to alcohols and tobaccos using tin-oxide sensors", *Sensors and Actuators* 18: 361-371.
- Skov T., R. Bro, (2005), "A new approach for modelling sensor based data", *Sensors and Actuators B: Chemical* B 106: 719-729.
- Taguchi N., US Patent 3 631 436.
- Takens F., (1980), "Detecting strange attractors in turbulence", in: D. Rand, L.S. Young (Eds.), "Dynamical Systems and Turbulence", Warwick, pp. 366-381.
- Vaihinger S., W. Göpel, J.R. Stetter, (1991), "Detection of Halogenated and other hydrocarbons in air: response functions of catalyst/electrochemical sensor systems", *Sensors and Actuators B: Chemical* B , 4: 337-343.
- Vergara A., E. Llobet, E. Martinelli, C. Di Natale, A. D'Amico, X. Correig, (2007) (a), "Feature extraction of metal oxide gas sensors using dynamic moments", *Sensors and Actuators B: Chemical* B 122: 219-226.
- Vergara A., E. Llobet, J. Brezmes, P. Ivanov, X. Vilanova, I. Gràcia, C. Cané Xa. Correig, (2005), optimised temperature modulation of metal oxide micro-hotplate gas sensors through multi-level pseudo random sequences, *Sensors and Actuators B: Chemical* B 111-112 271-280.
- Vergara A., E. Martinelli, E. Llobet, F. Giannini, A. D'Amico, C. Di Natale, (2007) (b), "An alternative global feature extraction of temperature modulated micro-hotplate gas sensor array using an energy vector approach", *Sensors and Actuators B: Chemical* B 124 352-359.
- Vilanova X., E. Llobet, R. Alcubilla, J.E. Sueiras and X. Correig, (1996), "Analysis of the conductance transient in thick-film tin oxide gas sensors", *Sensors and Actuators*, B 31: 175-180.
- Watson J and R.A. Yates, (1985), "A solid-state gas sensor", *Electron. Eng.*, Vol.57, no.701, 47-50.
- Williams D.E., P.T. Moseley, J.Mater. (1991), *Chem.* 1: 809-814.

- Wlodek S., K. Colbow and F. Consadori, (1991), "Signal-shape analysis of a thermally cycled tin oxide gas sensor", *Sensors and Actuators B: Chemical* B 3: 63-68.
- Zhang S., C. Xie, D. Zeng, Q. Zhang, H. Li, Z. Bi, (2007), "A feature extraction method and a sampling system for fast recognition of flammable liquids with a portable E-nose", *Sensors and Actuators B: Chemical* B 124: 437-443

Section II: Intelligent Systems and Engineering

Chapter 6

Multi-input Optimisation of River Flow Parameters and Rule Extraction Using Genetic-Neural Technique

Jianhua Yang, Evor L. Hines, Ian Guymer, Daciana D. Iliescu and Mark S. Leeson

6.1 INTRODUCTION

Longitudinal dispersion coefficient is a key variable for the description of longitudinal transport in a river. After being first introduced in Taylor (1954), extensive studies have been made based on experimental and field data for predicting the dispersion coefficient (Jobson 1997; Seo and Cheong 1998; Deng, Singh et al. 2001; Wallis and Manson 2004; Boxall and Guymer 2007). The majority of such work has used the Advection-Dispersion Equation approach because it's strong physical basis makes it more amenable to predicting conditions in rivers and streams for which no model has previously been calibrated (Wallis and Manson 2004).

In recent years, Artificial Neural Networks (ANNs) have become popular and useful tools for environmental modellers as they are perceived to overcome some of the difficulties associated with traditional statistical approaches; e.g. traditional approaches make assumptions with regard to stream geometry or flow dynamics (Maier and Dandy 1998). Several authors have reported successful applications of ANNs to the prediction of dispersion coefficient (Kashefipour, Falconer et al. 2002; Rowinski, Piotrowski et al. 2005; Tayfur and Singh 2005; Piotrowski, Rowinski et al. 2006; Tayfur 2006). For example, Rowinski, Piotrowski et al. (2005) applied a Multi-Layer Perceptron (MLP) with the Levenberg-Marquardt Algorithm (a variation of the Back-propagation Algorithm, BP) (Bishop 1995) to three different data sets which have been explored in the

literature. The lowest percentage of training data mean error was found to be 7.02%. Tayfur and Singh (2005) trained MLPs with the BP algorithm using data samples measured from 29 streams and rivers in the United States. They achieved the result that 90% of the dispersion coefficient was explained.

In these applications, however, little attention was given to the matter of whether or not the inputs used to train the ANN are the most appropriate ANN inputs. Due to the fact that not all of the available variables are necessarily equally informative (since some may be correlated, noisy or have no significant relationship with the output variable being modelled); input variable selection may be extremely important (Maier and Dandy 2000; Bowden, Dandy et al. 2005). According to Alexandridis, Patrinos et al. (2005) and Bowden, Dandy et al. (2005), the lack of input determination for ANNs may result in the following consequences:

- (1) Irrelevant variables may add extra noise which has consequential impact on the accuracy of the model.
- (2) Understanding complex models may be more difficult than understanding simple models that give comparable results.
- (3) As input dimensionality increases, the computational complexity and memory requirements of the model increase.

Techniques that are concerned with the determination of ANN inputs may be described differently in the literature (Yao 1999; Maier and Dandy 2000; Ramadan, Song et al. 2001; Alexandridis, Patrinos et al. 2005; Grivas and Chaloulakou 2006; Gualdron, Llobet et al. 2006). From the point of view of Feature Selection, which is a concept used in Pattern Recognition that implies reduction of the input dimensionality while at the same time retaining as much as possible of their class discriminatory information; these techniques can be classified into three broad categories: Sequential Backward Selection (SBS), Sequential Forward Selection (SFS), and stochastic selection (de 2001; Theodoridis and Koutroumbas 2003). SBS and SFS have been the subject of variable selection for many years (Gualdron, Llobet et al. 2006). Although the most frequently applied variable selection techniques so far are SBS and SFS, these two techniques are seldom used alone. This is because, according to Gardner, Boilot et al. (2005), they will only explore a small fraction of the whole set of configurations and can become trapped in local minima. Stochastic approaches, such as Genetic Algorithms (GAs) and simulated annealing, have

showed superiority over SBS/SFS and have many successful applications (Guo and Uhrig 1992; Weller, Summers et al. 1995; Alexandridis, Patrinos et al. 2005; Gardner, Boilot et al. 2005; Yang, Hines et al. 2007; Yang, Hines et al. 2008). For example, Gardner, Boilot et al. (2005) applied a modified GA to find a good subset of sensors within an array of 32 carbon-black polymer resistors to be used in Probabilistic Neural Network (PNN) classifiers. The methods were shown to be accurate and fast at determining the sensors that should be used to discriminate bacteria. In the context of predicting the longitudinal dispersion coefficient, a GA-based variable selection technique has been proposed in our previous works (Genetic Neural Mathematical Method, hereafter called GNMM) (Yang, Hines et al. 2007; Yang, Hines et al. 2007; Yang, Hines et al. 2008). In these works, GAs are utilized to identify variables that are being input into an MLP to predict longitudinal dispersion coefficient. The effectiveness of the method is demonstrated by means of case study data, which have previously been explored by other authors using various methods.

In fact, the GNMM method was developed in two phases. In the first phase, the input determination method was created and applied to a database that contained 196 data samples from 27 different rivers measured by the UK Environment Agency (EA). Using variables identified by our input determination method (2 out of a total of 49), we achieved an accuracy of longitudinal dispersion coefficient prediction of 0.72 for the coefficient of determination (R^2) and 11.66 for the Root Mean Square Error (RMSE) (Yang, Hines et al. 2007). In the second phase, mathematical programming-based rule extraction was incorporated. In this phase, a data set that contained 71 sets of measurements from 29 rivers in the United States was used in the case study. GNMM selected 3 variables out of 8 in the final training. As a result, an R^2 of 0.96 and RMSE of 34.85 were obtained (Yang, Hines et al. 2008). In addition, 13 regression rules were extracted from the MLP trained using the US data, which makes the training results potentially of greater utility to organizations such as the EA.

The aim of the current chapter is to provide an insightful analysis of our technique which uses GAs as an ANN input variable optimization tool in the context of longitudinal dispersion coefficient prediction. This is achieved by a detailed comparative study of GNMM's input determination process based on our previous works (Yang, Hines et al. 2007; Yang, Hines et al. 2008). Furthermore, Principal Component Analysis (PCA) and Self-Organizing Map

(SOM) analysis are performed to cross-validate the GA input variable selection results.

6.2 THE METHOD

We have presented a GA-based input determination method for ANNs, called GNMM, in the prediction of longitudinal dispersion coefficient in natural streams (Yang, Hines et al. 2007; Yang, Hines et al. 2007; Yang, Hines et al. 2008). This consists of three steps: 1) GA-based variable selection, 2) MLP training and 3) mathematical programming-based rule extraction. Let us consider each in turn:

- (1) Assume there are two data sets $\mathbf{X} = \{x_{(1,1)}, \dots, x_{(a,b)}\}$ and $\mathbf{Y} = \{y_1, \dots, y_a\}$, where $\mathbf{X}$ is the hydraulic/geometric measurements, $\mathbf{Y}$ is the corresponding longitudinal dispersion coefficient, a is the number of measurements that have been made and b denotes available variables. The GNMM process can be summarized as follows:

Randomly generate an initial population of chromosomes of size N_p . A chromosome consists of b genes, each representing an input variable. The encoding of a gene is binary, meaning that a particular variable is considered as an input variable (represented by '1') or not (represented by '0'). The assessment of the fitness of a chromosome is the Mean Squared Error (MSE) when a three-layer MLP is being trained with the input variable subset $\mathbf{X}_i$ and output target $\mathbf{Y}$ for a certain number of epochs N_e .

The GA input determination process is then realised by altering N_p , N_e , generation size N_g , crossover probability p_c and mutation probability p_m . As a result, the input variables which occur most frequently throughout all the populations can therefore be identified. The final subset formed by these variables $\mathbf{X}_f$ is the subset that produces the minimal error within a given number of epochs.

- (2) Taking $\mathbf{X}_f$ and $\mathbf{Y}$ as inputs and targets respectively, an MLP is trained using BP several times in order to minimize the error.
- (3) Let the activation function for all hidden layers be the 'hyperbolic tangent' (tanh)

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} = \frac{2}{1 + e^{-2x}} - 1 \quad (1)$$

and a linear function is used in the output layer. The following equation is used to approximate tanh:

$$g(x) = \begin{cases} 1 & x \geq \kappa \\ \beta_1 x + \beta_2 x^2 & 0 \leq x \leq \kappa \\ \beta_1 x - \beta_2 x^2 & -\kappa \leq x \leq 0 \\ -1 & x \leq -\kappa \end{cases} \quad (2)$$

in which $\beta_1 = 1.002$, $\beta_2 = -0.2510$, $\kappa = 1.9961$. Eq.(2) divides the input domain into four sub-domains. Therefore, once the training is complete, rules associated with the trained MLP can be derived.

This is a general description of how GNMM works. For a detailed description, see Yang, Hines et al. (2007; 2007; 2008).

6.3 GENETIC ALGORITHM – SOME THERORETICAL CONSIDERATIONS

The GA techniques have a solid theoretical foundation, which is based on the Schema Theorem (Holland 1975; Goldberg 1989). The theorem for a general case can be given as

$$M_H(t+1) \geq \left((1 - p_c) M_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} + p_c \left[M_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} (1 - losses) + gains \right] \right) (1 - p_m)^{O(H)} \quad (3)$$

Where $M_H(t+1)$ is the number of instances of the schema H in the generation $t+1$, $\hat{f}_H(t)$ is the average fitness of H in the generation t , $\hat{f}(t)$ is the average fitness of the population, $O(H)$ is the order of schema H , p_c and p_m are crossover and mutation probability respectively.

The GA technique used in GNMM is often referred to as the Simple Genetic Algorithm (SGA) (Vose 1999; Reeves and Rowe 2003). In the SGA, p_c , p_m and

the population size N_p are usually kept constant throughout. Also, for simplicity, *gains* are ignored and the conservative assumption is made that crossover falling in the significant portion of a schema always leads to disruption. Thus, Eq.(3) becomes

$$M_H(t+1) \geq \left((1-p_c)M_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} + p_c \left[M_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} (1 - \text{disruptions}) \right] \right) (1 - p_m)^{O(H)} \quad (4)$$

where *disruptions* overestimates losses. Let $P_H(t) = M_H(t)/N_p$ denote the proportional representation of H obtained by dividing $M_H(t)$ by the population size N_p . The probability that a randomly chosen mate samples H is just $P_H(t)$. Disruption is therefore given by:

$$\text{disruptions} = \frac{l_d}{l-1} (1 - P_H(t)) \quad (5)$$

where l and l_d are the length and the defining length of the schema H respectively. Substituting Eq.(5) into Eq.(4), we get

$$M_H(t+1) \geq \left((1-p_c)M_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} + p_c \left[M_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} \left(1 - \frac{l_d}{l-1} (1 - P_H(t)) \right) \right] \right) (1 - p_m)^{O(H)} \quad (6)$$

Dividing both sides of Eq.(6) by N_p and rearranging leads to the following expression of the Schema Theorem:

$$P_H(t+1) \geq P_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} \left(1 - p_c \frac{l_d}{l-1} (1 - P_H(t)) \right) (1 - p_m)^{O(H)} \quad (7)$$

This is the Schema Theorem that deals with SGAs, and also the theory behind the input variable determination method in GNMM. In the context of longitudinal dispersion coefficient prediction, Eq.(7) suggests that the search space is propelled towards the high fitness (i.e. better prediction) region by increasing/reducing the instances of a schema if its fitness is above/below the average. Eq.(7) also indicates that crossover and mutation helps preserve a reasonable level of population diversity, which enables the GA process to escape from sub-optimal regions of the solution space.

However, in GNMM as stated previously, the fitness value of schema H is determined as the MSE when an MLP is being trained with input variables corresponding to this binary schema (represented by '1's in H). Due to the randomness in the MLP initialization, there is uncertainty associated with $\hat{f}_H(t)$ in Eq.(7). This means that a schema that performs well in this generation may not survive into the next generation. This adds complexity to the situation, which is sometimes called dynamic optimization (Branke 2002; Kendall and Su 2004). In GNMM, the problem can be solved by applying two techniques: one is introducing an elite group into GAs (Reeves and Rowe 2003; Haupt and Haupt 2004); the other is what we call the *appearance percentage* (Yang, Hines et al. 2008).

The elite group is a collection of chromosomes that performed best and were made exempt from crossover and mutation and are retained in the next generation. Thus, Eq.(7) is modified as follows:

$$P_H(t+1) \geq P_H(t) \frac{\hat{f}_H(t)}{\hat{f}(t)} \left(1 - p_c \frac{l_d}{l-1} (1 - P_H(t) - P_e) \right) (1 - p_m)^{O(H)} \quad (8)$$

where $P_e = N_e/N_p$ is constant in a GA denoting the proportional representation of the elite group obtained by dividing the elite group size N_e by the population size N_p . In this way, the schema that performed best are protected and the rest of the population also crowds into high yielding regions of the search space. Introducing the elite group into GAs strengthens the ability to search, which can be explained as exploitative with respect to high yielding regions and explorative with respect to other regions.

Another important feature associated with the randomness of ANN is that later generations in a GA may not produce better results compared to earlier ones. This is because ANNs are initialized randomly, even if the same winning chromosomes are found in successive generations, they may not yield the same performance within a certain number of epochs. In this case the fitness value is not the only criteria for the evaluation of a schema. In Yang, Hines et al. (2008), we introduced the concept of *appearance percentage*. A gene's *appearance percentage* is defined as a gene's accumulated appearance in the winning chromosome of each generation divided by the total number of generations. Thus, although the GA's search evolves successively towards better generations, each generation is treated separately. From the point of view of schemata the *appearance percentage* is an averaged measure of the importance of one schema compared to others. Hence, the uncertainty associated with the randomness of the ANN is minimized. Due to the fact that in GNMM the coding is binary, correspondence between genes and input variables can be easily found.

6.4 GNMM'S INPUT DETERMINATION PROCESS

6.4.1 Data Pre-processing

Under some circumstances, the data are pre-processed in a stage before being processed by GNMM. Usually data pre-processing for ANNs may include scaling the inputs and targets so that they fall within a specified range. However, in GNMM the prior objective of data pre-processing is to reduce the dimensionality of the original set of inputs by eliminating redundant and/or dependant variables. This will result in a set of independent inputs that are not necessarily related to the dispersion coefficient. This subset of inputs can then be used in GNMM to determine which of these inputs are most appropriate for mapping to the output.

As mentioned previously in the introduction section, there are a total of 49 variables available in the UK EA database (Yang, Hines et al. 2007) (hereafter called Data I). In Data I, variables such as river name, flow exceedence/category, and start location grid reference are removed first of all. These variables are valuable in terms of dye tracing studies but do not provide useful information in the current context. Secondly, dependant variables are discarded. For example, start and end position of the river location elevation are discarded, while reach slope is kept; reach sinuosity is removed while reach length and straight distance are kept. After being processed, the EA data contains 16 variables which belong to 4 categories in the original dataset, e.g. start/end location, reach, and gauging station. These variables are: for the start/end location (subscripts s/e), catchment area (C), distance from injection point (D), theoretical mean flow (M), and theoretical Q95 flow (Q); for the reach in question, slope (S), reach length (L), and straight distance (D_r); for the gauging station, catchment area (C_g), average daily flow (A), daily mean flow (M_g), theoretical Q95 flow (Q_g), instant flow (I). All these variables are listed in Table 1.

Table 1: Variables in Data I and II

Data I (16 variables)		Data II (8 variables)	
Start location:	C_s (km ²), D_s (km) M_s (m ³ /s), Q_s (m ³ /s)	Independent:	B (m), H (m) U (m/s), u^* (m/s), α
End location:	C_e (km ²), D_e (km) M_e (m ³ /s), Q_e (m ³ /s)	Dependent:	B/H , U/u^* , β
Reach:	S , L (m), D_r (m)		
Gauging station:	C_g (km ²), A (m ³ /s) M_g (m ³ /s), Q_g (m ³ /s), I (m ³ /s)		

Also as described in the introduction section, the whole US dispersion data set, denoted by Data II, contains 8 variables apart from the longitudinal dispersion coefficient (Yang, Hines et al. 2008). There are five independent variables: channel width (B), flow depth (H), velocity (U), shear velocity (u^*) and river sinuosity (α = channel length/valley length). Dependant variables are width-to-depth ratio (B/H), relative shear velocity (U/u^*) and channel shape variable ($\beta = \ln(B/H)$). Data II variables are listed in Table 1 too.

It is worth noting that dependent variables exist in Data II (B/H , U/u^*), which indicates that data preprocessing is not always necessary in GNMM. Since the

aim of the data pre-processing in GNMM is to reduce the dimensionality of the data by eliminating redundant and/or dependant variables, obviously in Data II the dimensionality is not a main issue. On the other hand, as we will show later, Data II can be treated as an example of how GNMM handles dependent variables.

Table 2: Data I statistics

	Start Location				End Location			
	C_s (km ²)	D_s (km)	M_s (m ³ /s)	Q_s (m ³ /s)	C_e (km ²)	D_e (km)	M_e (m ³ /s)	Q_e (m ³ /s)
Max	3314.75	41.50	49.55	9.47	3315.25	46.50	49.55	9.47
Min	16.00	1.00	0.18	0.02	9.25	3.40	0.39	0.03
Avg	714.51	9.82	13.04	2.10	858.97	16.43	15.15	2.45
Avg _t	643.05	8.49	13.57	1.97	838.79	16.27	17.10	2.61
Avg _s	732.15	10.15	12.90	2.13	863.91	16.47	14.63	2.41

	Reach			Gauging Station				
	S	L (m)	D_r (m)	C_g (km ²)	A (m ³ /s)	M_g (m ³ /s)	Q_g (m ³ /s)	I (m ³ /s)
Max	0.0244	14697.0	12133.50	3314.80	47.14	75.00	6.60	75.00
Min	0.0000	1058.00	915.57	20.00	0.44	0.48	0.06	0.50
Avg	0.0023	6037.06	4342.88	792.39	12.38	10.08	1.93	10.20
Avg _t	0.0030	6775.73	4834.09	736.99	13.48	11.33	1.83	11.29
Avg _s	0.0022	5856.02	4222.49	805.96	12.11	9.78	1.95	9.93

6.4.2 Training and Testing Data

The first step in the GNMM input determination process is the division of the dataset into training and testing data. This is to avoid over fitting when chromosomes are being evaluated in an MLP (Lin and Lee 1996). The division is achieved by selecting representative sets for both of training and testing data.

- (1) Among the 196 data samples contained in Data I, some contain a high percentage of missing values, or indications that the data recorded is inaccurate. In order to obtain reliable results, these data are removed. As a result, the final data set contains 127 data samples. After division, the training subset, denoted by Data I_t, contains 102 samples; while the testing subset, denoted by Data I_s, consists of the remaining 25.
- (2) Similarly, Data II is divided into two sub-sets, Data II_t and Data II_s for training and testing respectively. Data II_t contains 49 data sets out of 71, while Data II_s consists of the remaining 22.

Normally when we have a large quantity of data we would typically use more data for training and less for testing. With small datasets we may repeat the process several times by randomly generating training and testing data to ensure that our results are reliable for the data set. Table 2 and Table 3 show statistics of these subsets respectively. Note that in these tables, Avg, Avg_t and Avg_s mean the average for the whole data set, training and testing subset average correspondingly.

Table 3: Data II statistics

	B (m)	H (m)	U (m/s)	u^* (m/s)	B/H	U/u^*	β	α	K (m ² /s)
Max	711.2	19.94	1.74	0.553	156.5	19.63	5.05	2.54	892.0
Min	11.9	0.22	0.03	0.002	13.8	1.29	2.62	1.08	1.9
Avg	83.0	1.70	0.54	0.088	51.7	7.62	3.79	1.39	107.7
Avg _t	62.9	1.31	0.49	0.084	51.4	7.13	3.79	1.37	98.4
Avg _s	127.6	2.55	0.66	0.097	52.4	8.72	3.77	1.42	128.4

6.4.3 GA Parameters

Selecting GA parameters is very difficult due to the many possible combinations in the algorithm. In addition, a GA relies on random number generators for creating the population, selection, crossover and mutation. A different random number seed produces different results. As such, selecting GA parameters is always problem-specific.

Table 4: GA parameters and CPU speeds/time

	Case	N_p	N_g	N_e (/ chromosome)	CPU Speed	CPU Time (s)
Data I	1	200	100	100	3.2 GHz (Pentium 4)	22760.52
	2	200	100	50	2.66 GHz (Celeron D)	34992.37
	3	400	100	100	900 MHz (UltraSPARC III)	139491.91
	4	400	200	20	3.2 GHz (Pentium 4)	34608.08
	5	400	400	20	900 MHz (UltraSPARC III)	216925.95
	6	200	200	20	2.66 GHz (Celeron D)	40111.13
	7	400	100	300	3.2 GHz (Pentium 4)	127634.52
Data II	1	200	100	100	3.20 GHz (Pentium 4)	17830.33
	2	200	100	50	2.66 GHz (Celeron D)	31772.62
	3	400	100	100	3.20 GHz (Pentium 4)	33600.17
	4	200	200	20	2.66 GHz (Celeron D)	45690.86
	5	200	200	100	3.20 GHz (Pentium 4)	39280.16

Generally speaking, large populations are used to thoroughly explore complicated cost surfaces. Crossover is then the operator of choice to exploit promising regions of phase space by combining information from promising solutions. Mutation is typically considered as a secondary operator of little importance – a mutation in the less critical genes may result in further exploitation of the current region. Schaffer, Caruana et al. (1989) have reported results on optimum parameter settings for a binary GA. They used discrete sets of parameter values $N_p = 10, 20, 30, 50, 100, 200$; $p_m = 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.10$; $p_c = 0.05$ to 0.95 in increments of 0.10 ; and 1 or 2 crossover points, which means that there were a total of 8400 possible combinations. Each combination was averaged over 10 independent runs. These authors found the best performance resulted for the following parameter settings: $N_p = 20$ to 30 , $p_c = 0.75$ to 0.95 , $p_m = 0.005$ to 0.01 . As we will see later on, parameter settings for GNMM fall into this range apart from the population size N_p – since in Data I we have 2^{15} possible solutions, our N_p is set much greater.

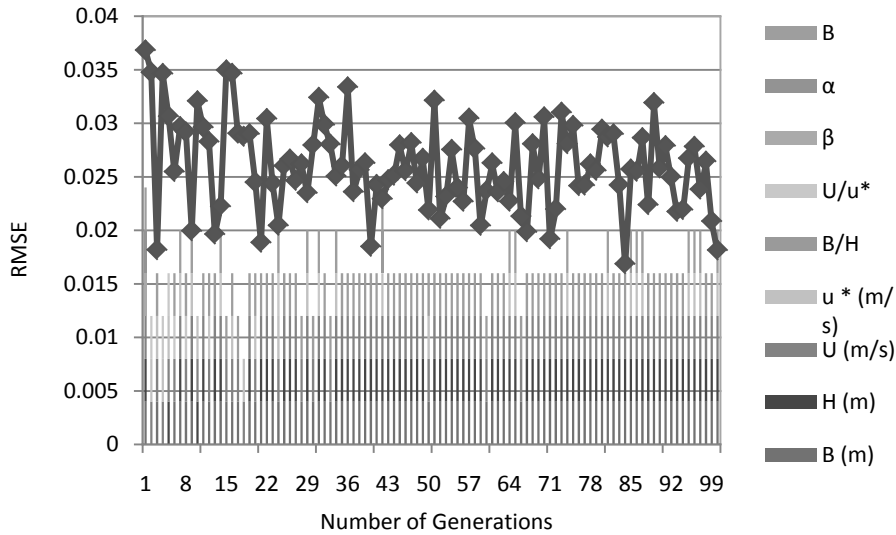


Figure 1: RMSE and winning variables for Case 2 of Data II

GNMM is implemented in MATLAB (v7.2), mainly using the Genetic Algorithm and Direct Search Toolbox, as well as the Neural Network Toolbox. GA parameters are set as follows: $p_c = 0.8$, $p_m = 0.01$, the learning rate $\alpha = 0.01$,

the elite group size $N_e = 2$. Other settings for each GA run are shown in Table 4, along with CPU speed and CPU time. It should be noted that N_e in Table 4 stands for ‘number of epochs per chromosome’.

6.4.4 The Elite Group and Appearance Percentage

As described previously, in order to minimize the randomness in the MLP initialization, we introduced the elite group and *appearance percentage* into GNMM. We will now consider these two techniques through the following example. For easier visual presentation purpose, Data II is chosen because it contains fewer variables than Data I. However, similar results can be found in all cases listed in Table 4.

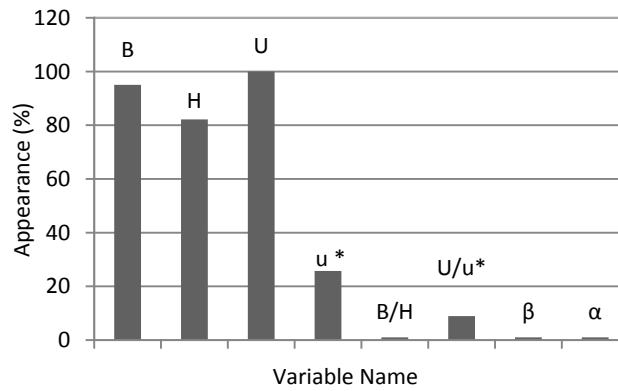


Figure 2: Appearance percentage for Case 2 of Data II

Figure 1 illustrates the RMSE of the winning chromosome and its component variables for each generation in Case 2 of Data II. It can be seen that although the overall trend of the RMSE is decreasing, it is not necessarily the case that later generations produce lower RMSE than earlier ones. This is a distinct feature of GNMM compared with other GA approaches. It is also evident from Figure 1 that the changing range of the RMSE is narrowing too, which implies that the GA has identified a high yielding region and was searching exploitatively. Several successive generations around the 50th yield the same winning chromosome. This is the effect of the elite group. As mentioned before,

chromosomes which performed best are protected so that they can compete in the next generation. Apparently, these chromosomes also won in the next generation.

Compared to Figure 1, which depicts component variables in each winning chromosome, Figure 2 shows some statistical information for all the variables by means of *appearance percentage*. It can be seen that the most frequently appearing variables are U (100%), B (95%) and H (82%), followed by u^* (26%), U/u^* (9%), whereas β , α and B/H are all less than 1%. These results seem promising; however, GNMM does not rely on a single run. This is because GA operations are based on random number generators. Once the random number seed changes, these results change accordingly. This also explains why there are 7 cases for Data I and 5 cases for Data II. The results are shown graphically in Figure 3 and Figure 4.

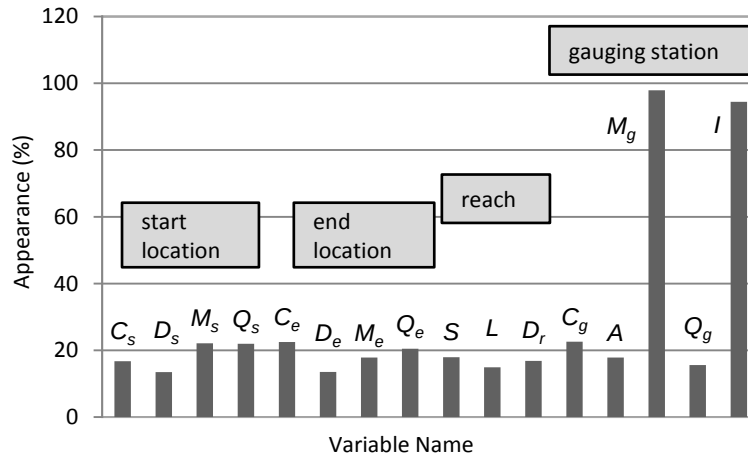


Figure 3: Appearance percentage for Data I

After running the GA seven times for Data I and five times for Data II, the *appearance percentage* of the input variables of these two data sets are shown in Figure 3 and Figure 4 respectively. As such, M_g and I for Data I and B , H and U for Data II are kept during the next MLP training stage in GNMM, while the rest of the variables are all removed. It is interesting to note that Figure 4 has a very similar distribution to Figure 2. The only difference is that the high *appearance percentage* shown in Figure 2 is slightly reduced in Figure 4; and vice versa for the low *appearance percentage*. This is the averaging effect of the *appearance*

percentage technique. It should also be noted that dependant variables B/H , U/u^* and β are automatically filtered out.

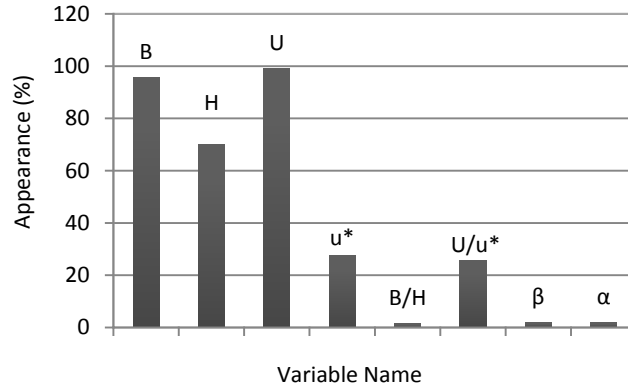


Figure 4: Appearance percentage for Data II

After the MLP training in the next stage of GNMM, briefly, the results for these two data sets are: $R^2 = 0.72$ and $RMSE = 11.66$ for Data I; $R^2 = 0.96$ and $RMSE = 34.85$ for Data II.

6.4.5 Rule Extractions

An important feature of GNMM is that by eliminating redundant input data, understanding complex models has been made simple. This can be illustrated by applying mathematical-programming based rule extraction to trained MLPs as discussed in the method section.

Rules extracted from trained MLPs for Data I and Data II are shown in Table 5 and Table 6 respectively, where the number of training and test data samples associated with each rule is also listed. Recall that in Eq.(2) the input domain of hidden neurons is divided into four sub-domains, this corresponds to the actual value of the digits in each rule ranging from 1 to 4. On the other hand, the length of each rule corresponds to the number of neurons in the hidden layer. Since 5 and 3 neurons were used in the hidden layer for Data I and Data II respectively, hence the corresponding length of each rule set is 5 and 3.

Regression rules summarised below give us an idea of the importance of each rule and the distribution of the data. These rules can also be written in the

antecedent/consequent format. For example, Rule 2 in Table 6, which is executed most of the time for both the training and test data in Data II, can be rewritten as

IF (input to neuron 1) ≥ 0.5939
 AND (input to neuron 2) ≥ 2.0104
 AND $2.1432 \leq (\text{input to neuron 3}) \leq 4.1393$
 THEN neural network output = $-1.7143 + 1.3519(\text{input to neuron 3}) - 0.3154(\text{input to neuron 3})^2$

Table 5 Rules fired for Data I

No.	Rule	Data I _t	Data I _s
1	24121	2	1
2	41342	2	
3	41442	55	13
4	42341	18	5
5	42342	8	
6	43241	7	2
7	43341		1
8	44131	3	
9	44141	3	1
10	44241	1	
11	44242	3	2

Table 6: Rules fired for Data II

No.	Rule	Data II _t	Data II _s
1	112	1	2
2	113	24	6
3	123	3	1
4	124	6	6
5	133		2
6	213	3	
7	214	2	1
8	223	5	
9	224	1	1
10	233	4	
11	431		1
12	441		1
13	442		1

These regression rules could provide environmental managers or field response teams with a rapid assessment tool for identifying values of the longitudinal dispersion coefficients required for the prediction of contaminant spread and concentrations immediately following an accidental spill.

6.5 DISCUSSIONS

In order to validate the effectiveness of GNMM's input determination method, comparisons between the prediction results of GNMM and other methods presented in the literature has been given in Yang, Hines et al. (2008). Particularly, the results for Data II have been validated against those in Tayfur and Singh (2005), in which the same data set was applied to an ordinary MLP using the BP algorithm. It was demonstrated that GNMM converges a little faster

($N_e = 19887$ vs. $N_e = 20000$), and more importantly it is much more accurate ($R^2 = 0.96$ vs. $R^2 = 0.90$). Although MLPs are being adopted in both applications, the difference lies in the fact that only a portion of available variables are used in GNMM instead of using them all as in Tayfur and Singh (2005).

In the following sections, Principal Component Analysis (PCA) and Self-Organizing Maps (SOMs) are applied to Data I and II to cross-validate the input variables identified by GNMM.

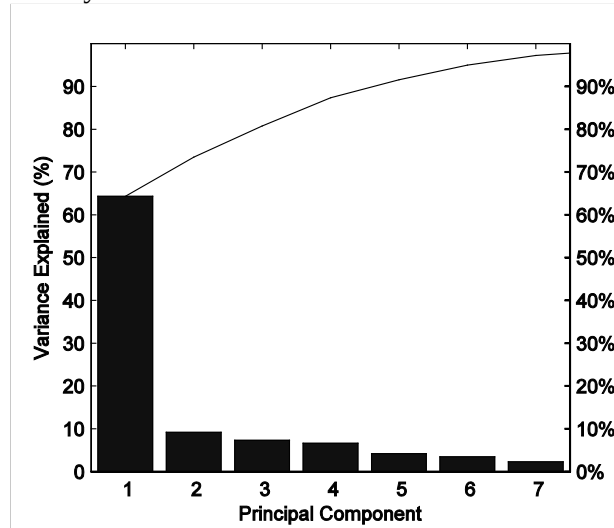


Figure 5: Percentage of the first 7 principal components in Data I

6.5.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a statistical technique used to transform a data space into a smaller space of the most relevant features (Hand, Mannila et al. 2001; Engelbrecht 2002). The aim is to project the original data space onto a linear subspace such that the variance in the data is maximally explained within the smaller subspace. Features (or inputs) that have little variance are thereby removed. The principal components of a data set are found by calculating the covariance (or correlation) matrix of the data patterns, and by selecting the minimal set of orthogonal vectors (the eigenvectors) that span the space of the Covariance matrix. Given the set of orthogonal vectors, any vector in the space can be constructed with a linear combination of the eigenvectors.

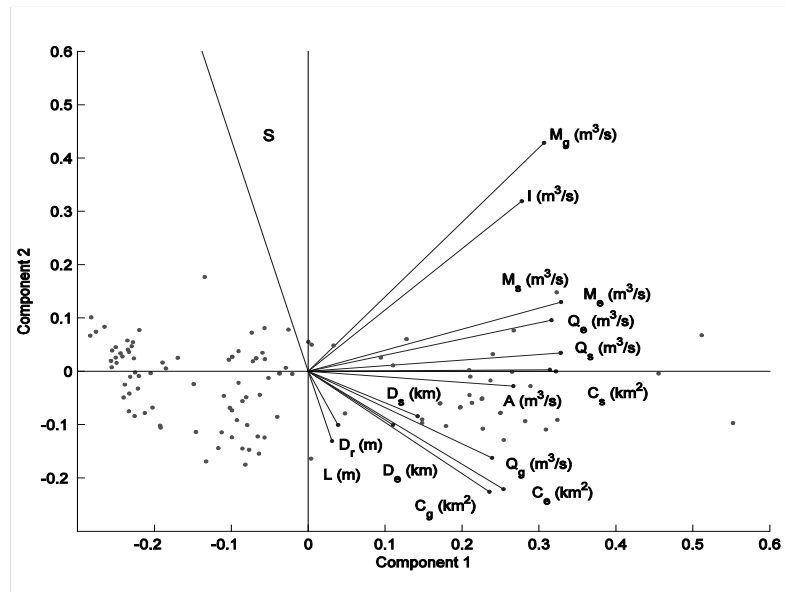


Figure 6: Projections of Data I points and variables onto the first two principal components

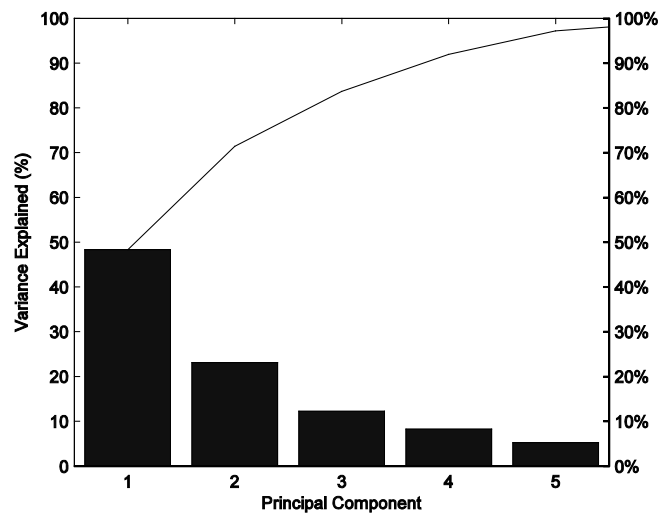


Figure 7 Percentage of the first 5 principal components in Data II

Figure 5 shows the percentage of the first 7 principal components in Data I. It can be seen that the only clear break in the amount of variance accounted for by each component is between the first and second components. However, the first component by itself explains more than 60% of the variance; with the second components, the variance explained is more than 70%. Therefore, it is reasonable to assert that the first two components can be regarded as being representative of Data I.

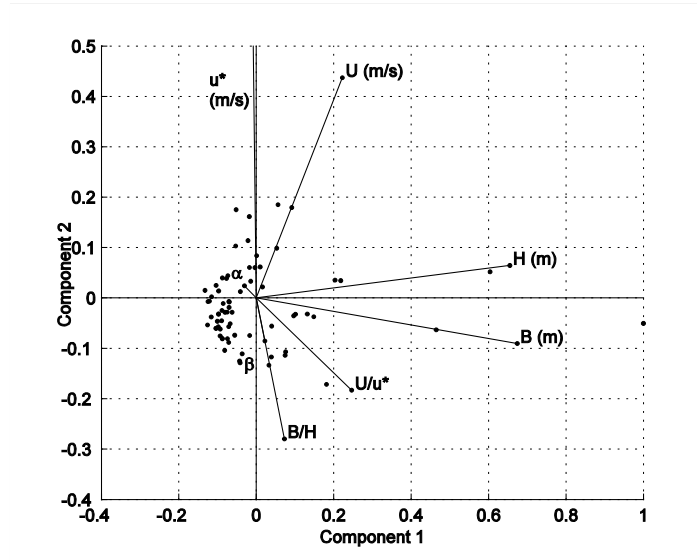


Figure 8: Projections of Data II points and variables onto the first two principal components

The projections of data samples and variables in Data I onto the first two principal components are depicted in Figure 6. It is interesting to note that the two variables selected by the GNMM (M_g and I) are clustered together although they are not outstanding in terms of contributions to the first principal component. Thus it may be appropriate to ask as to whether or not GNMM was working properly. It should be noticed that the dimensionality reduction achieved by PCA is realised by preserving as much of the relevant information from the original data as possible. From Figure 3, it can be seen that none of the variables apart from the two selected by GNMM has a high *appearance percentage*. Therefore, the judgement is that although the first principal component preserves most information in Data I, it contains little information about the longitudinal

dispersion coefficient. Moreover, the fact that M_g and I are grouped together is the most important finding in Figure 6, since it is not necessary for any of the principal components to preserve the information about the longitudinal dispersion.

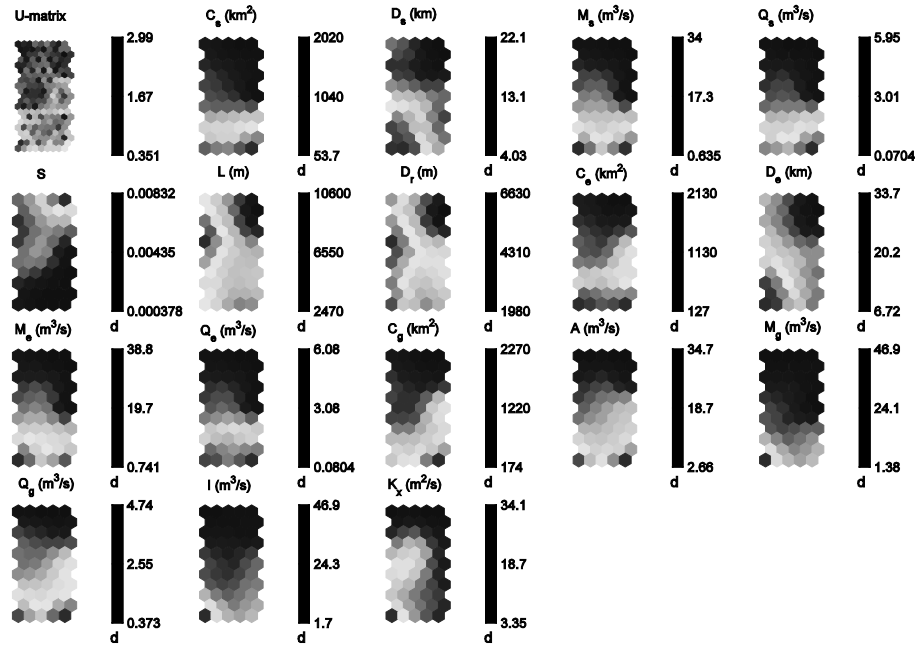


Figure 9 SOM analysis of Data I

Figure 7 shows the percentage of the first 5 principal components in Data II; while Figure 8 illustrates the projections of data samples and variables in Data II onto the first two principal components. It is evident that B and H are grouped together, and they are the most important variables in the first principal component, which counts for around 50% of the total variance. Following a similar analysis to Data I, it can be concluded that the first principal component presents most of the longitudinal dispersion information in Data II.

6.5.2 Self-Organizing Map (SOM)

The Self-Organizing Map (SOM) is a multidimensional scaling method to project an input space to a discrete output space, effectively performing a compression of input space onto a set of vectors. The output space is usually a two-dimensional

grid. The SOM uses the grid to approximate the probability density function of the input space, while still maintaining the topological structure of the input space. That is, if two vectors are close to one another in input space, so is the case for the map representation. For a detailed description of SOM, please refer to (Haykin 1994; Engelbrecht 2002). In the present study, the SOM analysis is performed using the MATLAB SOM Toolbox 2.0 developed at the Helsinki University of Technology (2005).

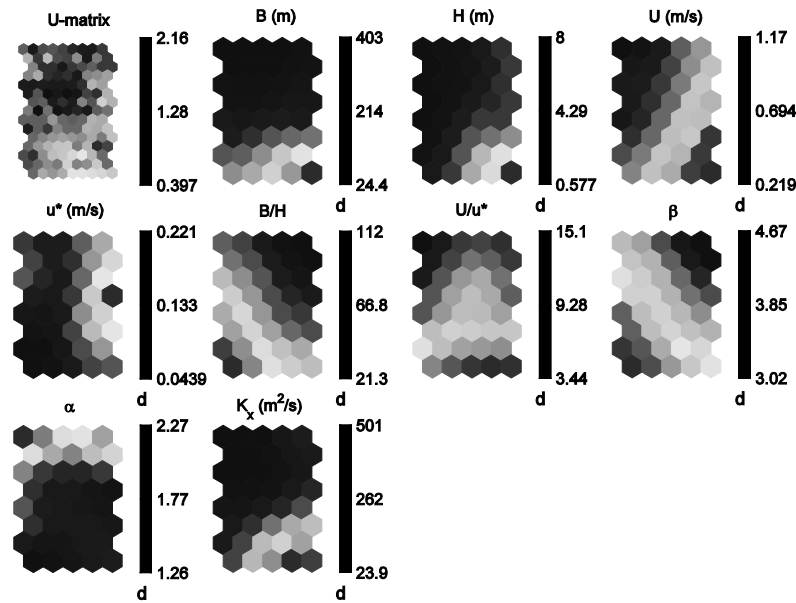


Figure 10: SOM analysis of Data II

The effect of the SOM training process is to cluster together similar patterns, while preserving the topology of the input space. Training results in a set of trained weights with no explicit cluster boundaries. An additional step is required to find these cluster boundaries. One way to determine and visualize the cluster boundaries is to calculate the unified distance matrix (U-matrix), which contains a geometrical approximation of the vector distribution in the map. The U-matrix expresses the distance to the neighbouring vectors for each neuron. Large values within the U-matrix indicate the position of cluster boundaries. Figure 9 shows the results of SOM analysis of Data I. In the U-matrix, a neuron's colour

represents the distance to its neuron neighbours – a low value indicates a small distance between neurons. The ‘component planes’ show what variable values the neurons have taken. This can be used to look for relationships between individual variables. These plots will have the same number of cells as there were neurons in the map. As each plot is a ‘slice’ of the output if two plots appear to have a similar distribution of values then this shows the variables to be related. Note that apart from the U-matrix and individual variables, Figure 9 also has the longitudinal dispersion coefficient (K_x), which is the training target. From Figure 9 it is clear that the pattern presented by the whole data set (U-matrix) is quite different from K_x , which corresponds to our analysis in the Principal Component Analysis (PCA) section. This also indicates that too much irrelevant information is contained in Data I. On the other hand, variables found by GNMM (M_g and I) have similar representations to the training target (K_x). This similarity validates our results for input determination for Data I.

Following similar steps as to the above, Figure 10 presents the results of SOM analysis for Data II. Unlike Figure 9, in Figure 10 the whole data set and the training target have similar distributions. This is illustrated by comparing patterns in the U-matrix and K_x . Furthermore, GNMM’s choices of input variables (B, H and U) all present these similar patterns.

6.6 CONCLUSIONS

In the current chapter, a GA-based input variable determination technique for ANNs (Genetic Neural Mathematical Method, GNMM) has been applied to two data sets. The first data set contains 196 data samples from 27 different rivers measured by the UK Environment Agency (EA). Using variables identified by this the input determination method (2 out of a total of 49), we achieved an accuracy of longitudinal dispersion coefficient prediction of 0.72 for the coefficient of determination (R^2) and 11.66 for the Root Mean Square Error (RMSE). The second data set contains 71 sets of measurements from 29 rivers in the United States. GNMM selected 3 variables out of 8. As a result, an R^2 of 0.96 and RMSE of 34.85 were obtained. Rules extracted from trained MLPs were also presented, which not only demonstrate the importance of each rule and the distribution of the data, but also provide environmental managers or field response teams with a rapid assessment tool for identifying values of the

longitudinal dispersion coefficients required for the prediction of contaminant spread and concentrations immediately following an accidental spill.

In the current chapter we provide an insightful analysis of the technique that uses GAs as an ANN input variable optimization tool in the context of longitudinal dispersion coefficient prediction. This is achieved by a detailed comparative study of the GNMM's input determination process. Moreover, Principal Component Analysis (PCA) and Self-Organizing Map (SOM) analysis are performed to cross-validate the results of GA input variable selection.

References

- Alexandridis, A., P. Patrinos, et al. (2005). "A two-stage evolutionary algorithm for variable selection in the development of RBF neural network models." *Chemometrics and Intelligent Laboratory Systems* 75(2): 149-162.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford ;; New York, Clarendon Press ;; Oxford University Press.
- Bowden, G. J., G. C. Dandy, et al. (2005). "Input determination for neural network models in water resources applications. Part 1--background and methodology." *Journal of Hydrology* 301(1-4): 75-92.
- Boxall, J. B. and I. Guymer (2007). "Longitudinal mixing in meandering channels: New experimental data set and verification of a predictive technique." *Water Research* 41(2): 341-354.
- Branke, J. (2002). *Evolutionary optimization in dynamic environments*. Boston, Kluwer Academic Publishers.
- de, S. J. P. M. (2001). *Pattern recognition : concepts, methods, and applications*. Berlin ;; New York, Springer.
- Deng, Z. Q., V. P. Singh, et al. (2001). "Longitudinal dispersion coefficient in straight rivers." *Journal of Hydraulic Engineering* 127(1): 919-927.
- Engelbrecht, A. P. (2002). *Computational intelligence : an introduction*. Chichester, England ; Hoboken, N.J., J. Wiley & Sons.
- Gardner, J. W., P. Boilot, et al. (2005). "Enhancing electronic nose performance by sensor selection using a new integer-based genetic algorithm approach." *ISOEN 2003 - Selected Papers from*

- the 10th International Symposium on Olfaction and Electronic Noses. 106(1): 114-121.
- Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Reading, Mass., Addison-Wesley Pub. Co.
- Grivas, G. and A. Chaloulakou (2006). "Artificial neural network models for prediction of PM10 hourly concentrations, in the Greater Area of Athens, Greece." *Atmospheric Environment* 40(7): 1216-1229.
- Gualdron, O., E. Llobet, et al. (2006). "Coupling fast variable selection methods to neural network-based classifiers: Application to multisensor systems." *Sensors and Actuators B: Chemical* 114(1): 522-529.
- Guo, Z. and R. E. Uhrig (1992). Using genetic algorithms to select inputs for neural networks. 92TH0435-8), Baltimore, MD, USA, IEEE Comput. Soc. Press.
- Hand, D. J., H. Mannila, et al. (2001). Principles of data mining. Cambridge, Mass., MIT Press.
- Haupt, R. L. and S. E. Haupt (2004). Practical genetic algorithms. Hoboken, N.J., John Wiley.
- Haykin, S. S. (1994). Neural networks : a comprehensive foundation. New York, Maxwell Macmillan International.
- Holland, J. H. (1975). Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. Ann Arbor, University of Michigan Press.
- Jobson, H. E. (1997). "Predicting travel time and dispersion in rivers and streams." *Journal of Hydraulic Engineering* 123(11): 971-978.
- Kashefipour, S. M., R. A. Falconer, et al. (2002). Modeling longitudinal dispersion in natural channel flows using ANNs. Lisse, the Netherlands, A.A. Balkema/Swets & Zeitlinger.
- Kendall, G. and Y. Su (2004). Learning with imperfections - A multi-agent neural-genetic trading system with differing levels of social learning, Singapore, Institute of Electrical and Electronics Engineers Inc., New York, NY 10016-5997, United States.
- Lin, C. T. and C. S. G. Lee (1996). Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. NJ, USA, Prentice-Hall, Inc. Upper Saddle River.
- Maier, H. R. and G. C. Dandy (1998). "Understanding the behaviour and optimising the performance of back-propagation neural networks: an empirical study." *Environmental Modelling and Software* 13(2): 179-191.
- Maier, H. R. and G. C. Dandy (2000). "Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications." *Environmental Modelling and Software* 15(1): 101-124.
- Piotrowski, A., P. M. Rowinski, et al. (2006). Assessment of longitudinal dispersion coefficient by means of different neural networks. 7th International Conference on Hydroinformatics, Nice, FRANCE.
- Ramadan, Z., X.-H. Song, et al. (2001). "Variable selection in classification of environmental soil samples for partial least square and neural network models." *Analytica Chimica Acta* 446(1-2): 231-242.
- Reeves, C. R. and J. E. Rowe (2003). Genetic algorithms : principles and perspectives : a guide to GA theory. Boston, Kluwer Academic Publishers.
- Rowinski, P. M., A. Piotrowski, et al. (2005). "Are artificial neural network techniques relevant for the estimation of longitudinal dispersion coefficient in rivers?" *Hydrological Sciences Journal* 50(1): 175-187.
- Schaffer, J. D., R. A. Caruana, et al. (1989). "A study of control parameters affecting online performance of genetic algorithms for function optimization." *Proceedings of the third*

- international conference on Genetic algorithms: 51-60.
- Seo, I. W. and T. S. Cheong (1998). "Predicting longitudinal dispersion coefficient in natural streams." *Journal of Hydraulic Engineering* 124(1): 25-32.
- Tayfur, G. (2006). "Fuzzy, ANN, and regression models to predict longitudinal dispersion coefficient in natural streams." *Nordic Hydrology* 37(2): 143-164.
- Tayfur, G. and V. P. Singh (2005). "Predicting longitudinal dispersion coefficient in natural streams by artificial neural network." *Journal of Hydraulic Engineering* 131(11): 991-1000.
- Taylor, G. (1954). "The Dispersion of Matter in Turbulent Flow through a Pipe." *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 223(1155): 446-468.
- The Helsinki University of Technology. (2005). 2007, from <http://www.cis.hut.fi/projects/somtoolbox/>.
- Theodoridis, S. and K. Koutroumbas (2003). *Pattern recognition*. Amsterdam ; Boston, Academic Press.
- Vose, M. D. (1999). *The simple genetic algorithm : foundations and theory*. Cambridge, Mass., MIT Press.
- Wallis, S. G. and J. R. Manson (2004). "Methods for predicting dispersion coefficients in rivers." *Water Management* 157(3): 131-141.
- Weller, P. R., R. Summers, et al. (1995). Using a genetic algorithm to evolve an optimum input set for a predictive neural network. *Proceedings of 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sheffield, UK, IEE.
- Yang, J., E. L. Hines, et al. (2007). GNMM and accurate longitudinal dispersion coefficient prediction. the seventh UK CARE Annual Meeting, ISBN-13: 978-0-9551965-2-2. Greenwich, London.
- Yang, J., E. L. Hines, et al. (2007). Prediction of Longitudinal Dispersion Coefficient in Rivers using Variables Identified by Genetic Algorithms. *The Fifth International Symposium on Environmental Hydraulics (ISEH V)*. Tempe, Arizona.
- Yang, J., E. L. Hines, et al. (2008). A Genetic Algorithm-Artificial Neural Network Method for the Prediction of Longitudinal Dispersion Coefficient in Rivers. *Advancing Artificial Intelligence through Biological Process Applications*. A. Porto, A. Pazos and W. Buño, Idea Group Inc.
- Yao, X. (1999). "Evolving artificial neural networks." *Proceedings of the IEEE* 87(9): 1423-1447.

Chapter 7

Kernel Machines in Remote Sensing Image Classification

Gustavo Camps-Valls, J. Muñoz-Marí, L. Gómez-Chova and J. Calpe-Maravilla

7.1 INTRODUCTION

Materials in a scene reflect, absorb, and emit electromagnetic radiation in a different way depending of their molecular composition and shape (Richards, Jia, 1999). Remote sensing exploits this physical fact and deals with the acquisition of information about a scene (or specific object) at a short, medium or long distance. The radiation acquired by an (airborne or satellite) sensor is measured at different wavelengths and the resulting spectral signature (or spectrum) is used to identify a given material.

Remote sensing imaging instruments can be considered to comprise of two broad types. (i) Passive optical remote sensing relies on solar radiation as the main source of illumination. The signal which is measured at the satellite, by an imaging spectrometer, is the emergent radiation from the Earth-atmosphere system in the observation direction. (ii) On the other hand, in active synthetic aperture radar (SAR) imaging, microwave pulses are transmitted by an antenna towards the Earth's surface and the energy scattered back to the satellite is measured.

The use of remotely sensed images for Earth Observation is nowadays a consolidated technology since the relatively high number of acquired spectral bands, along with the commonly computed spatial/contextual/textural features, allows the characterization, identification, and classification of the land-covering (Richards, Jia, 1999). However, traditional classifiers such as Gaussian maximum likelihood or neural networks are affected by high input sample dimension, tend to overfit data in the presence of noise (such as thermal noise in

optical images, or speckle in SAR images), or perform poorly when a low number of training samples are available (Hughes, 1968, Fukunaga, Hayes, 1989).

In the last few years, the use of support vector machines (SVMs) (Vapnik, 1998, Schölkopf, Smola, 2002) for hyperspectral image classification has been of interest. The method integrates into the same classification procedure: (i) a *feature extraction* step, as samples are mapped to a higher dimensional space where a simpler (linear) classification is performed, becoming non-linear in the input space; (ii) a *regularization* procedure by which model's complexity is efficiently controlled; and (iii) the minimization of an upper bound of the generalization error, thus following the Structure Risk Minimization (SRM) principle. These theoretical properties make the SVM in particular, and kernel methods in general, very attractive in the context of remote sensing image classification where the classifier commonly deals with a low number of high dimensional training samples (Camps-Valls, Gómez-Chova, et al., 2004, Camps-Valls, Bruzzone, 2005, Camps-Valls, Rojo-Álvarez, et al. (Eds.), 2006).

Nevertheless, SVMs must be adapted to the specific needs of the problem in hand. Note for instance that in a pixel-based SVM classifier the spatial (contextual) information is not taken into account. Therefore, inclusion of contextual information in the classifier is necessary to produce spatially homogeneous classification maps. Other sources of information are commonly available and should be incorporated in the classifier; images acquired by different sensors over the same scene could be used together so methods which consider the use of information fusion could improve the results. Also, a given sensor can acquire information over a given area to be monitored in a temporal sense so, by considering the (possibly non-linear) correlations among temporal images, classification accuracy could be improved. In all these problems, heterogeneous information fusion is a key aspect to bear in mind to correctly address the problem. In this chapter, we present some novel kernel-based formulations and results that address remote sensing information fusion of multi-temporal, multi-source and contextual/textural features through the use of composite kernels. Also, the particular problem of change detection is addressed by *kernelizing* standard algorithms.

Besides this, remote sensing image classification is hampered by both the number and quality of labeled training samples. In order to alleviate this problem, SVMs (or any other kernel-based classifier) should exploit the information contained in the wealth of unlabeled samples along with the low number of

labeled samples and thus work as a *semi-supervised learning* paradigm. Essentially, in semi-supervised learning (SSL), the algorithm is provided with some available supervised information in addition to the unlabeled data. The SSL framework has become popular and has recently attracted a considerable amount of research in machine learning (Chapelle, Schölkopf, Zien, 2006). In this chapter, we present some experimental evidence of the use of the Laplacian SVM (Belkin, Niyogi, Sindhwani, 2006) for the classification of challenging remote sensing problems.

Another different concern is that a complete and representative training set is essential for a successful classification. In particular, it is noteworthy that little attention has been paid to the case of having an incomplete knowledge of the classes present in the scene which is being investigated. This may be critical since, in many applications, acquiring ground truth information for all classes is very difficult, especially when complex and heterogeneous geographical areas are to be analyzed. In this chapter, we introduce the *one-class SVM* for remotely-sensed image classification with incomplete training data. This method is a recent kernel-based development that only considers samples belonging to the class of interest in order to learn the underlying data class distribution.

The rest of this chapter is outlined as follows. Section 2 presents a brief introduction to kernel methods and reviews their basic properties. Section 3 is devoted to the review of several standard kernel-based methods and new developments for remote sensing applications; in the context of supervised, semi-supervised, and one-class classification paradigms. Besides this, specific kernel methods for multi-temporal and multi-sensor data fusion are presented, and kernel-based change detection algorithms are further presented. Extensive experimental results are presented in Section 4. In Section 5, we conclude the chapter with a discussion and some final remarks.

7.2 AN INTRODUCTION TO KERNEL METHODS

Kernel methods offer a general framework for machine learning problems. In this section, we briefly review the fundamentals of kernel methods and revise the formulations of the kernel methods that we focus on here.

7.2.1 Background to kernels methods

When using linear algorithms, well-established theory and efficient methods are often available. Kernel methods exploit this fact by embedding the dataset S defined over the input or attribute space X ($S \subseteq X$) into a higher (possibly infinite) dimensional Hilbert space H , or *feature space*, and then they build a linear algorithm therein, resulting in an algorithm which is nonlinear with respect to the input data space. The mapping function is denoted as $\phi: X \rightarrow H$. Though linear algorithms will benefit from this mapping because of the higher dimensionality of the feature space, the computational load would dramatically increase because we should compute and deal with sample coordinates in that high dimensional space. This computation is avoided through the use of the *kernel trick* by which, if an algorithm can be expressed with dot products in the input space, its (non-linear) kernel version only needs the dot products among *mapped samples*.

For instance, by defining a mapping $\phi: \mathbb{R} \rightarrow \mathbb{R}^3$ such that $\phi(x) = \{1, \sqrt{2}x, x^2\}$ where $x \in \mathbb{R}$, we can spread samples in the *feature space* which is being *mapped*, which facilitates the (linear) classification. Note, however, that by doing this, the classifier needs to know the sample coordinates in this new space. This may be problematic especially if the new feature space needs to be large to allow correct classifications. Fortunately, by computing the dot product $\mathbf{K}_{ij} = K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \{1, \sqrt{2}x_i, x_i^2\} \cdot \{1, \sqrt{2}x_j, x_j^2\} = (1 + \langle x_i, x_j \rangle)^2$, there is no need to know the mapping ϕ explicitly, as computing distances in the feature space is expressed as a function of *only* the data in the original space. Therefore, we can work implicitly in a very high dimensional feature space by just replacing original inner products of the original samples with inner products among mapped samples by the proper choice of a kernel function K .

7.2.2 Properties of Kernels

The bottleneck for any kernel method is the definition of a kernel function K that accurately reflects the similarity among samples. However, not all metric distances are permitted. In fact, valid kernels are only those fulfilling Mercer's Theorem (Mercer, 1905, Aizerman, Braverman et al., 1964) and the most common ones are the linear $K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle$, the polynomial $K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d$, $d \in \mathbb{Z}^+$, and the radial basis function (RBF), $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2 / 2\sigma^2)$, $\sigma \in \mathbb{R}^+$. Let us now review the form of the kernel-based algorithms used in this work.

We should mention here than one can build new kernels from already built ones. Given two Mercer's kernels K_1 and K_2 defined on $S \times S$, $\mathbf{A}$ a symmetric positive *semi-definite* $n \times n$ matrix, and $\mu > 0$, then these are valid kernels:

$$K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) + K_2(\mathbf{x}, \mathbf{z}) \quad (1)$$

$$K(\mathbf{x}, \mathbf{z}) = \mu K_1(\mathbf{x}, \mathbf{z}) \quad (2)$$

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T \mathbf{A} \mathbf{z} \quad (3)$$

7.3 KERNEL-BASED METHODS

This section reviews the standard formulation of some kernel-based classifiers, and presents details of the novel kernel machines that have been developed for remote sensing applications.

7.3.1 Support Vector Classifier

Given a labeled training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$, and given a nonlinear mapping $\phi(\cdot)$, the support vector classifier (SVC) method solves:

$$\min_{\mathbf{w}, \xi_i, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right\} \quad (4)$$

constrained to:

$$y_i (< \phi(\mathbf{x}_i), \mathbf{w} > + b) \geq 1 - \xi_i \quad \forall i = 1, \dots, n \quad (5)$$

$$\xi_i \geq 0 \quad \forall i = 1, \dots, n \quad (6)$$

where $\mathbf{w}$ and b define a linear classifier in the feature space, and ξ_i are positive slack variables enabling us to deal with permitted errors (Figure 1(a)). Appropriate choice of non-linear mapping ϕ guarantees that the transformed samples are more likely to be linearly separable in the (higher dimension) feature space. Parameter C controls the generalization capabilities of the classifier, and it must be selected by the user. The primal problem (4) is solved using its dual problem counterpart (Schölkopf, Smola, 2002), and the decision function for any test vector $\mathbf{x}_*$ is given by

$$f(\mathbf{x}_*) = \text{sgn} \left(\sum_{i=1}^n y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_*) + b \right) \quad (7)$$

where α_i are Lagrange multipliers corresponding to constraints in (4), and b can be easily computed from a few support vectors (SVs), which are those training samples $\mathbf{x}_i$ with non-zero Lagrange multipliers α_i (Schölkopf, Smola, 2002).

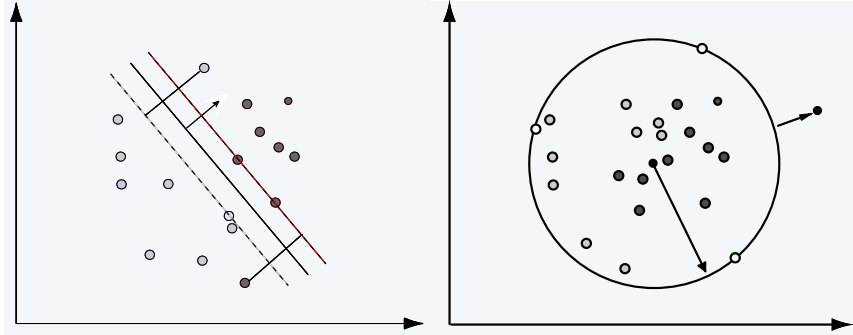


Figure 1: Kernel classifiers. (a) SVC: Linear decision hyperplanes in a non-linearly transformed space, where *slack* variables ξ_i are included to deal with errors. (b) SVDD: The hypersphere containing the target data is described by center a and radius R . Samples in the boundary and outside the ball are unbounded and bounded support vectors, respectively.

7.3.2 One-Class Support Vector Machine

A different problem statement for classification is given by the one-class support vector machine algorithm (also known as support vector domain description, SVDD.) Now, let $\{\mathbf{x}_i\}_{i=1}^n$ be a dataset belonging to a given *class of interest*. The purpose is to find a minimum volume *hypersphere* in a high dimensional feature space H , with radius $R > 0$ and center $\mathbf{a} \in H$, which contains most of these data objects (Tax, D., Duin, R. P., 1999) (Figure 1(b)). Since the training set may contain outliers, we introduce a set of *slack variables* $\xi_i \geq 0$, and the problem then becomes:

$$\min_{R, \mathbf{a}} \left\{ R^2 + C \sum_{i=1}^n \xi_i \right\} \quad (8)$$

constrained to

$$\|\phi(\mathbf{x}_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i \quad \forall i = 1, \dots, n \quad (9)$$

$$\xi_i \geq 0 \quad \forall i = 1, \dots, n \quad (10)$$

C controls the trade-off between the volume of the hypersphere and the permitted errors. One can define the rejection fraction parameter to be tuned as $\nu = 1/nC$, as noted in Schölkopf, Williamson, et al. (1999).

The dual functional is a quadratic programming problem that yields a set of α_i corresponding to Lagrange multipliers of (9). These allow us to calculate the distance from a test point to the center $R(\mathbf{x}_*)$, i.e.,

$$R(\mathbf{x}_*) = K(\mathbf{x}_*, \mathbf{x}_*) - 2 \sum_{i=1}^n K(\mathbf{x}_i, \mathbf{x}_*) + \sum_{i,j=1}^n K(\mathbf{x}_i, \mathbf{x}_j) \quad (11)$$

which is to be compared against ratio R . Unbounded SVs are those samples $\mathbf{x}_i$ satisfying $0 < \alpha_i < C$, while bounded SVs are samples whose associated $\alpha_i = C$, and they are considered outliers.

7.3.3 Laplacian Support Vector Machine

The Laplacian SVM (LapSVM) incorporates the information of the labeled and unlabeled samples in the formulation of the standard SVM. This is achieved by (i) expanding the solution over both labeled and unlabeled samples with a proper representer theorem, and (ii) including a regularizer over the unlabeled distribution of samples which is modeled with a graph (Belkin, Niyogi et al., 2006). The nodes in the graph represent both labeled and unlabeled samples connected by weights W_{ij} (Chapelle, Schölkopf et al. 2006).

Notationally, we are now given a set of l labeled samples, $\{\mathbf{x}_i, y_i\}_{i=1}^l$, and a set of u unlabeled samples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, where $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$. Then, the minimizing functional in (4) becomes:

$$\min_{\substack{\xi \in \mathbb{R}^l \\ \alpha \in \mathbb{R}^{l+u}}} \left\{ \frac{1}{l} \sum_{i=1}^l \xi_i + \gamma_L \alpha^T \mathbf{K} \alpha + \gamma_M (l+u)^2 \alpha^T \mathbf{K} \mathbf{L} \mathbf{K} \alpha \right\} \quad (12)$$

constrained to:

$$y_i \left(\sum_{j=1}^{l+u} \alpha_j K(x_i, x_j) + b \right) \geq 1 - \xi_i \quad (13)$$

$$\xi_i \geq 0 \quad (14)$$

γ_L controls the complexity of the decision function $f(\mathbf{x}_*) = \langle \mathbf{w}, \phi(\mathbf{x}_*) \rangle + b$ in the associated Hilbert space H , and γ_M controls its complexity in the intrinsic geometry of the marginal data distribution. Regularizing the graph is equivalent to penalizing the rapid changes of the classification function evaluated between close samples in the graph Laplacian, $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{W}$ given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$ and $D_{ij} = 0$ for $i \neq j$. This is a coupled *quadratic programming* problem which yields a solution expanded over labeled and unlabeled samples. Finally note that, by optimizing parameters γ_L and γ_M over a wide-enough range, the LapSVM theoretically outperforms several unsupervised and semi-supervised classifiers (Belkin, Niyogi, et al., 2006). The use of LapSVM is particularly interesting in practical remote-sensing applications (Gómez-Chova, Camps-Valls, et al., 2008) since obtaining additional unlabeled samples is cheap and easy, while labeling is expensive, difficult, or even impossible.

Table 1: Formulation of the composite kernel framework: input vector definition and the associated non-linear kernel transformations.

Kernel	Feature vector	Eq.
Stacked	$\phi(\mathbf{x}) = \phi(\mathbf{x})$	(15)
Direct sum	$\phi(\mathbf{x}) = \bigcup_{p=1}^P \{ \mathbf{A}_p \varphi_p(\mathbf{x}^{(p)}) \}$	(16)
Weighted	$\phi(\mathbf{x}) = \bigcup_{p=1}^P \sqrt{\mu_p} \{ \mathbf{A}_p \varphi_p(\mathbf{x}^{(p)}) \}$	(17)
Cross-info	$\phi(\mathbf{x}) = \bigcup_{p=1}^P \{ \mathbf{A}_p \varphi_p(\mathbf{x}^{(p)}) \} \bigcup_{p=1}^P \{ \mathbf{B}_p \varphi_p(\mathbf{x}^{(p)}) \}$	(18)
Difference	$\phi(\mathbf{x}^{(t_0)}) = \mathbf{A}_{t_0} \varphi(\mathbf{x}^{(t_0)}) - \mathbf{A}_{t_0-1} \varphi(\mathbf{x}^{(t_0-1)})$	(20)
Ratio	$\phi(\mathbf{x}^{(t_0)}) = \{ (\sqrt{\gamma} \mathbf{A}_{t_0} \varphi(\mathbf{x}^{(t_0)}))^* , (\mathbf{A}_{t_0-1} \varphi(\mathbf{x}^{(t_0-1)}))^* \}^* \sqrt{\langle \mathbf{A}_{t_0} \varphi(\mathbf{x}^{(t_0)}), \mathbf{A}_{t_0} \varphi(\mathbf{x}^{(t_0)}) \rangle}$	(21)

7.3.4 Kernels for Multi-temporal Image Classification

In this section, we present a novel methodological framework for information fusion based on kernels. This composite kernel framework allows us to easily combine different information sources $p=1,...,P$ as summarized in Table 1 and can be useful for multi-sensor or multi-temporal image classification and change detection problems (Camps-Valls, Gómez-Chova et al., 2006a, Camps-Valls, Gómez-Chova et al., 2008). Properties of kernels in the previous section are exploited here.

Let us now assume a multi-temporal set of labeled training samples (pixels) at a time t , $\{\mathbf{x}_i^{(t)}\} \in \mathbb{R}^N$, and their corresponding output labels $\{y_i^{(t)}\} \in \mathcal{Z}$, where $i=1,...,n$, and $t=1,...,t_o-1$ or $t=1,...,t_o$ depending on the availability of labeled data at the prediction time, t_o (see Figure 2). Let us define $T=\max(t)$. An important assumption in the following is that images at subsequent dates are co-registered, so that, from a machine learning perspective, pixels $\{\mathbf{x}_i^{(t)}\}$ are different (temporal) views of the same object or *pixel entity* $\mathbf{x}_i$. Also, let $\Omega=\{\omega_1,...,\omega_{N_C}\}$ be the set of N_C classes that characterize the geographical area at any time, thus assuming that the spatial distribution of such classes changes, but their number does not. In standard and operational situations, it is fair to assume that the number of classes of interest is commonly pre-specified by the user.

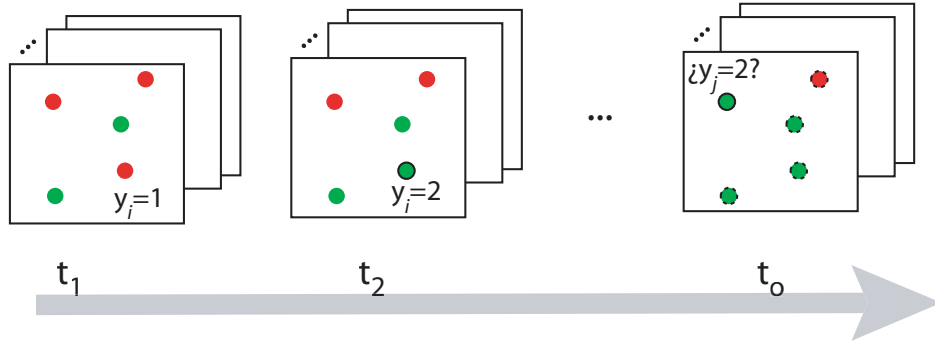


Figure 2: Scheme for multi-temporal classification and change detection. The problem of multi-temporal classification consists of classifying a given pixel j at time t_o , $\mathbf{x}_j^{(t_o)}$, using all available information ($t \leq t_o$), while change detection tries to identify if pixel's class label changed or not.

7.3.4.1 Stacked input vectors kernel

The most common approach to exploit the multi-temporal information is to stack vectors at different time instants in order to predict sample label at t_o . The composite input vector is here given by $\mathbf{x}_i \equiv \bigcup_{t=1}^T \mathbf{x}_i^{(t)}$, which yields the stacked kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \quad (15)$$

However, though this straightforward approach to data merging can yield good performance with respect to previously proposed methods, it does not include explicit cross-relations between samples at different time instants, $\mathbf{x}_i^{(t)}$.

7.3.4.2 The direct summation kernel

A simple composite kernel combining the *static* available information comes from the concatenation of nonlinear transformations for each $\mathbf{x}_i^{(t)}$ defined in Table 1, which can be easily computed as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^T K_t(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) \quad (16)$$

Note that this composite kernel basically sums the similarities of samples at each time instant individually, while avoiding the stacking of features. Again, no temporal correlation among pixels in different images is included in the classifier.

7.3.4.3 The weighted summation kernel

By exploiting kernel properties, a composite kernel that balances the temporal content in (10) can also be created, as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^T \mu_t K_t(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}), \quad (17)$$

where μ_t is a function assigning different weights to each time-dependent kernel, and can be either estimated from the data or fixed by the user. A good choice is an exponential decay, this is, $\mu_t = \lambda^{-(t_o-t)}$, $\lambda \in (0,1)$.

7.3.4.4 The cross-information kernel

In order to account for the cross relationship among subsequent time instants, we can use the cross information kernel in Table 1, which yields:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^T K_t(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) + \sum_{t,t'=1}^T K_{t,t'}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t')}) \quad (18)$$

Note that this is a complex composite kernel that contains the cross-information among all possible kernel matrices computed at different time instants. It is easy to show that this general equation can be simplified for the case of considering correlation only for subsequent time instants t and $t+1$, and then the composite kernel takes the form:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{T-1} [K_t(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) + K_t(\mathbf{x}_i^{(t+1)}, \mathbf{x}_j^{(t+1)}) + K_t(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t+1)})] \quad (19)$$

We should note that each term in the kernel summation can be of different types such as RBF, polynomial, linear, and so on.

7.3.5 Kernels for Change Detection

In this section, we present two novel kernel-based formulations to deal with the particular problem of change detection. The methods are inspired in the usual difference and ratioing operations, but instead of in the input space, they are calculated in the feature space. The two main advantages of defining these operations in a high dimensional feature space are that: (i) it allows us to deal with the likely non-linear nature of the relationships among samples; and (ii) the free parameters are learned from the data.

7.3.5.1 Image difference in feature spaces

In remote sensing, change identification traditionally involves subtracting the subsequent images and then applying a threshold that is tuned either heuristically or under other more sophisticated criteria (Moser, Serpico, 2006). This difference vector can be formulated in the *kernel feature space*, as shown in Table 1, leading to the composite kernel:

$$\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_j) = & K_{t_o}(\mathbf{x}_i^{(t_o)}, \mathbf{x}_j^{(t_o)}) + K_{t_o-1}(\mathbf{x}_i^{(t_o-1)}, \mathbf{x}_j^{(t_o-1)}) \\
& - K_{t_o, t_o-1}(\mathbf{x}_i^{(t_o)}, \mathbf{x}_j^{(t_o-1)}) - K_{t_o-1, t_o}(\mathbf{x}_i^{(t_o-1)}, \mathbf{x}_j^{(t_o)})
\end{aligned} \tag{20}$$

7.3.5.2 Image ratioing in feature spaces

Another classical change detection method is the ratioing between images at two different dates, which helps to accommodate changes due to factors such the angle of the sun and shadows, and is widely used for SAR data processing. By defining the ratio kernel mapping as in Table 1, we obtain the ratioing operation with kernels:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \gamma \delta_{ij} + \frac{K_{t_o-1}(\mathbf{x}_i^{(t_o-1)}, \mathbf{x}_j^{(t_o-1)})}{K_{t_o}(\mathbf{x}_i^{(t_o)}, \mathbf{x}_j^{(t_o)})}, \tag{21}$$

where a regularization parameter γ is introduced to make the matrix positive definite. Otherwise, there would not be any warranty for this kernel to be a valid Mercer's kernel.

7.3.6 Kernels for Multi-source Classification

Similarly, we can integrate multi-sensor information. Now, if we have optical and radar information associated to the same co-registered pixel at time t_o , we can define the *optical feature* vector $\mathbf{o}_i^{(t_o)}$, the *radar feature* vector $\mathbf{r}_i^{(t_o)}$, and its concatenation $\mathbf{x}_i^{(t_o)} \equiv \{\mathbf{o}_i^{(t_o)}, \mathbf{r}_i^{(t_o)}\}$. The fact that we are using these vectors forces us to develop the stacked kernel given by $K(\mathbf{x}_i^{(t_o)}, \mathbf{x}_j^{(t_o)})$. Other kernels can be computed instead: the direct summation kernel, given by $K(\mathbf{x}_i^{(t_o)}, \mathbf{x}_j^{(t_o)}) = K_o(\mathbf{o}_i^{(t_o)}, \mathbf{o}_j^{(t_o)}) + K_r(\mathbf{r}_i^{(t_o)}, \mathbf{r}_j^{(t_o)})$; and the cross-information kernel, given by $K(\mathbf{x}_i^{(t_o)}, \mathbf{x}_j^{(t_o)}) = K_o(\mathbf{o}_i^{(t_o)}, \mathbf{o}_j^{(t_o)}) + K_r(\mathbf{r}_i^{(t_o)}, \mathbf{r}_j^{(t_o)}) + K_{or}(\mathbf{o}_i^{(t_o)}, \mathbf{r}_j^{(t_o)}) + K_{ro}(\mathbf{r}_i^{(t_o)}, \mathbf{o}_j^{(t_o)})$. Here again note that $\mathbf{o}_i^{(t_o)}$ and $\mathbf{r}_j^{(t_o)}$ need to have the same dimension ($N_r = N_o$) for this formulation to be valid, which is not a common situation.

7.4 EXPERIMENTAL RESULTS

In this section, the potential of the methods presented, for the solution of some complex urban monitoring classification problems, is illustrated.

7.4.1 Data collection and feature extraction

The images used in this experiment were collected in the Urban Expansion Monitoring (UrbEx) ESA-ESRIN DUP project (Castracane, Iavarone, et al., 2003). Data from UrbEx project were used to perform the analysis of the selected test site and for validation purposes as well. For further details, visit: <http://dup.esrin.esa.int/ionia/projects/summary30.asp>. The test sites considered were Rome and Naples (Italy), where images from ERS2 SAR and Landsat TM sensors were acquired in 1995 and 1999 (Figure 3). In this set of experiments, only two time instants are available and thus the complexity of the temporal classifiers reduces significantly, as $t_o = 2$.

An external digital elevation model (DEM) and a reference land cover map provided by the Italian Institute of Statistics (ISTAT) were also available. The

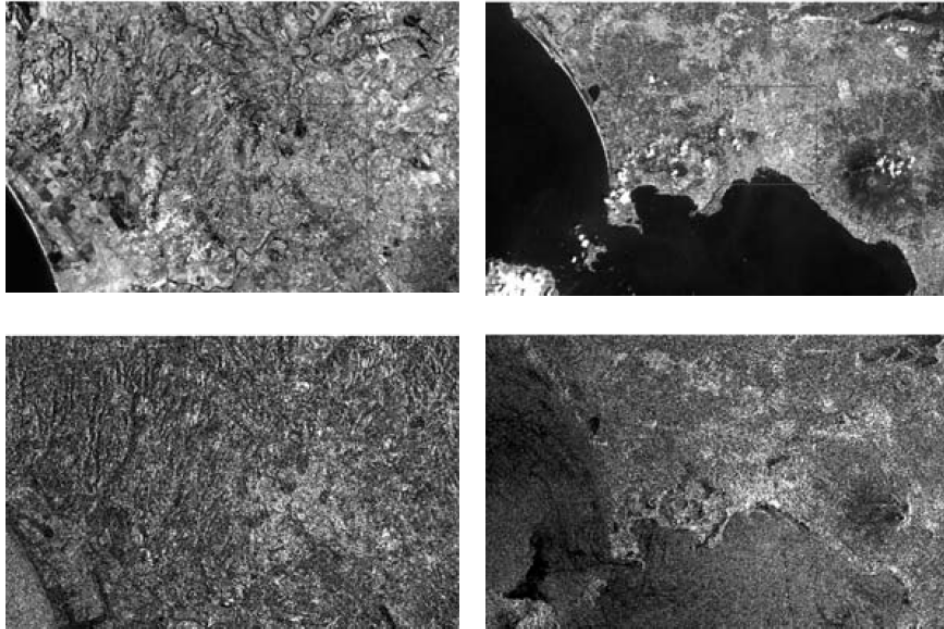


Figure 3: Images of the test areas of Rome (left) and Naples (right) acquired at 1999 both in RGB composite from L3, L2 and L1 bands (top row) and SAR log-intensities (bottom row).

ERS2 SAR 35-day interferometric pairs were selected with perpendicular baselines between 20-150 m in order to obtain the interferometric coherence from each complex SAR image pair. The available features were initially labeled as: L1-L7 for Landsat bands; In1-In2 for the SAR backscattering intensities (0-35 days); and Co for the coherence.

Since these features come from different sensors, the first step was to perform specific processing and conditioning of the optical and SAR data, and to co-register all images. The seven bands of Landsat TM were co-registered with the ISTAT classification data, and resampled to 30x30 m with the Nearest-Neighbor algorithm. The registration for the multi-source images was performed at sub-pixel level obtaining a root-mean-squared error of about 10m, which potentially enables good urban classification ability (Gomez-Chova, Fernández-Prieto, et al., 2006).

In the case of the *optical* images, the seven Landsat TM spectral bands (containing three *VIS*, one *Near IR*, two *Short-Wave IR*, and one *Thermal IR* bands) were used directly, $\mathbf{o}_i = \{L1, \dots, L7\}$. In the case of the SAR images, intensity and coherence were computed. However, since *speckle* disturbs image interpretation, a multi-stage spatial filtering approach over coherence images was followed to increase the urban areas discrimination (Gomez-Chova, Fernández-Prieto, et al., 2006), which yielded the fourth radar input feature, Co' . Therefore, we define in this case $\mathbf{r}_i = \{\text{In1}, \text{In2}, \text{Co}, \text{Co}'\}_i$.

Once features were extracted from optical and SAR images, we analyzed their potential use for urban change detection through scatter plots and density estimation. The results indicated that there was a high degree of overlap for areas of change and no-change pixels which indicated that this is an extremely difficult change detection problem and suggested that non-linear methods should be deployed. Therefore, we also computed spatial and textural features from these optical and SAR features. Specifically, the spatial features for the optical images were the average of all pixels in the surrounding 7x7 window, and the textural features for the SAR data were Gabor-filtered versions of $\mathbf{r}_i$ at different scales ($\theta = 1, \dots, 4$) and orientations ($\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$), thus yielding the textural radar features.

In the following experiments, we selected subset images from the Rome and Naples scenes containing 200×200 pixels in areas with substantial urban changes. For the case of the Rome scene, 1392 pixels changed to 'non-urban',

780 pixels changed to ‘urban’, and 2978 changed to the ‘unknown’ status in this 4-year period. For the case of the Naples scene, 1826 pixels changed to ‘urban’, 215 pixels changed to ‘non-urban’, and 1973 changed to the ‘unknown’ status. Pixels belonging to the unknown class were not considered, and hence, this becomes a classical binary problem of ‘change’ vs. ‘no-change’ identification.

7.4.2 One-class SVM

In this experiment, we randomly selected 25% of the changed pixels for training and used 5-fold cross-validation for free parameter tuning. Then, we tested the resulting classifier on the whole subset image. The RBF kernel widths were tuned in the range $\sigma = \{10^{-3}, \dots, 10^3\}$, and the rejection fraction parameter for the one-class SVM method was tuned in the range $\{10^{-3}, \dots, 10^0\}$. We compared the proposed One-class SVM method with other known one-class methods: Gaussian domain description (GDD), mixture of Gaussians (MoG) classifier, and the k -nearest neighbor (k -NN) classifier (Muñoz-Marí, Bruzzone, et al., 2007).

Figure 4 shows the true classification map for the Rome dataset 1999 and test results obtained by the classifiers averaged for 10 realizations. The classification maps for all of them are also included. Several conclusions can be obtained. First, SVDD offers very good results since rather than building a separating hyperplane ‘urban’/‘non-urban’, the method tries to model the ‘urban’ class accurately. Second, the standard Gaussian classifier fails due to the fact that the data do not fit the assumed distribution. The mixture of Gaussians classifier improves the results obtained by the Gaussian classifier, but its accuracy is slightly worse than that of SVDD. Furthermore, the KnnDD classifier produces the worst results, suggesting that data lie in much more complex underlying subspace than that detected locally through neighbors. Visually, results produced by the SVDD method are much better than the rest of the classifiers, yielding more homogeneous areas and a lower number of false detections, as analyzed in detail in (Muñoz-Marí, Bruzzone et al., 2007).

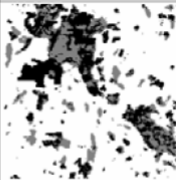
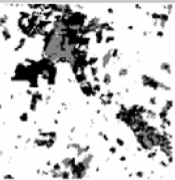
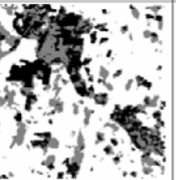
True map	GDD	MoG	k -NN	SVDD
				
	κ (OA %)	κ (OA %)	κ (OA %)	κ (OA %)
	0.58 (66.2)	0.61 (69.7)	0.48 (59.1)	0.63 (77.3)

Figure 4: From left to right: True thematic map for the Rome dataset in 1999, and classification maps by different one-class methods. *White* pixels represent the class ‘non-urban’, *black* pixels are ‘unknown class’, and *gray* pixels are ‘urban’. We indicate the kappa coefficient and overall accuracy averaged over 10 realizations.

7.4.3 Laplacian SVM

We used both the linear kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, and RBF kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2)$, where $\sigma \in \mathbb{R}^+$ is the kernel width for the SVM, LapSVM, and transductive SVM (TSVM) (Vapnik, 1998). The graph Laplacian, $\mathbf{L}$, consisted of $l+u$ nodes connected using k -nearest neighbors, and compute the edge weights W_{ij} using the Euclidean distance among samples.

For the experiments, we generated training and validation sets consisting of $l=400$ labeled samples (200 samples *per* class), and added $u=200$ unlabeled (randomly selected) samples from the analyzed images to the training set for the LapSVM and TSVM. We focused on the ill-posed scenario and vary the rate of both labeled and unlabeled samples independently, i.e. $\{2, 5, 10, 20, 50, 100\}\%$ of the labeled/unlabeled samples of the training set were used to train the models in each experiment. In order to avoid skewed conclusions, we ran all experiments for a number of realizations where the training samples used were randomly selected, and then averaged results are shown. All classifiers are compared using the overall accuracy OA[%] and the estimated kappa statistic κ as a measure of robustness in the classification.

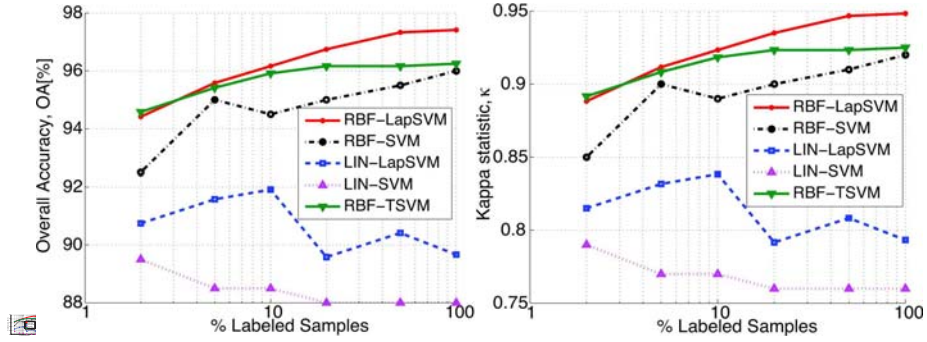


Figure 5: Overall Accuracy OA[%] (left) and Kappa statistic κ (right) over the validation set as a function of the rate of labeled training samples used to build models.

Free parameters γ_L and γ_M were varied in steps of one decade in the range $[10^{-4}, 10^4]$, the number of neighbors used to compute the Laplacian graph was varied from 3 to 9, and the Gaussian width was tuned in the range $\sigma = \{10^{-3}, \dots, 10\}$ for the RBF kernel. The selection of the best subset of free parameters was done by cross-validation.

Figures 5 and 6 show the validation results for the analyzed SVM-based classifiers. Several conclusions can be drawn from this figure. First, LapSVM classifiers produce better classification results than SVM in all cases (note that SVM is a particular case of the LapSVM for $\gamma_M = 0$) for both the linear and the RBF kernels. LapSVM also produces better classification results than TSVM when the number of labeled samples is increased. Differences among methods are numerically very similar when a low number of labeled samples is available.

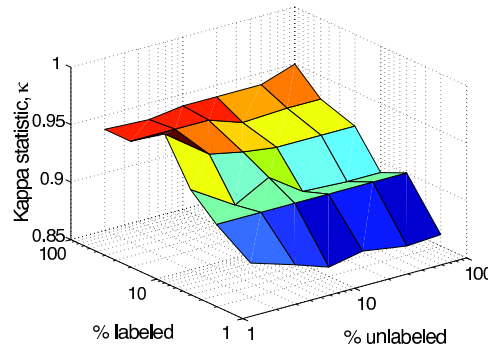


Figure 6. Kappa statistic validation surface for the best RBF-LapSVM classifier as a function of the rate of both labeled and unlabeled training samples.

In the case of the plot on the right, the κ surface for the LapSVM highlights the importance of the labeled information in this problem (Gómez-Chova, Camps-Valls et al., 2008).

7.4.4 Kernel-based Data Fusion and Multi-temporal Classification

This battery of experiments extends the previous ones, which used all input features in a stacked vector approach, by exploiting the proposed composite kernels framework to build classifiers that, in addition to the temporal information, also integrate contextual, textural and multi-source data. The experiments consider multi-temporal classification and change detection in real images. In all cases, we randomly selected 25% of the changed pixels for training and used 5-fold cross-validation for free parameter tuning. Then, we tested the resulting classifier on the whole image.

Table 2 shows the results obtained by different supervised classifiers. Specifically, we compare SVC and SVDD under multi-temporal classification (left) and change detection (right) using different temporal, spatio-spectral, and multi-source composite kernels for the scenes of Rome and Naples. We also analyze class-by-class accuracies for especially interesting cases, and assess statistical differences among classifiers through Wilcoxon's rank sum test at 95% confidence interval.

In all cases and scenes, it becomes obvious that the use of the RBF kernel provides much better results than the linear kernel which, in turn, constitutes an upper bound of the model's performance for (change detection) thresholding methods. For the Rome scene (Table 2[top]), and in the case with unlabeled information for the prediction date image (1999), i.e. $t < t_o$, only labeled samples in 1995 can be used to train a classifier and predict in 1999. In this complex situation, a purely supervised approach like SVC yields poor solutions ($OA[\%]<70\%$ and $k<0.6$) since there is no information about the change. On the other hand, SVDD produces good results since rather than building a separating hyperplane 'urban'/'non-urban', the method tries to model the 'urban' class accurately. In all cases, the best composite kernel for integrating the spatial and the different data sources was constituted by the *summation kernel*, i.e. dedicating separated kernels for the Landsat bands, SAR features, contextual Landsat features, and textural Gabor-filtered SAR features. This best method yielded a maximum accuracy of 84.2%, but with biased classifications ($k<0.51$),

which was confirmed by looking at the individual class accuracies (90.3% for urban and 53.4% for non-urban, respectively). Finally, it is also worth noting that solutions are much sparser for the SVDD (average of 22% of SVs) than for the SVC (average of 59% of SVs). For the Naples scene, similar results are obtained (see Table 1[bottom]). Again, when no information is available at time t_o , the SVDD constitutes a better approximation either with RBF or linear kernel

Table 2: Results for the Rome (top) and Naples (bottom) scenes. Overall accuracy (OA[%]) and kappa statistic (κ) for the different scenarios ($t < t_o$ and $t \leq t_o$), time integration, multi-source fusion, and classifiers algorithms. Average results over 10 realizations are shown for SVC and SVDD (using both a linear and RBF kernel). Best results (bold) and second best (italics) are highlighted for each column and kernel type. Statistically different results, either in terms of OA[%] (tested through paired Wilcoxon's rank sum test at 95% confidence interval) or κ (considering it is normally distributed) from the best classifier are marked with ‘*’.

			Multi-temp $t < t_o$	Multi-temp. $t \leq t_o$	Change detection, $t \leq t_o$			
			Spatio-Multi- Spectrasource	Summat. Eq. (10)	Summat. Eq. (10)	Cross-terms Eq. (11)	Weighted Eq. (13)	Kernel Diff. Eq. (14) Kernel Ratio Eq. (15)
Rome scene								
SVM	Sum	Stack	55.0 (0.20)	83.2 (0.45)	68.2 (0.61)	70.4 (0.64)	81.1 (0.70)	80.2 (0.70)
LIN	Cross	Stack	54.1 (0.22)	81.4 (0.49)	69.2 (0.62)	71.4 (0.63)	82.2 (0.68)	81.3 (0.71)
	Sum	Sum	57.1 (0.31)	84.1 (0.51)	70.2 (0.63)	73.4 (0.72)	74.1 (0.72)*	79.4 (0.71)
SVDD	Sum	Stack	58.1 (0.33)	71 (0.52)*	73.4 (0.63)	68.1 (0.62)*	82.1 (0.71)	80.2 (0.71)
D	Cross	Stack	58.3 (0.34)	77.6 (0.55)	74.1 (0.62)	69.1 (0.54)*	81.3 (0.73)	78.3 (0.72)
LIN	Sum	Sum	66.1 (0.40)	78.2 (0.55)	69.1 (0.62)	78.3 (0.68)	82.2 (0.61)	83.1 (0.72)
SVM	Sum	Stack	61.1 (0.51)*	91.4 (0.67)	83.1 (0.70)	89.5 (0.78)	95.3 (0.81)	95.1 (0.77)
RBF	Cross	Stack	66.5 (0.43)*	92.1 (0.69)	89.2 (0.71)	88.8 (0.77)	96.1 (0.79)	95.8 (0.80)
	Sum	Sum	68.3 (0.60)	93.2 (0.77)	94.3 (0.78)	93.3 (0.81)	94.1 (0.83)	93.3 (0.80)
SVDD	Sum	Stack	77.3 (0.63)	83 (0.66)*	85.3 (0.68)	88.1 (0.71)	92.3 (0.80)	91.1 (0.79)
D	Cross	Stack	75.1 (0.64)	88.7 (0.68)	84.5 (0.70)	79.0 (0.55)*	91.9 (0.82)	90.2 (0.79)
RBF	Sum	Sum	81.4 (0.70)	92.1 (0.75)	92.2 (0.75)	91.1 (0.79)	95.6 (0.80)	96.0 (0.81)
Naples scene								
SVM	Sum	Stack	58.3 (0.39)	79.8 (0.48)	80.8 (0.51)	81.0 (0.55)	85.1 (0.60)	81.3 (0.61)
LIN	Cross	Stack	56.2 (0.41)	77.1 (0.45)	80.3 (0.50)	82.9 (0.55)	87.4 (0.64)	86.3 (0.62)
	Sum	Sum	59.1 (0.41)	77.8 (0.45)	81.3 (0.52)	86.1 (0.55)	87.4 (0.64)	86.1 (0.62)
SVDD	Sum	Stack	60.1 (0.44)	77.8 (0.45)	71.8 (0.43)	71.1 (0.49)*	85.9 (0.60)	85.9 (0.61)
D	Cross	Stack	61.4 (0.52)	72.6 (0.45)	70.6 (0.41)*	72.8 (0.44)*	87.8 (0.64)	85.1 (0.62)
LIN	Sum	Sum	64.3 (0.54)	75.8 (0.55)	75.8 (0.42)	76.0 (0.51)	84.5 (0.62)	87.3 (0.62)
SVM	Sum	Stack	70.1 (0.40)*	90.3 (0.71)	92.0 (0.59)	93.3 (0.60)	95.1 (0.71)	95.8 (0.75)
RBF	Cross	Stack	68.9 (0.45)*	94.1 (0.60)	88.1 (0.59)	94.4 (0.60)	98.0 (0.75)	96.3 (0.76)
	Sum	Sum	66.4 (0.55)*	95.0 (0.73)	96.8 (0.64)	97.5 (0.66)	96.9 (0.73)	96.9 (0.76)
SVDD	Sum	Stack	80.2 (0.50)	89.1 (0.57)	91.2 (0.56)	91.1 (0.59)	95.9 (0.70)	95.8 (0.71)
D	Cross	Stack	82.6 (0.50)	92.7 (0.55)	90.4 (0.60)	92.2 (0.57)	97.2 (0.74)	95.3 (0.71)
RBF	Sum	Sum	84.2 (0.51)	95.3 (0.67)	95.8 (0.61)	96.0 (0.65)	97.6 (0.71)	97.6 (0.71)

embedding. These results are not only numerically different, but also differences are statistically significant (see star symbols in Table 1, $p < 0.005$) (Camps-Valls, Gómez-Chova et al., 2008).

In the case with available labeled information for t_o , several composite kernels have been tested, improving significantly the results in both scenes. This is a clear consequence of using labeled samples from the t_o image. In these cases, SVC classifiers show the best results, but it can be appreciated that SVDD classifiers also produce stable and robust outcomes, which confirms its suitability to application scenarios in which incomplete or partially complete information is available. Similar results have been lately observed in Muñoz-Marí, Bruzzone, et al., (2007). The best overall result was obtained using simple summation kernels for integrating the spatio-spectral information and, in some cases, more complex cross-information kernels to process the temporal information. This type of classifier yielded a maximum OA=94.3%, statistically balanced model ($k=0.78$), and good individual classification accuracies (97.1% for urban and 82.5% for non-urban, respectively) for the Rome image, and a maximum OA=96.8% ($k=0.64$), and individual classification accuracies of 98.3% for urban and 83.3% for non-urban in the Naples image (Camps-Valls, Gómez-Chova et al., 2008).

7.4.5 Kernel-based Change Detection

Table 1 (right part) shows the results for the *difference and ratio kernels* for change detection. In these cases, labeled information for t_o is provided in the form of ‘change’ vs. ‘no-change’ for the 1999 image, so it can be considered as a supervised learning strategy. In general, a significant (both numerical and statistical) difference is observed by using RBF-based kernel classifiers (e.g. accuracy is about +12% higher). Note that, in this case, even a linear regularized method yields significantly better overall accuracy (+8%) (Camps-Valls, Gómez-Chova et al., 2008). The SVMs yield very good results in terms of accuracy (OA>90%, $k>0.7$), but SVDD provides better kappa values (although no significant statistical differences are appreciated), which indicates well-balanced classifications with reduced false detections. For the best SVC (SVDD) classifier, the individual accuracies were 97% (98%) for the ‘change’ class and 69% (74%) for the ‘no change’ class in the Rome image dataset. For the Naples dataset, results between SVC and SVDD did not differ significantly (98% vs. 97% for the change class, and 75% vs. 74% for the ‘no change’ class), probably due to the fact that this constitutes an easier problem and no particular assistance

to learn a specific class is included in the methods. Note, however, that this is a different (and much easier) experimental setup than the multi-temporal approach, as the classifier only has to detect if pixel's are labeled changed or not, rather than the class label. The classification maps offered by the best methods (bold faced in Table 1) are represented in Figure 7 for Rome (top) and Naples (bottom). The numerical results (already discussed before) are in general confirmed by visual inspection. For instance, in the case of $t < t_o$, results offered by the SVDD method are much better than the SVC (more homogenous areas and lower number of false detections), which is also observed in the case of using the difference or ratio kernels for change detection, even though (slightly) better accuracies are obtained using the SVC. In the case of $t \leq t_o$, SVDD does not outperform SVC, mainly because it poorly integrates the spatial/textural information (e.g. see southern parts of the Rome scene or Naples middle east, where evident changes occur).

7.5 CONCLUSIONS

In this chapter, we summarized our latest developments based on kernel methods for efficient hyperspectral image classification. In particular, we analyze supervised and semi-supervised support vector machines (SVMs), one-class SVM approaches, kernel-based fusion of heterogeneous information, multi-temporal classification kernel schemes, and the specific problem of change detection with kernels. We analyzed the methods theoretically, and showed their effective performance in several illustrative (yet challenging) real scenarios.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry for Education and Science under projects DATASAT/ESP2005-07724-C05-03 and CONSOLIDER/ CSD2007-00018.

References

- Aizerman, M. A., Braverman, E. M., Rozoner, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote Control* 25, 821–837.
- Belkin, M., Niyogi, P., Sindhvani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research* 7, 2399–2434.
- Camps-Valls, G., Bruzzone, L. (2005). Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* 43 (6), 1351–1362.
- Camps-Valls, G., Gómez-Chova, L., Calpe, J., Soria, E., Martín, J. D., Alonso, L., Moreno, J., (2004). Robust support vector method for hyperspectral data classification and knowledge discovery. *IEEE Transactions on Geoscience and Remote Sensing* 42 (7), 1530–1542.
- Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Rojo-Álvarez, J. L., Martínez-Ramón, M., (2008). Kernel-based framework for multi-temporal and multisource remote sensing data classification and change detection. *IEEE Transactions on Geoscience and Remote Sensing* Accepted for publication.

- Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., Calpe-Maravilla, J., (2006a). Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters* 3 (1), 93–97, iSSN:1545-598X.
- Camps-Valls, G., Rojo-Álvarez, J. L., Martínez-Ramón, M. (Eds.), (2006b). *Kernel Methods in Bioengineering, Signal and Image Processing*. Idea Group Publishing, Hershey, PA (USA), iISBN: 1-559904-042-5.
- Castracane, P., Iavarone, F., Mica, S., Sottile, E., Vignola, C., Arino, O., Cataldo, M., Fernández-Prieto, D., Guidotti, G., Masullo, A., Pratesi, I., (2003). Monitoring urban sprawl and its trends with EO data. UrbEx, a prototype national service from a WWF-ESA joint effort. In: *2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*. pp. 245–248.
- Chapelle, O., Schölkopf, B., Zien, A. (2006). *Semi-Supervised Learning*, 1st Edition. MIT Press, Cambridge, MA and London, England.
- Fukunaga, K., Hayes, R. R. (1989). Effects of sample size in classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (8), 873–885.
- Gómez-Chova, L., Camps-Valls, G., Muñoz-Marí, J., Calpe-Maravilla, J., (2008). Semi-supervised image classification with Laplacian support vector machines. *IEEE Geoscience and Remote Sensing Letters*.
- Gómez-Chova, L., Fernández-Prieto, D., Calpe, J., Soria, E., Vila-Francés, J., Camps-Valls, G., (2006). Urban monitoring using multitemporal SAR and multispectral data. *Pattern Recognition Letters*, Special Issue on “Pattern Recognition in Remote Sensing” 27 (4), 234–243.
- Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory* 14 (1), 55–63.
- Mercer, J., (1905). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A* CCIX (A456), 215–228.
- Moser, G., Serpico, S. (2006). Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery. *IEEE Transactions on Geoscience and Remote Sensing* 44 (10(2)), 2972–2982.
- Muñoz-Marí, J., Bruzzone, L., Camps-Valls, G., (2007). A support vector domain description approach to supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 45 (8), 2683–2692.
- Richards, J. A., Jia, X. (1999). *Remote Sensing Digital Image Analysis. An Introduction*, 3rd Edition. Springer-Verlag, Berlin, Heidelberg.
- Schölkopf, B., Smola, A. (2002). *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press Series.
- Schölkopf, B., Williamson, R. C., Smola, A., Shawe-Taylor, J., (1999). Support vector method for novelty detection. In: *Advances in Neural Information Processing Systems 12*. Denver, CO.
- Tax, D., Duin, R. P. (1999). Support vector domain description. *Pattern Recognition Letters* 20, 1191–1199.
- Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons, New York.

Chapter 8

Dynamic Error Recovery in the ATLAS TDAQ System: Using Multi-Layer Perceptron Neural Networks

John Erik Sloper, Giovanna Lehmann Miotto and Evor L Hines

In this chapter we will show how intelligent system (IS) approaches can be successfully applied to an existing distributed software system of significant size; comprising 3000 nodes running more than 20000 processes. We will here use several IS engineering techniques in the context of the ATLAS (A Toroidal LHC ApparatuS) TDAQ (Trigger and DataAcQuisition) system currently being developed and used at CERN, Geneva.

8.1 INTRODUCTION

The ATLAS (Atlas Collaboration, 1994) is at the very forefront of particle physics research and incorporates a large number of custom hardware and software modules. It consists of a large cylinder (43 m length x 25 m diameter) of detecting devices, placed around one of the beam collision points of the Large Hadron Collider (Evans, 2007). Its scope is to determine which particles are produced during collisions of high energy protons every 25 ns. This section will give a brief overview of the ATLAS TDAQ and the current error handling used within the system.

8.1.1 *The ATLAS TDAQ system*

The ATLAS TDAQ system is a large heterogeneous system consisting of a wide variety of software and hardware components. The final working system will

consist of approximately 3000 nodes running more than 20000 processes. The system is divided into a number of sub-systems, each performing a well-defined task such as readout from the detector, filtering of data, etc. Figure 1 shows an overview of the different parts of the system. A detailed description of the TDAQ system can be found in (ATLAS High Level Data Acquisition and Controls - Technical Design Report, 2003).

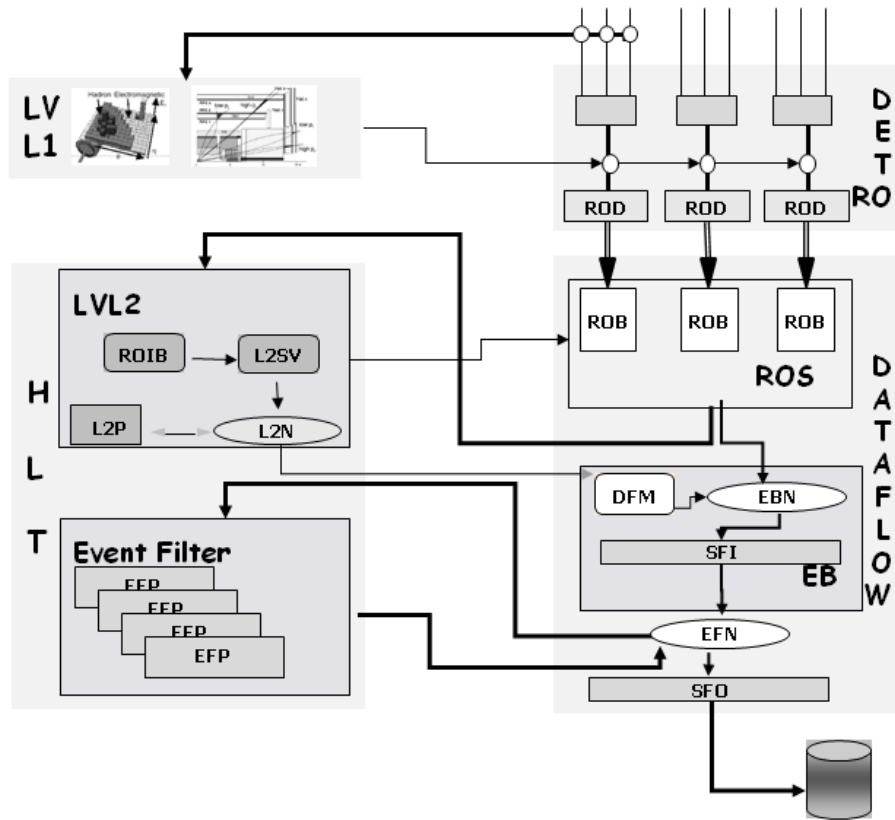


Figure 1: Schematic overview of the TDAQ system.

8.1.2 Error handling in ATLAS TDAQ

The TDAQ system has been under development for more than a decade. There has not, until recently, been much effort concerned with the development of an advanced Error Recovery (ER) system. Previous error recovery systems

have been very simple and thus were only able to handle relatively trivial cases such as restarting or ignoring dead processes, or putting the system into an error state to be recovered through human intervention. As the start-up of the experiment (summer 2008) is coming closer, larger and larger system configurations are being used. Data must be gathered at full capacity without operational dead time while the accelerator is delivering particles (over periods of approximately 10 hours). During the testing phase of the detector itself the TDAQ system should be able to collect data at low rates continuously over several days. Due to the large timeframes and the number of components involved in the data gathering, errors of various types and severities will almost certainly occur. It is therefore critical that the system is able to detect error situations as soon as possible as it arises and that the situation be recovered with minimal adverse effect on the data acquisition. Both the detection and recovery should be realized with the need for as little user interaction as possible.

```
(defrule ignore-if-dies

  "Ignore an application that died"
  (declare (salience 1000))

  (object (is-a RUNCONTROLLER) (RC-FAULT FALSE) (STATUS UP) )
  ?app<-(object (is-a APPLICATION) (APP-NAME ?name) (HOST ?host)
            (STATUS ABSENT) (FAULT TRUE)
            (IF-DIESIgnore) (MEMBERSHIP IN)
            (ERROR-CODE EXITED | SIGNALLED))
=>
  (clipsServer-log 3 "INFO [ignore-if-dies] fired.")
  (ers-warning "ApplicationIssue" "Dead Application " ?name "
    will be ignored." )
  (send ?app exclude)
)
```

Figure 2: An example of a rule from the knowledgebase.

The current error handling system (Sloper, Lehmann Miotto, & Hines, April 29 2007-May 4 2007) is based on an expert system. In this approach expert knowledge is encoded as IF-THEN rules which react to different known error situations. Figure 2 shows an example rule from the knowledgebase.

The expert system was implemented using CLIPS (C Language Integrate Production System) (Riley, Culbert, & Lopez, 1991). While the expert system

approach allows for a lot of flexibility it has some drawbacks. For example, first of all the expert knowledge must be gathered and implemented in the system. This is usually done through discussion with the relevant sub-system experts. As the number of developers and system experts grow, this becomes increasingly difficult. The problem is particularly even more challenging in the case of the ATLAS TDAQ system where developers and other experts may also be located at different geographical locations around the world!

Another limitation is that the system is generally poor at generalizing, as the antecedents and consequents of the rules must be explicitly coded by the developer. For some of the error situations arising in the TDAQ system this is a real issue, and we are therefore looking at other options for detecting and/or handling such error situations.

8.2 AN OVERVIEW OF SOME OF THE MAIN TECHNIQUES WHICH WE WILL USE

In this section we will briefly illustrate the different techniques that we will apply to the ATLAS TDAQ system.

8.2.1 Fuzzy C-means (FCM) clustering

Clustering, in general, is a method for dividing a set of data into two or more groups and is accomplished by identifying similarities between the individual data points and categorizing groups where data points in that group exhibit some similarities.

FCM clustering was first proposed in 1973 by J.C. Dunn (Dunn, 1973) and was further developed by Bezdek (Bezdek J. C., 1981). It is a method where a data point is allowed to belong to two or more clusters according to a fuzzy membership function for each cluster.

A detailed explanation of FCM clustering can be found in chapter 17.

8.2.2 Principal Component Analysis

Principal component analysis (PCA) is a mathematical method that reduces the number of dimensions in a dataset while retaining as much of the variance as possible. The method has a wide range of uses including data mining,

dimensionality reduction and de-correlation, pattern recognition and (lossy) data compression (Liu, Ma, Qian, & Lv, 2006) (Tien, Lim, & Jun, 2-6 Nov. 2004). We will use it to simplify our dataset and to eliminate redundant information and filter out noise.

Principal component analysis is realized by projecting the original M dimensional data into a D dimensional space where $D \leq M$. This D dimensional space consists of orthogonal vectors called principal components. The principal components are chosen in such a way that a maximum of variance is achieved and they are ordered by the amount of variance explained by each of them. In this way the first principal component explains more variance than the second and so on. By choosing $D < M$ one can therefore project the data into a lower dimensional space while retaining as much variance as possible.

There exist a number of variants of PCA (Wei-min & Chein-I, 23-28 July 2007) though we will use the regular form in this chapter.

8.2.3 Self Organized Maps

Self organized map (SOM) is another technique we will use for clustering and data visualization. It has widespread use in data mining, classification and data visualization. The SOM was first introduced by Kohonen (Kohonen, 1982). SOM is based on unsupervised learning, meaning that it does not use any a priori knowledge about the data to be clustered. It is capable of finding classes/clusters inherent in the data (though depending on the dataset this will not always be true).

A detailed explanation of SOM can be found in chapter 17.

8.2.4 Multi Layer Perceptron (MLP)

MLP is a neural network where there is one or more hidden layers between the input and output nodes. The MLP nodes include nonlinearity in their output usually represented by a logistic function. Each node in a layer is connected to every node in the next layer. This connection is weighted and is modified when training the network. Training of the network is usually done using the error back propagation algorithm (Rumelhart, Hinton, & Williams, 1986). The input is propagated through the network and the output is compared to a known correct value (supervised learning). Any error is then propagated back towards the input

and the weights of the different nodes are updated. This process is repeated for a specified number of iterations or until the error is within a pre-established tolerance. Figure 3 shows an overview of an example MLP.

8.2.5 Time-Delay Neural Network (TDNN)

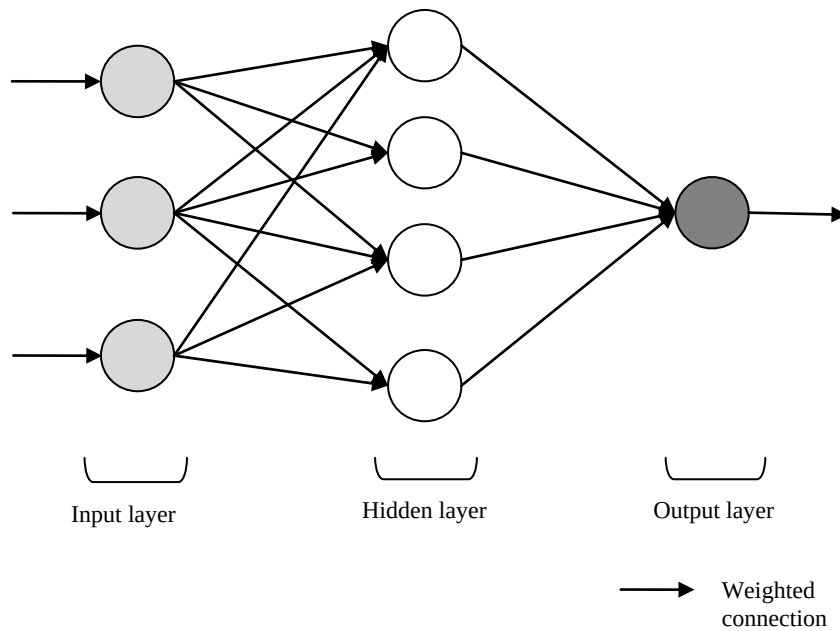


Figure 3: A simple MLP with 3 inputs, one hidden layer (with 4 units) and one output (1 unit).

TDNNs were first introduced by Waibel in 1987 for use in speech recognition (Waibel, Waibel, Hanazawa, Hinton, Shikano, & Lang, 1989). By utilizing a number of delayed inputs, this method allows the network to learn temporal patterns in the data.

Compared to a regular MLP each unit in the first hidden layer is modified to be presented with D_N delayed input vectors in addition to the un-delayed input. If we consider an input vector to be of size $I=5$ and a delay of $N=2$ the hidden units will have 15 weighted inputs. As each input vector is presented to the hidden units at $N+1$ different point in time, it allows the TDNN to correlate temporal changes in the input data. It is possible to introduce inputs at subsequent

hidden layers, but we will use the simple form of delays only at the input in this chapter.

TDNNs normally use the back propagation algorithm for learning, modified to take into account the delayed inputs.

8.3 CASE STUDY

In this chapter we will look at the application of the techniques described previously to the ATLAS TDAQ system, using data gathered from a real system using a test configuration. Errors will be deliberately induced at given points for evaluation purposes.

8.3.1 Data gathering

In this section we will focus on the data available in the TDAQ system and how it was gathered for analysis.

8.3.1.1 Overview

There are two main sources of data; error messages and performance measures of the different applications or sub-systems. Let us consider each in turn:

- (1) There are two main packages in the TDAQ system facilitating the gathering of error messages from the applications; Error Reporting Service (ERS) and the LogService (Murillo-Garcia & Lehmann Miotto, 2007). The ERS provides a common mechanism via which all applications in the system can report errors. It is possible to configure global settings that define the behaviour of the error reporting, such as where errors are sent, amount of information for each error, etc. This common framework also makes it possible to collect errors into classes/groups to be reused by other applications. For example in the case of a 'file not found' or a 'connection not possible to establish', different applications developed by different developers can use the same 'error class' (or sub-classes thereof) to report the issue. The TDAQ system also includes a 'log service' which gathers all errors reported through ERS by applications and stores them in a common database. It is then possible to browse this data and retrieve subsets of it based on

parameters such as level of severity, application type, time, etc. The ‘log service’ stores the ERS messages in a specific format, which naturally limits what information is available when analyzing the data. The attributes available for each ERS message are shown in Table 1.

- (2) Data about the performance of the different components are also available; including CPU rates, buffer levels, data throughput, the total amount of data processed by the different sub-systems, etc. These data are available while the system is running. We have implemented some tools to gather and store time-series for the different measures which can then be used for analysis and to develop error detection techniques. The tool gathers performance datasets from 5 different sub-systems and creates a time-series of the information in each. The performance rates are gathered once per second and stored in a file for later analysis.

Throughout this chapter we will be considering the error messages generated by the applications though many of the techniques can be applied to the performance rates as well.

Table 1: Attributes available in each ERS message.

Field	Description
Message id	This is a unique name assigned to a certain error type.
Machine name	The name of the machine the error originated from.
Application name	Identifier of the application sending the error (as set in the configuration database).
Severity	Severity of the error.
Message	A text description of the problem. Usually for human readability, but they are in some cases parsed by the system automatically.
Time	The date and time the error was created.
Parameters	A number of other parameters such as file name and line number where the issue was raised, the user running the process, etc.

8.3.1.2 Experimental setup

For the initial experiments a relatively simple configuration of the TDAQ system is used. This configuration includes all parts of the data gathering chain; starting with detector readout through filtering and finally storing data on disks. The total number of applications/processes in this particular configuration is approximately 30, though some of them are not directly involved in the ‘data chain’. A simple description of each component can be found in Table 2 and a

schematic view of the experimental system is shown in Figure 4. A full description of the TDAQ system in general can be found in (ATLAS High Level Data Acquisition and Controls - Technical Design Report, 2003).

There are several reasons for starting with a simple configuration. First of all it is easier to control the behaviour and environment surrounding the setup. This is important when developing the techniques as the methods we are using are based on supervised and/or unsupervised learning techniques. Secondly it allows for an easier assessment of the overall status of the system, ranging from 100% functional through partially functional to 100% non-functional. For practical reasons we will here explore situations where the system is either fully functionally or completely non-functional.

Table 2: A description of the components using the experimental setup.

Abbreviation	Name	Description
ROS	Read out System	Part of the system responsible for receiving data from the detector electronics.
L2SV	Level 2 supervisor	Responsible for distributing data within the level 2 system and communicating accept/reject decisions to the DFM.
L2PU	Level 2 processing unit	Assigned tasks by the L2SV and takes decision which are communicated back to the L2SV.
DFM	Dataflow manager	Gets information from the L2SV. Orchestrates the correct flow of data between ROSs and SFIs.
SFI	Sub-farm input	Part of the event building system where data from the entire detector is collected into single units called events.
EFD	Event filter dataflow	Responsible for dataflow within the event filter where further filtering of the data is made.
PT	Processing Task.	Part of the event filter. Responsible for making an accept/reject decision for a single event.
SFO	Sub-farm output	Receives the final events and forwards them to mass storage.

Let us consider a scenario in which we will try to detect error conditions where a ROS application has silently stopped working. This could happen due to a number of reasons such as an internal failure, or some operating system or network related problems. We will simulate this situation by using POSIX signals. POSIX is an IEEE standard ([ISO/IEC 9945](#)) for user and software application interface to the operating system and is supported by the UNIX systems we use for our experiments. Among other things, it defines an interface that allows us to send signals/commands to any running process. There are a

number of signals available, but we will use the SIGSTOP signal, which according to the standard should stop a computer program for later resumption. This allows us to simulate the situation where a process is no longer working properly (or rather not at all) without actually terminating the process. We can then resume its operation when we choose by sending it the SIGCONT signal.

This procedure is repeated four times and the data from each test have been combined. The final dataset consists of 1475 ERS messages.

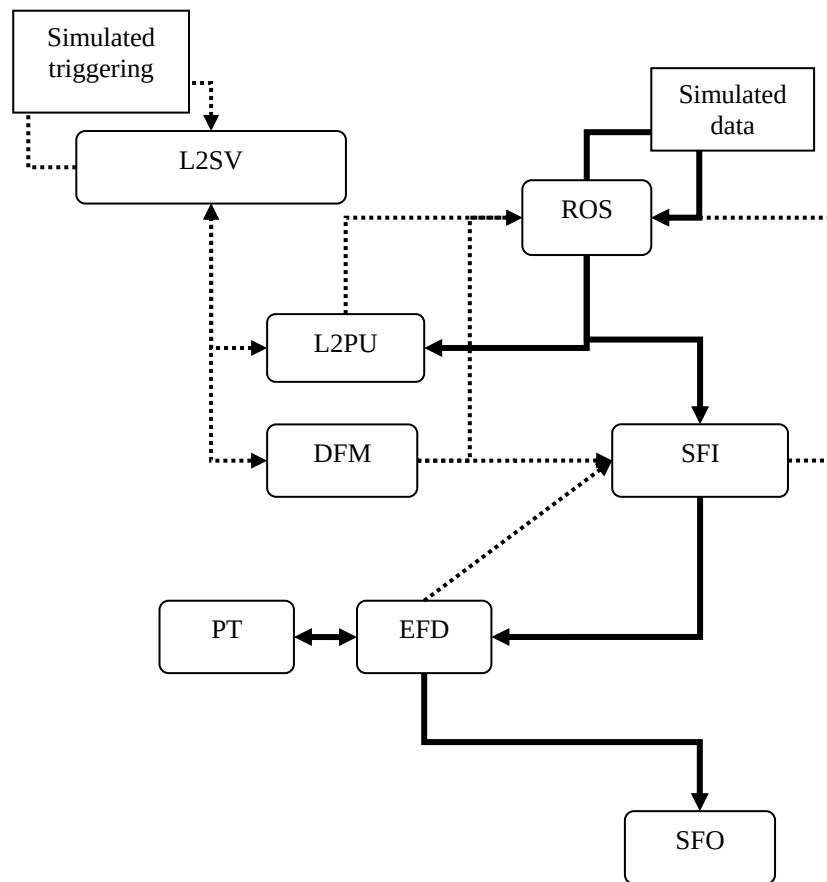


Figure 4: The flow of information and control messages (dashed arrows) and data (bold arrows) in the experimental setup.

8.3.2 Analyzing the datasets

In this section we will do some simple analysis of the datasets using different statistical techniques and look at how we can pre-process the data so that we will be able to apply FCM clustering, SOM and MLP to the data. However there are many considerations when gathering a dataset and we will first discuss some of the factors that will affect our analysis, in particular any uncertainties in and incompleteness of the datasets.

8.3.2.1 Uncertainty

As in the case of most datasets there are a number of factors leading to uncertainty, these include:

- (1) First of all, it is not obvious how to choose what features of the data is of real importance. In our example there are a number of features for each error message, such as the machine and the application it originated from, as well as time, severity, etc. To be able to analyse the data as efficiently as possible some of these fields might be discarded, modified or converted when the data is pre-processed. For example, fields containing text, such as the application name, might have to be converted into numerical values depending on the analysis method used.
- (2) Another problem is identifying what sub-set of the data is of relevance to a particular analysis. In such a large system there will be a lot of ‘noise’ which may not be relevant to the particular problem one is looking at. As an example, for a system of $O(10000)$ applications while one is trying to find the reasons or symptoms for a particular error, there will be a lot of messages from applications which are not related to the particular problem being investigated. Indeed even for smaller systems currently being used the number of messages gathered over a few days is in the order of millions.
- (3) Due to the distributed nature of the system another form of uncertainty arises. Applications will send error messages based on individual conditions such as lost connections, missing events, etc. There might be a number of factors determining the time when these conditions arises, such as current bandwidth, CPU scheduling, etc. Identical error

conditions can therefore produce different sets of error messages even if all configuration data available to be set by the user are identical.

8.3.2.2 Incompleteness

There are two main aspects of incompleteness to be considered:

- (1) Since we are analyzing messages sent by applications this clearly depends on the developers of that application. We are assuming that there exists a pattern of messages leading to a certain error (for example warning of full buffers, or delayed responses). However, there are no guarantees that an application raises the correct cause of the malfunction (or any at all) for a specific problem. It is up to each developer of the specific application to put in place appropriate mechanisms for handling errors in a consistent way and raise the correct ‘alarm’ in order to represent the error in the most appropriate way. As there are many tens of developers working on the ATLAS TDAQ system this suggest that mistakes are likely to be made during development. We will account for this when interpreting our results.
- (2) Due to the distributed nature of the system, the possible effects of having a large number of developers and a relatively frequent release cycle (about two releases each year) the reciprocal influence of the different components is not necessarily fully understood. In addition, since all the components have a highly specialized behaviour it is very difficult for a single person to have complete knowledge of all the parts of the system and how they interact. Thus the implications of some errors on the rest of the system are near-impossible to deduce without actually running the system and gathering empirical datasets which can then be analyzed.

8.3.2.3 Pre-processing the data

Before being analysed the data is transformed appropriately; that is data pre-processing. One common aspect to all the techniques we will apply is to transform any text data into numerical values. This is achieved using a custom hash function which maps each text string to a unique (collisions are theoretically possible, but are very unlikely) numerical value. The hash function puts more emphasis on the leading characters to make sure that similar text values are

mapped to similar numerical values. The data is then normalized so that all input lie in the range of [0,1].

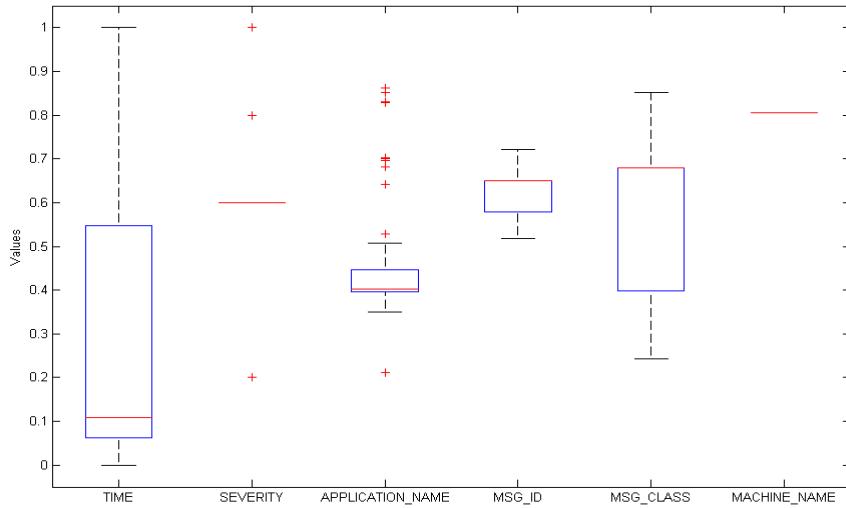


Figure 5: Shows statistics for the different attributes after preprocessing.

After applying pre-processing we can now easily present the variance of the different variables in our dataset as shown in Figure 5, though this of course depends on how the mapping from text to numerical values is done. In the figure the box lines represent the lower quartile, the median and the upper quartile. This allows us to make some immediate observations. One can for example observe that the majority of the messages are located early in the time span (upper quartile at 0.55 and mean at 0.1). Also note that machine name has no variability as the entire system was run on a single machine.

8.3.3 Results

In this section we will explore the results obtained using the techniques described in X.2 on the pre-processed data. For all the clustering the Davies-Bouldin (DB) index is used to choose the ‘correct’ number of clusters. The DB index (Davies & Bouldin, 1979) has been shown to compare well to other methods of choosing clusters (Bezdek & Pal, 1998).

8.3.3.1 FCM clustering

We perform the clustering using between 2 and 12 clusters and for each number of clusters we perform the clustering a number of times and choose the best result. In our tests we choose to do the clustering 5 times which was found to be a reasonable trade-off in terms of efficiency and accuracy. Using the DB index we find an optimal of 10 clusters in the data. The result of the DB index and sum of squared error (SSE) for each cluster size are shown in Figure 6. Figure 7 shows the dataset plotted by application name, message class and time, where each point is labelled by the corresponding class. Note that some overlapping is expected as the FCM clustering is performed using all dimensions of the data while the presentation is done in 3 dimensions. Also, bear in mind that due to the discrete nature of our data there might be a large number of observations corresponding to the same data point in our plot. Figure 8 shows the number of messages in each cluster giving us a better idea of how the data are divided among the clusters.

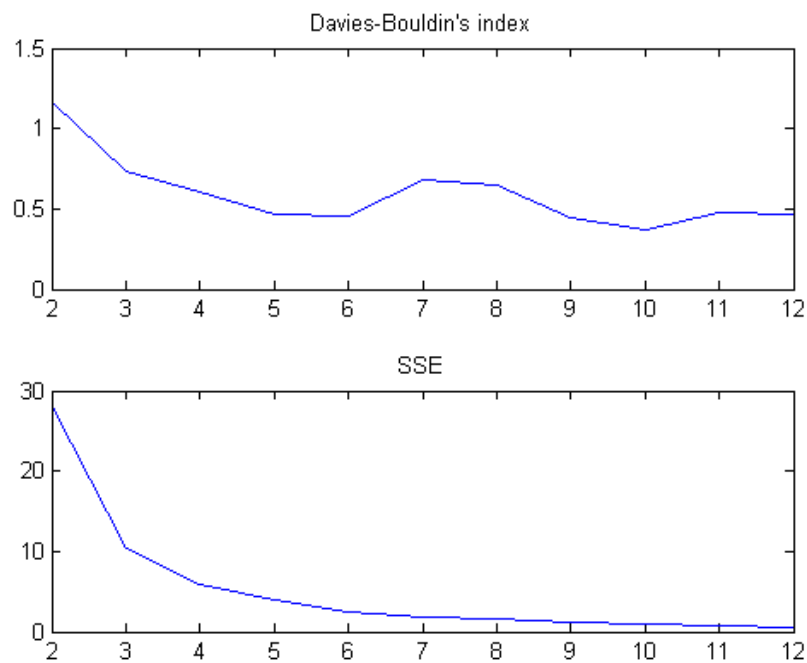


Figure 6: DB index and error for each cluster size.

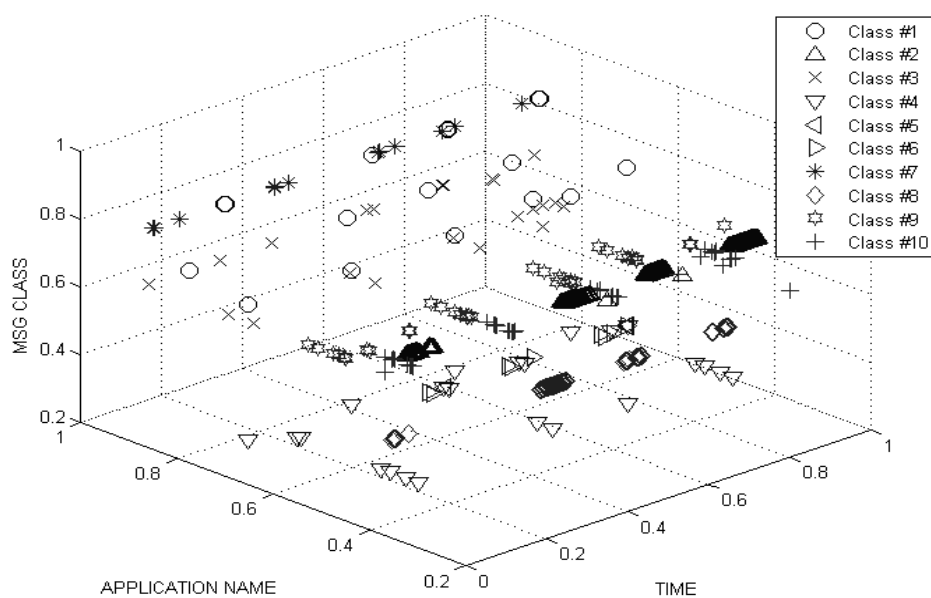


Figure 7: The dataset plotted using message name, class and time.

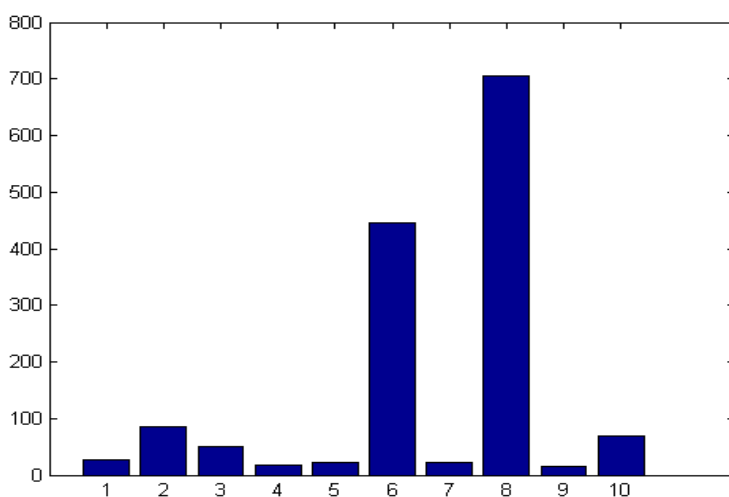


Figure 8: Number of messages in each cluster.

Examining the largest clusters we see that the time of the messages correspond well with the time when the errors of the ROS were simulated. Cluster 6 consists of messages solely from the SFI-1, while cluster 8 consists of messages from the L2PU and the L2SV. Most of the messages are from applications that are directly connected to the ROS application (see Figure 4) which is naturally to be expected. The messages from the L2SV, which have no direct link to the ROS, are probably due to the fact that the L2PU cannot operate properly because the ROS is halted, which then causes the L2SV to issue errors about the failing L2PU. These kinds of ‘cascade’ of errors are expected, but may be difficult to predict a priori and may also depend on various configuration parameters such as time-outs, buffer sizes, etc.

8.3.3.2 PCA

We will retain the 3 first principal components since PCA suggests that they explain 85% of the variance in the data. Figure 9 shows the dataset projected onto the two first principal components. As several observations correspond to the same point (due to the discrete nature of the dataset) we have chosen to use a marker with a size corresponding to the number of observations at a given point. This helps to properly visualize the data and gives a clearer indication of any

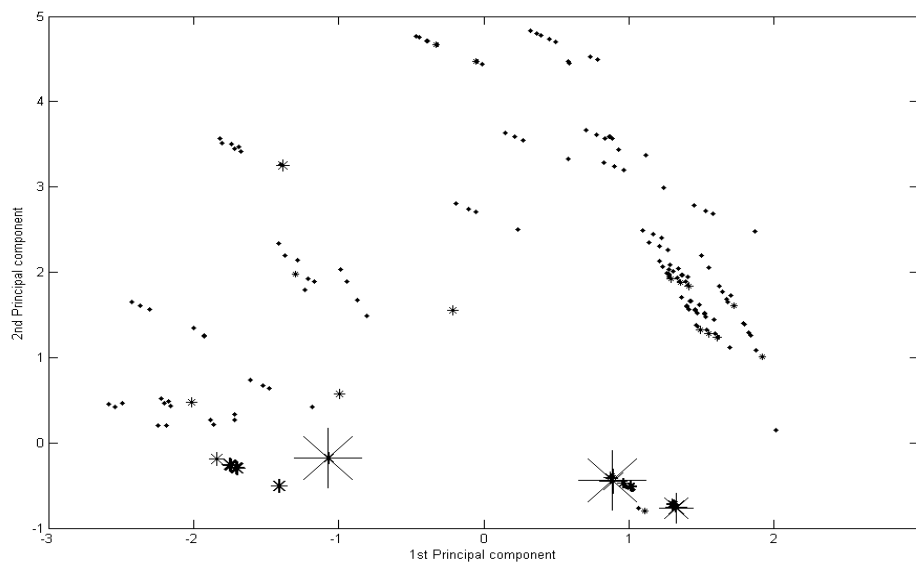


Figure 9: Shows the dataset projected onto the two main principal components. A larger marker indicates a greater number of observations.

clusters.

Using FCM to cluster the data obtained using PCA (retaining the three first components) we find 6 clusters as shown in Figure 10. The clustering is performed in the same way as described in the “FCM clustering” section. Looking more closely at the clusters we find that clusters 2, 3 and 4 corresponds to the circumstances when errors occurred in the system. Comparing this with the main clusters found using FCM clustering on the original data we see that they are almost identical in size (within 3%) and 96.7% of the messages in the FCM clustering of the PCA data are also in the FCM clusters using the original data.

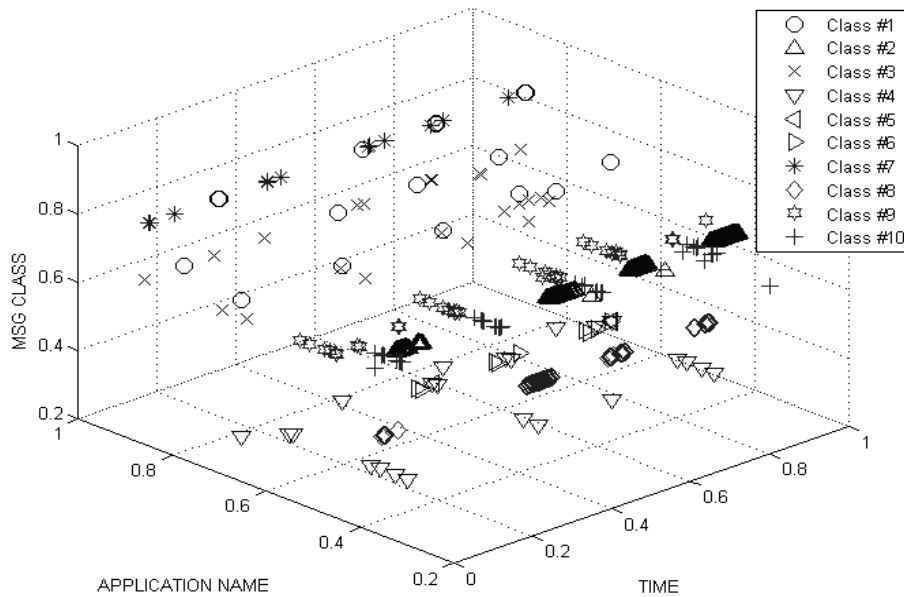


Figure 10: Clustering of PCA data using FCM.

8.3.3.3 SOM

The SOM was created using default parameters in the MATLAB implementation using a hexagonal lattice structure. The training was performed using the ‘batch’ training algorithm.

The u-matrix of the trained SOM (see chapter 17 for details) including maps of how the different attributes correspond to the map units is shown in Figure 11.

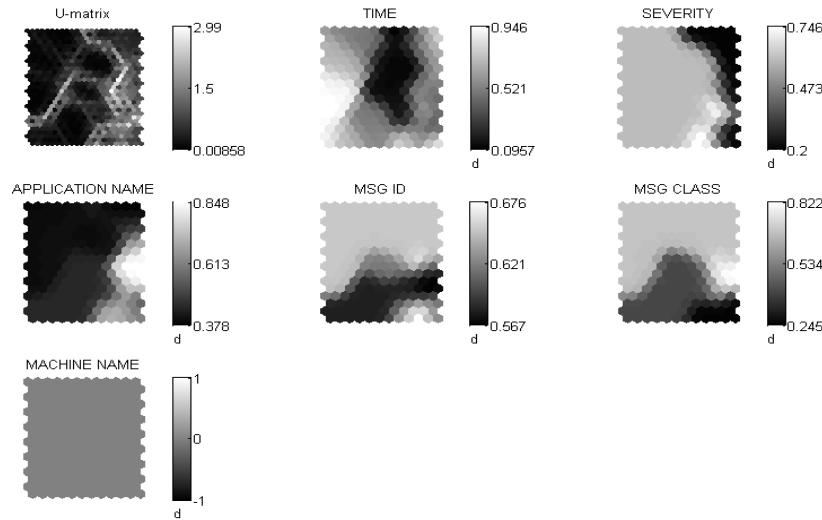


Figure 11: SOM u-matrix and attribute map.

To automatically identify any clusters in the SOM we used FCM to cluster map units. We found an optimum of 7 clusters (using the DB index) as shown in Figure 12. In Figure 13 one can see how many messages correspond to each unit in the map giving a better understanding of the clustering. Looking more closely at the four largest clusters (3,5,6 and 7) we find that they correspond very well to the clusters found using FCM clustering of the PCA data and the FCM clustering of the original data (within 3% both in size and uniqueness of messages). For example, Cluster 3 and 7 contain messages from the SFI only. This is virtually identical to Cluster 6 using FCM on the original data except that it has been split into two clusters. Looking at the attribute map, we see that the two clusters differ in ‘time’ and ‘message id’ indicating that there may be some finer granularity than what was discovered using the FCM approach.

The SOM analysis confirmed the results already found using FCM and PCA. There seems to be some main clusters of messages corresponding to the induced errors.

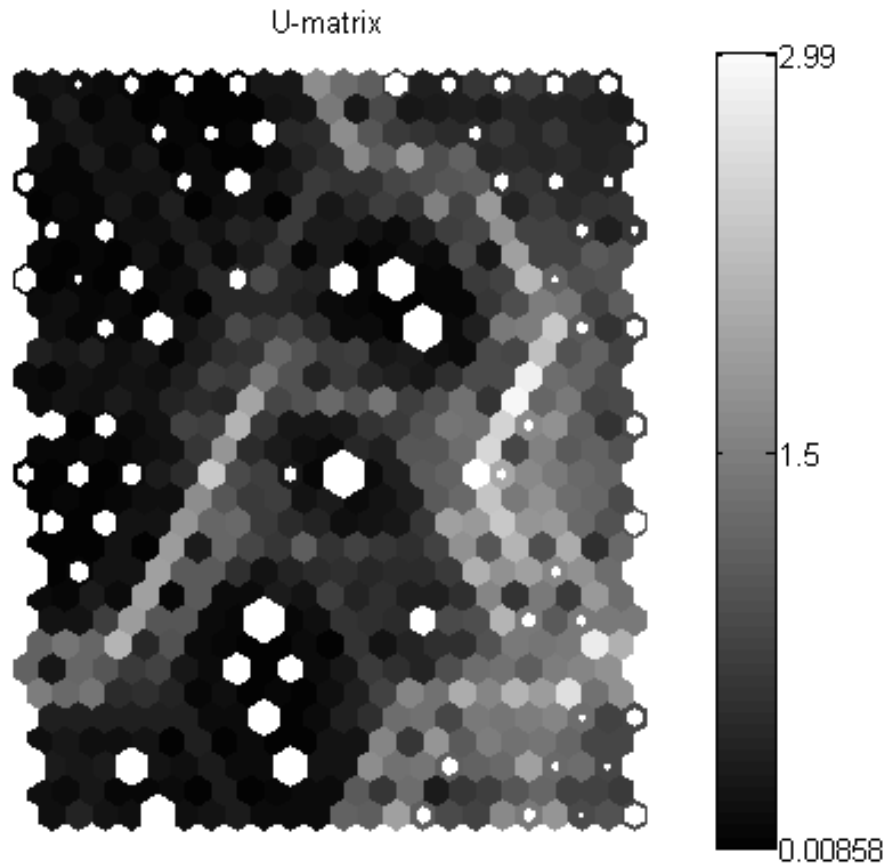


Figure 13: SOM showing how many observations correspond to each map unit (larger marker means more hits).

8.3.3.4.1 Datasets and networks

A decision needs to be made in terms of how best to input the data into the network. As messages are issued in time the idea is to identify an error situation as soon as it appears. The data must therefore be subdivided in such a way that this is possible. We also have to consider which attributes of each message should be used when training the MLP as some of them may or may not provide useful information. As a starting point we will only exclude the time of the message, as this will vary greatly from dataset to dataset and it would therefore not be possible for a network to generalize using this data. Three different datasets were created:

- Dataset-1: The data was used ‘as is’ with the exception of the excluded parameters.
- Dataset-2: PCA analysis was used (retaining 2 principal components) to reduce the number of inputs, and hence the training and run time of the MLP.
- Dataset-3: A ‘sampling’ of messages was done at 1 second intervals. If no message exists at the given time, a zero vector of the same size was used in its place. As a starting point, if two or more messages arrived at the same time frame the first one was used as an input.

All the datasets were then correlated with the time of the known error situations to create the data for training and testing. In addition to this we add a test set where the test system was used without inducing any errors. This is done to further test the generalization ability of the networks.

Four different networks were trained using the different datasets; see Figure 14:

- (1) The first approach divided Dataset-1 into blocks containing n consecutive messages as they arrived in time. These blocks were then used as a single input to a regular MLP. After some empirical testing, blocks of 5 messages was found to be reasonable.
- (2) The second approach used Dataset-2, and as in (1) grouped the data in blocks of 5 consecutive messages used as a single input to a regular MLP.
- (3) In the third approach we used Dataset-1; retaining the same order as they arrived in time. A TDNN was used to learn from the data using a time-delay of size 2. This method does not account for the fact that messages might be separated by large time spans and still be treated as ‘adjacent’ for the network.
- (4) For the fourth approach Dataset-3 was used to train a TDNN with a delay size of 5. A single message was used as input to the network.

8.3.3.4.2 Training results

Figure 16 shows the output of the trained network using the first approach (5 consecutive messages from Dataset-1). Since the target data is either 1 or 0 we will modify the network outputs so that they are rounded off to nearest integer. By making this modification for all the networks we achieved the results shown in Table 3. The table shows the percentage of correctly classified inputs for

training and test data. Note that the percentage rate of successful classification might be somewhat misleading as the training sets may contain a majority of similar and/or identical patterns making it trivial to obtain ‘good’ results. For example, if the target values contain 20% error situation, even a trivial solution of always returning false will lead to 80% success. To better estimate the performance of each network we will also use the following measures:

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

A sensitivity of 100% indicates that the all ‘positive’ cases are correctly classified.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

A specificity of 100% indicates that all ‘negative’ cases are correctly classified.

$$\text{Positive predictive value} = \frac{TP}{TP + FP}$$

A high positive predictive value (PPV) indicates that a positive result from the network is likely to be correct.

$$\text{Negative predictive value} = \frac{TN}{TN + FN}$$

A high negative predictive value (NPV) indicates that a negative result from the network is likely to be correct.

Looking more closely at the results in Table 3 we can see that all the networks achieve very good results on the training data (98-99%). The time-delay network from the fourth approach has an overall better performance for the three sets, but does have a significantly lower PPV for test set #2. For our problem, a low PPV means that we have to perform further tests of the system to verify the error, though these kinds of extra tests are likely to be performed in any case.

As we can see the use of PCA data also achieved good results for training, but worse for the two sets of test data. Though given the relatively more simple nature of the input data it is a method that should be considered depending on the requirements of the final application of the technique.

Overall the results show that it is possible to achieve very good results using several of the techniques. Further tests on larger datasets should be performed before making a final decision as to which approach to use.

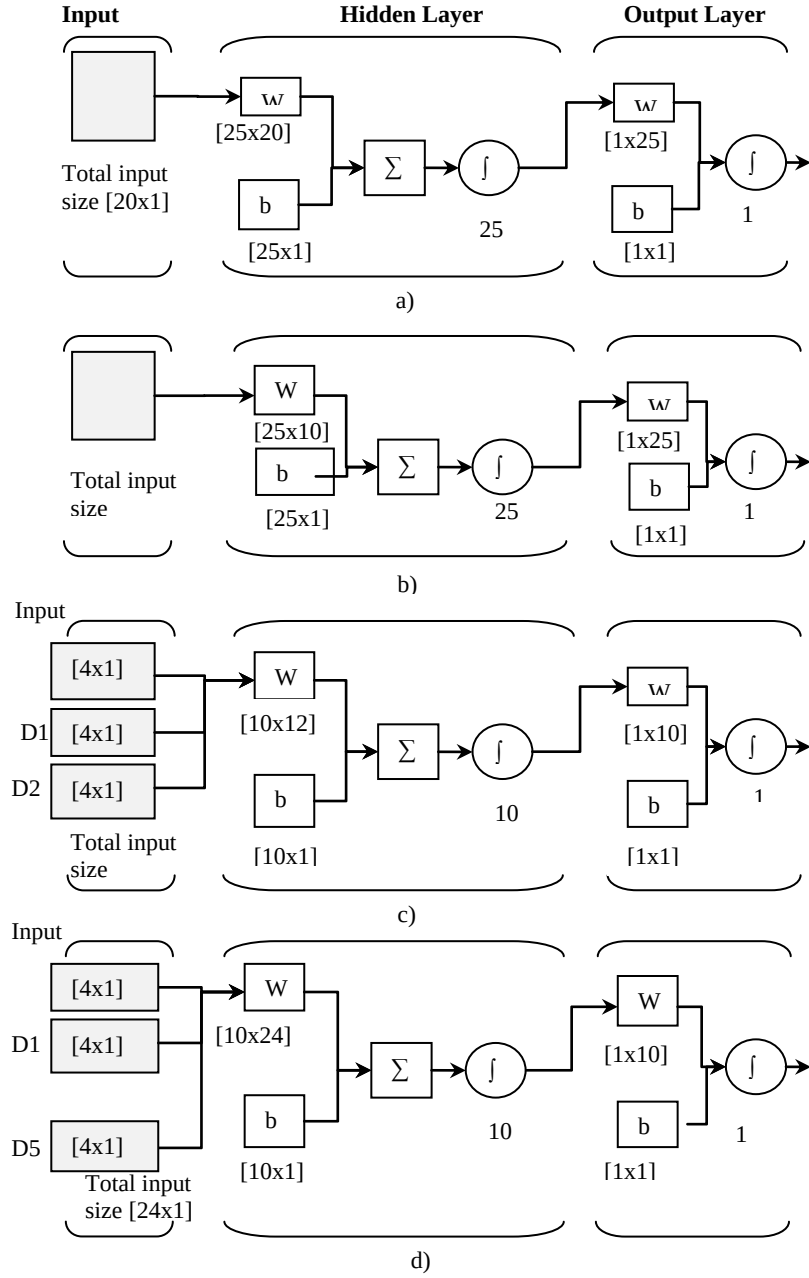


Figure 15: Layer diagram for the network for (a) the first approach, (b) the second approach, (c) the third approach and (d) the fourth approach.

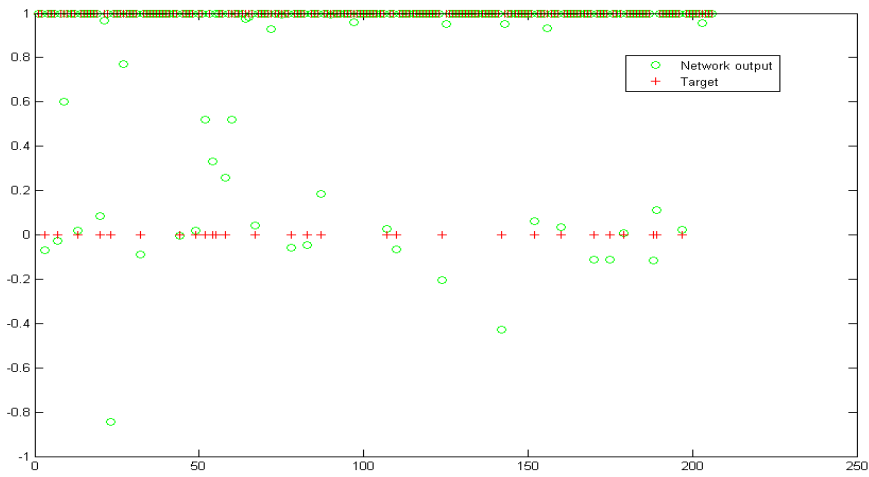


Figure 16: Network output vs. target data for MLP using regular data.

Table 3: MLP results.

Network	Set	Training data	FP /FN	Specificity	Sensitivity	Positive predictive rate	Negative predictive rate
Back propagation last 5 messages	Training	99.02%	2/0	91.3%	100%	98.92%	100%
	Test#1	95.45%	4/0	76.47%	100%	94.74%	100%
	Test#2	65.30%	17/0	65.3%	-	-	100%
Back propagation PCA data	Training	99.02%	2/0	91.3%	100%	98.92%	100%
	Test#1	81.81%	7/9	58.82%	87.5%	90.00%	52.63%
	Test#	18.37%	40/0	18.36%	-	-	100%
Time delay regular data	Training	98.06%	4/0	82.6%	100%	97.86%	100%
	Test#1	93.26%	5/1	70.59%	98.61%	93.42%	92.31%
	Test#2	83.67%	8/0	83.67%	-	-	100%
Time delay sampled data ¹	Training	97.99%	11/2	98.3%	98.47%	92.14%	99.61%
	Test#1	86.74%	36/1	87.10%	98.36%	62.50%	99.45%
	Test#2	96.88%	26/0	96.88%	-	-	100%

¹ The datasets used to train this network differs in size compared to the ones used for the training the other networks. One should therefore be careful when directly comparing the results as the dataset may contain more or less trivial and/or duplicated inputs.

8.4 DISCUSSIONS AND CONCLUSIONS

We have shown that the different clustering techniques can be used to group our data into meaningful clusters. These clusters have been confirmed using the clustering techniques, all showing similar results with the main clusters found using each technique remaining within 3% deviation both in terms of size and messages contained.

Different methods of pre-processing the data have been used and several different networks have been trained using these datasets. This has shown that it is possible to distinguish errors based on the messages received from the system, with the most successful network achieving worst case detection rates of 85% and specificity and sensitivity of 87% and 98%. Future work will try to distinguish different error situations and possibly give an indication of the severity of the problem. Further network topologies should be tested and parameter tuning should be done to optimize the networks. The methods will also be tested on a ‘full-scale’ system containing a much larger number of applications and hardware.

References

- Atlas Collaboration. (1994). ATLAS: technical proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN. CERN.
- (2003). ATLAS High Level Data Acquisition and Controls - Technical Design Report. CERN.
- Bezdek, J. C. (1981). Pattern recognition with fuzzy objective function algorithms. Plenum.
- Bezdek, J., & Pal, N. (1998). Some new indexes of cluster validity. IEEE Transactions on Systems, Man, and Cybernetics, Part B , 28, 301-315.
- Davies, D., & Bouldin, D. W. (1979). A cluster separation measure. IEEE Trans. Pattern Anal. Machine Intelligence , 1, 224-227.
- Dunn, J. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. 3.
- Evans, L. (2007). The Large Hadron Collider. New Journal of Physics , 9, 335.
- Kohonen, T. (1982). Self-organised formation of topologically correct feature map. Biological Cybernetics , 43, 56-69.

- Liu, Y., Ma, Z., Qian, H., & Lv, P. (2006). Early Identification of Machine Fault Based on Kernel Principal Components Analysis., 3, pp. 149-152.
- Murillo-Garcia, R., & Lehmann Miotto, G. (2007). A Log Service Package for the ATLAS TDAQ/DCS Group. *IEEE Transaction on Nuclear Science* , 54, 202-207.
- Riley, G., Culbert, C., & Lopez, F. (1991). C Language Integrated Production System (CLIPS). NASA Tech Briefs .
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. 318-362.
- Sloper, J. E., Lehmann Miotto, G., & Hines, E. (April 29 2007-May 4 2007). Dynamic Error Recovery in The ATLAS TDAQ System. Real-Time Conference, 2007 15th IEEE-NPSS , 1-5.
- Tien, D., Lim, K.-W., & Jun, L. (2-6 Nov. 2004). Comparative study of PCA approaches in process monitoring and fault detection. Industrial Electronics Society, 2004. IECON 2004. 30th Annual Conference of IEEE , 3, 2594-2599 Vol. 3.
- Waibel, A., Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. (1989). Phoneme recognition using time-delay neural networks. #IEEE_J_ASSP# , 37, 328-339.
- Wei-min, L., & Chein-I, C. (23-28 July 2007). Variants of Principal Components Analysis. Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International , 1083-1086.

Chapter 9

Discriminating Fiber-Reinforced Plastic Signatures Related to Specific Failure Mechanisms: Using Self-Organizing Maps and Fuzzy C-Means

XuQin Li, Carlos Ramirez, Evor L. Hines, Mark S. Leeson, Phil Purnell and Mark Pharaoh

Acoustic Emission (AE) can be used to discriminate the different types of damage occurring in composite materials, because any AE signal contains useful information about the damage mechanisms. A major issue in the use of the AE technique is how to discriminate the AE signatures which are due to the different damage mechanisms. Conventional studies have focused on the analysis of different parameters of such signals, say the frequency. But in previous publications where the frequency is employed to differentiate between events, only one frequency is considered and this frequency was not enough to thoroughly describe the behavior of the composite material. So we introduced the second frequency. A fast Fourier transform (FFT) is then applied to the signals resulting from the two frequencies to discriminate different failure mechanisms. The data was then analyzed using self-organizing map (SOM) and fuzzy C-means (FCM). The results shows that the two approaches have been very successful in discriminating fiber-reinforced plastic signatures related to specific failure mechanisms.

9.1 Introduction

The identification of the type of discontinuities or failure mechanisms within fiber-reinforced plastic (FRP) structures normally requires the use of local nondestructive testing (NDT) methods. Among which, acoustic emission (AE) has been widely used due to its high sensitivity. AE is generated by the material itself and once generated, will be detected by AE sensors and acquired by the AE data acquisition system. The recorded AE data are in the form of signal parameters, such as amplitude, duration, signal strength and hit. These are the key parameters in the structural evaluation. Normally, FRPs consist of more than one material. As a result, the failure mechanism will occur in one or a combination of materials. The various types of damage mechanism of FRP are matrix cracking, fiber breakage, fiber-matrix debonding, delamination, and fiber pullout.

Many research works have focused on observing AE signal parameters to identify failure mechanisms. This is usually done by using some intelligent system to perform pattern recognition of AE data relating to different failure mechanisms. The aim of the present study is to use intelligent systems to identify the unique characteristics of different failure mechanisms from AE data so that it can recognize the distinctive pattern of an unknown failure mechanism sample.

For the current study, pattern recognition by means of intelligent systems is applied to a group of AE bursts rather than to individual sensor hits. This is because an individual hit, which may be one among thousands, is not likely to be representative of emission. In the work here we use two widely used approaches self-organizing map (SOM) and fuzzy C-means (FCM).

9.2 Theoretical Background to Self-organizing Map (SOM) and Fuzzy C-means (FCM)

9.2.1 A typical neural network structure

Neural networks (NNs) first started to be developed in the 1940s, motivated by for example an attempt to simulate the human brain on computers. NNs have been applied to prediction problems, to pattern recognition, classification and optimization problems. As long as we have enough data, we can train a NN

system so that it can perform some practical functions. In the research we describe here, the NN system will be used for pattern recognition of AE data.

Usually, a NN consists of a number of neurons which are distributed over several layers. Input information, which could be a series of numbers representing data, is input into neurons in an input layer. Then the data is propagated to the output through subsequent neurons. As the information travels through the network, the input is modified. Subsequently, the output neurons will generate the output.

Normally, the neurons are arranged into different layers as shown in Figure 1. Each neuron in each layer performs a different activity as described in the following three steps:

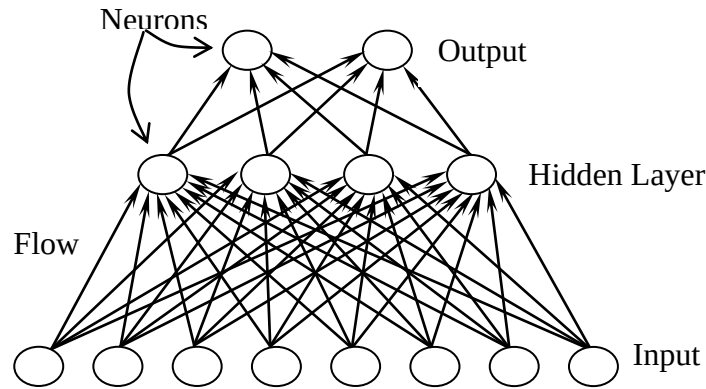


Figure 1: Basic architecture of a typical neural network.

(1) In this model the input neurons only receive information in the form of input values. They transfer the data to the next layer. Neurons in the input layer perform no neural function.

(2) Hidden neurons receive the output from the input neurons or other hidden neurons through the connections. Each connection has a weight by which to multiply the signal before it enters the hidden neurons. A hidden neuron usually receives more than one signal. The signals will be combined and transferred to an appropriate activation function. The output of the activation function will then be usually transferred to output neurons but sometimes via hidden neurons (as suggested in Figure 2).

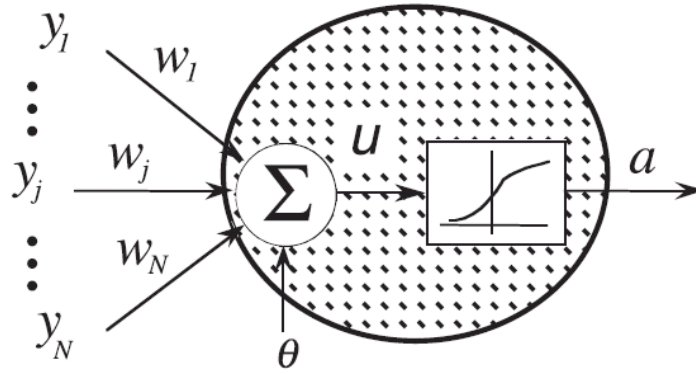


Figure 2: An example artificial neuron.

(3) Output layer neurons behave similarly to the ones in the hidden layer except that the pattern of the outcome from output neurons may be interpreted as the outcome of using the NN to process the input data.

Training the system is the key to determining whether or not the NN will be able to process the input data appropriately. In a supervised NN the training process allows NNs to compare the output result with the real answer for every training input. The difference between the generated output and the desired answer is called the ‘error’. The NN will adjust the weight of each connection in such a way as to reduce the error. At the beginning, the NN might produce a large error. However, after several cycles of training with the set of data, the error should reduce. Here we choose self-organizing map (SOM) as the structure of neural networks for analyzing the data set, because for a given set of data, the algorithm is required to self-organize the output layer as a map on which similar shapes will be detected as clusters of neurons which are close to each other. The following sections provide an overview of the SOM.

9.2.2 Self-organizing map (SOM)

According to Kohonen SOMs are connectionist techniques capable of generating topology preserving clustering. The SOM algorithm was developed by Kohonen to transform a data set of arbitrary dimensions into a one or more dimensional discrete map.

An example of the network architecture of a Kohonen SOM is shown in Figure 3, after Godin. It consists of a two dimensional array of $n \times n$ discrete units and the Kohonen network associates each of the vector inputs to a representative output.

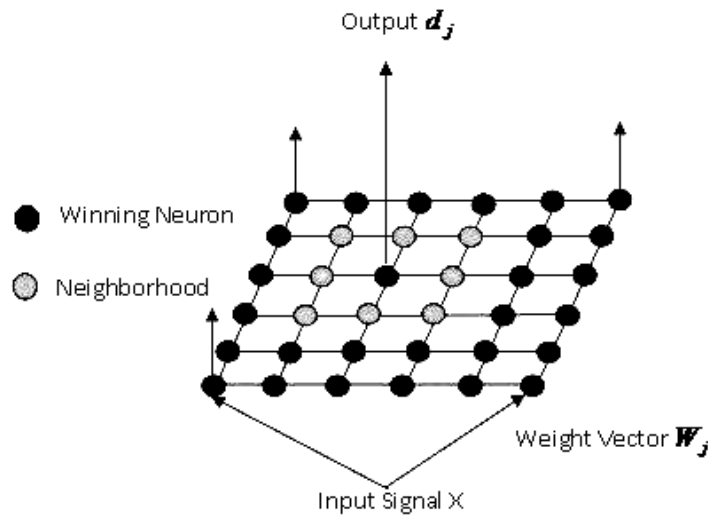


Figure 3: Schematic of a self organizing map topology.

Suppose w_{ij} are the components of the weight vector W_j , connecting the inputs i to output node j ; the x_i are components of the input vector X , the output of the neuron is the quadratic (Euclidean) distance d_j between these two vectors (see Figure 3). The description of each step for using the data to train the SOM is summarized as follows:

- (1) Initialize all the weights to random values between 0 and 1.
- (2) Randomly select an input vector X ; present it to all the neurons of the network and evaluate the corresponding quadratic distance output d_j according to the following equation, which are stages 1 and 2 of Figure 3.

$$d_j = \|X - W_j\|^2 = \sum_{i=1}^n (w_{ij} - x_i)^2 \quad (1)$$

Where n is the number of input vector components.

(3) Select the neuron with the minimum output d_j as the winning neuron, i.e. the nearest vector to the input vector (Stage 3 in Figure 3). Let j^* denote the index of the winner, the minimum output will be:

$$d_{j^*} = \min_{j \in \{1, \dots, m\}} \|X - W_j\|^2 \quad (2)$$

where m is the number of neurons.

(4) Update the weight of the winning neuron (Stage 4 in Figure 3) according to Eq.(3).

$$w_{ij^*}(k+1) = w_{ij^*}(k) + \eta(k)[x_i(k) - w_{ij^*}(k)] \quad (3)$$

where k is the learning iteration count and η the gain term.

(5) The neighbors of the winning neuron, defined by the neighborhood function $N(j^*)$ (the neighborhood function $N(j^*)$ defines how many neurons in the neighborhood of the winning one will be updated for each learning input). are also updated following Eq.(4) and (5) (Stage 5 in Figure 3).

$$w_{ij}(k+1) = w_{ij}(k) + \eta(k)[x_i(k) - w_{ij}(k)] \quad (4)$$

if $j \in \text{neighborhood of } j^*, N(j^*)$

$$w_{ij}(k+1) = w_{ij}(k) \quad (5)$$

if $j \notin \text{neighborhood of } j^*$

(6) Repeat steps 2 to 5 until all the input vectors X had been used at least once.

9.2.3 Fuzzy C-means algorithm

The objective of cluster analysis is to classify objects according to similarities among them, and organize a data set into subsets. Usually, clustering techniques are considered to be among the unsupervised methods, they do not use prior information about groupings before clustering. It can detect the underlying structure in a data set, thus it can be used for classification and pattern recognition.

Different kinds of definitions of a cluster can be formulated, depending on the objective of clustering. Usually we regard a cluster as a group of objects which

bear more similarities to one another than to the members of other clusters. We can define similarity by means of a distance norm. It can be measured among the data vectors themselves, or as a distance from a data vector to some prototypical object (perspective center) of the cluster. The prototypes may be vectors of the same dimension as the data objects and are usually not known beforehand, and usually they are sought by the clustering algorithms heuristically with the partitioning of the data.

Fuzzy C-means (FCM) algorithm, also known as Fuzzy ISODATA, is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade. This technique was originally introduced by Jim Bezdek in 1981 as an improvement on earlier clustering methods. It provides a method that shows how to group data points that populate some multidimensional space into a specific number of different clusters. The FCM-based algorithms are in practice the most widely used fuzzy clustering algorithms.

The objective of the FCM clustering algorithm is to minimize an objective function called C-means functional, which is formulated as:

$$J(U, V) = \sum_{j=1}^C \sum_{i=1}^N (u_{ij})^m \|x_i - v_j\|^2 \quad (6)$$

Where $X = \{x_1, x_2, \dots, x_N\}$, $x_i \in R^n$, represent a given set of feature data. $V = \{v_1, v_2, \dots, v_C\}$ are the cluster centers. $U = (u_{ij})_{N \times C}$ is a fuzzy partition matrix, in which each member u_{ij} indicates the degree of membership between the data vector x_i and the cluster j . The values of matrix U should satisfy the following conditions

$$u_{ij} \in [0, 1], \quad 1 \leq i \leq N; \quad 1 \leq j \leq C \quad (7)$$

$$\sum_{j=1}^C u_{ij} = 1, \quad 1 \leq i \leq N \quad (8)$$

The exponent $m \in [1, \infty]$ is the weighting exponent, which determines the fuzziness of the clusters. The most commonly used distance norm is the standard

Euclidean distance $d_{ij} = \|x_i - v_j\|$, although Babuska suggests that other distance norms could produce better results.

Minimization of the cost function $J(\mathbf{U}, \mathbf{V})$ is a nonlinear optimization problem, which can be minimized with the following iterative algorithm:

Step 1: Initialize the membership matrix $\mathbf{U}$ with random values so that the conditions in Eq.(7) and (8) are satisfied.

Choose appropriate exponent m and the termination criteria.

Step 2: Calculate the cluster centers $\mathbf{V}$ according to the equation:

$$v_j = \frac{\sum_{i=1}^N (\mu_{ij})^m x_i}{\sum_{i=1}^N (\mu_{ij})^m}, \forall j = 1, \dots, C \quad (9)$$

Step 3: Calculate the new distance norms:

$$d_{ij} = \|x_i - v_j\|, \forall i = 1, \dots, N, \forall j = 1, \dots, C \quad (10)$$

Step 4: Update the fuzzy partition matrix $\mathbf{U}$:

If $d_{ij} > 0$ (indicating that $x_i \neq v_j$)

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}} \quad (11)$$

Else $\mu_{ij} = 1$

Step 5: If the termination criterion has been met, stop.

Else go to Step 2.

We can see that the FCM algorithm is a simple iteration through Eq.(9) to (11).

A suitable termination criterion will be used to evaluate the cost function (Eq.6) and to see whether it is below a certain tolerance value or if its improvement, compared to the previous iteration, is below a certain threshold.

We can also use the maximum number of iteration cycles as a termination criterion.

9.3 Test Apparatus and Procedure

In this section we first give a brief introduction to the AE equipment and the AE software system. After that a description of test procedure is included in the second subsection.

9.3.1 Acoustic emission equipment

Due to the requirement to have a flexible and feasible process for manufacturing the test coupons during the current research ‘hand lay up’ was the most suitable solution. The fiber glass volume fraction that is normally achieved by this manufacturing process is 57%. All the Glass/Polyester plates were manufactured in-house in order to have a closer control of the parameters that influence the mechanical properties; and ultimately the acoustic response of the materials. These parameters included fiber orientation and coupon dimensions. The coupon dimensions for the current research were 20 mm x 200 mm x 1.8 mm with three layers of unidirectional glass tape. Each plate was cut into different orientations in 15° steps against the direction of the load.

A broadband WD piezoelectric sensor manufactured by Physical Acoustic Corporation was used to capture the stress waves (AE events). After capturing the event, the piezoelectric sensor transforms the mechanical deformation of the material surface into an electrical signal. This signal is transferred to a sound card with a high sampling rate, which processes it and displays it as a wave. The software used with this sound card can display on screen the transient waveform for each event. Each event can then be individually exported into an ASCII file, which is then processed by a program written in the ‘R’ programming language (open source code language). With the use of this program it is possible to perform, and visualize a fast Fourier transformation (FFT) for each event and select the most dominant frequency, which is to be used as the descriptor of the event. The distribution of these dominant frequencies as the test progresses shows the trends of the events occurring in the test specimen.

The acoustic emissions software was the TRA (transient recorder package) from Physical Acoustics Corporation 2001. The hardware was a sound card AEDSP-32 (Acoustic Emissions Digital Signal Processor) with a digital signal processor by Texas Instruments (TMS320C40). It has a maximum sampling speed of 8 MHz and 16 bits resolution, between 20 and 100 dB.

From the literature review, it appears that there are several different approaches to the task of event recognition analysis, from simple hit analysis to amplitude-based and frequency-based studies. The amplitude of an acoustic wave is attenuated as the wave passes through the material. Thus the amplitude recorded at a sensor depends upon the distance the sensor is from the source of the emission. Based on that, for the present study the dominant frequency of the signals was considered to be the most relevant characteristic.

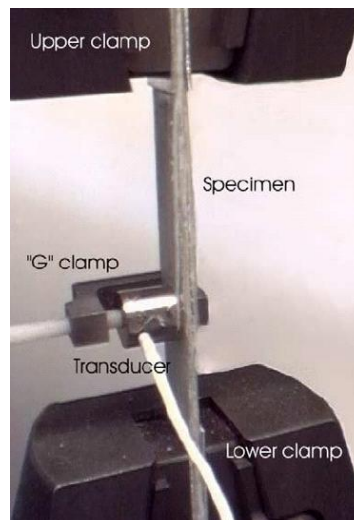


Figure 4: Test configuration, sensor coupled to the specimen using a G clamp.

9.3.2 Test procedure

According to Qi 90% of AE activities for glass fiber reinforced (GFR) composite materials are concentrated in the frequency range 10–550 kHz. One wide band sensor model PAC WD with an operational range of frequencies between 100 and 1000 kHz with a resonant frequency at 125 kHz was used. The sensor was attached to the specimen by means of a G aluminum clamp with a

plastic screw. The surface of the sensor was covered with silicon grease in order to provide good acoustic coupling between the specimen and the sensor. The signal was detected by the sensor and enhanced by a 2/4/6-AST pre-amplifier. The gain selector of the preamplifier was set to 40 dB, as the material used was not considered 'loud'.

Figure 4 shows the test configuration, with the sensor clamped to the test specimen and the INSTRON grips where used for applying the tensile force.

9.4 Data Analysis

Initially, the MISTRAS software that records the counts from each event was used. It was observed that as each test progressed, the number of counts increases significantly. Nevertheless, the data do not provide any information about the type of failure or the location of the failure. The TRA software, although using the same hardware, acquires the information in a different way; it records the complete waveform and stores the event. A transformation is then applied to define the frequencies related with these waveforms, which facilitates their analysis. According to Qi these frequencies are useful to “distinguish different AE signals from various possible failure modes in fiber reinforced composites”. A waveform event contains several component frequencies, and it is not possible to determine them visually, therefore a program was needed that calculates the FFT for each waveform. The procedure produces a power vs. frequency graph of a waveform (power spectrum). Higher powers contained at particular frequencies are then shown as peaks on this representation of the waveform. If more than one frequency is super-imposed on the waveform each component appears as the secondary frequency and so on.

The program that generates the FFT identifies by default the five frequencies from each signal with the highest power peaks as shown in Figure 5: (a) transient event waveform; and (b) FFT power spectrum. Sometimes the power difference between the highest power frequency and the subsequent lower power frequencies is so distinct that it can be assumed that the single frequency constitutes the definition of that waveform. Based on our analysis of the data, it was evident that we can assume that the event primary frequency (1stEF) alone was enough to characterize each acoustic event.

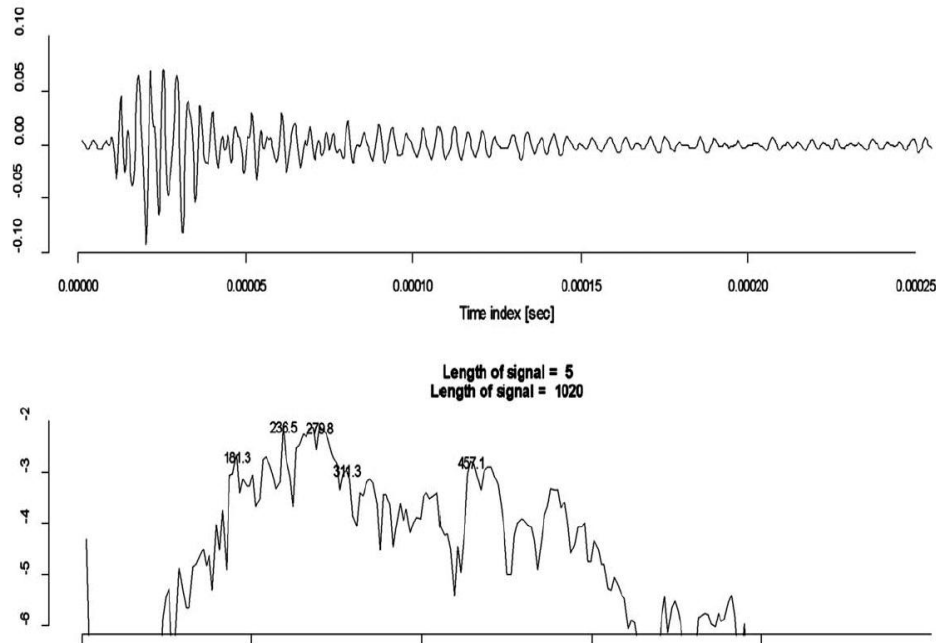


Figure 5: (a), (top) transient event waveform; (b) (bottom) FFT power spectrum.

Figure 5 shows a typical AE event recorded by the system. The FFT power spectrum in this specific case shows several peaks, the highest one being 280 kHz and having a second higher peak at 236 kHz. The FFT power spectrum can be used as the ‘fingerprint’ of each event and therefore may be used as a means of distinguishing between them. A possible interpretation is that the waveform is formed from several individual sources associated with an event and each one has its own frequency.

In previous publications where the frequency is employed to differentiate between events, only one frequency is mentioned. As a first approach the frequency with the highest power value was used as an identifier for the events. However, this frequency was not enough to thoroughly describe the behavior of the composite material.

Previous tests have found that the 1stEF contributes more than 50% of the total power of the waveform; the remaining highest power frequencies contribute the remaining power. Some researchers (e.g. Papadakis, 2004) attempted to find a relationship between the first and the second frequency, through a direct plot of

the two of them and analyzing any relationship by two approaches (connecting vectors and kernel density estimations).

Figure 6 shows a plot of the 1st frequency against the 2nd frequency from one set of FFTs calculated from waveforms recorded on a tensile test of 0° Carbon/Polyester. It can be seen that there is some clusterings of values where a relationship could be drawn, however according to Papadakis this representation requires more information for it to be used as a sensible tool to differentiate the events that originated each waveform.

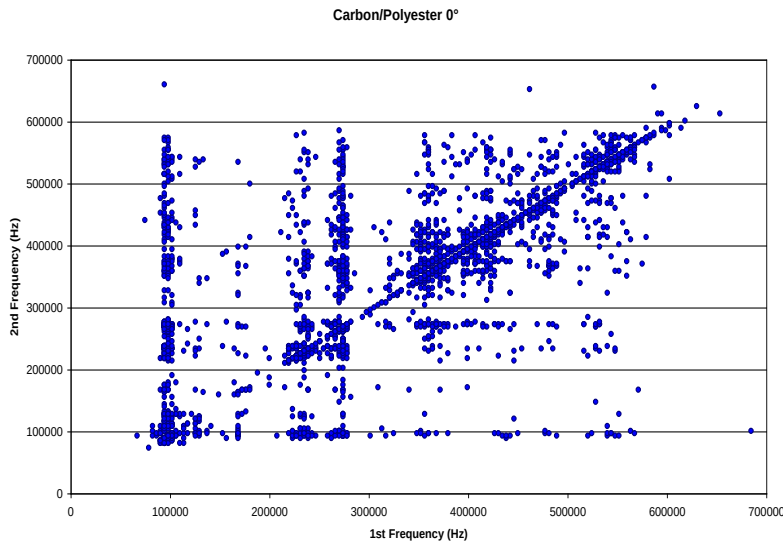


Figure 6: Relationship between the 1st and 2nd highest power frequencies.

Although these preliminary results supported the hypothesis that frequency could be used as an identifier for different failure events in composite materials, nevertheless subsequent analysis showed that using only the first frequency as a descriptor is not enough to produce optimal results. It was therefore decided to add the second frequency and to analyze the data obtained from different fiber orientations using neural networks.

In what follows, a self organizing map (SOM) is used to explore possible classification of the AE data recorded from one of the tests. The input training

data for this first run was the 1st and 2nd highest power frequencies from a tensile test of Glass/Polyester 0 °(Figure 7).

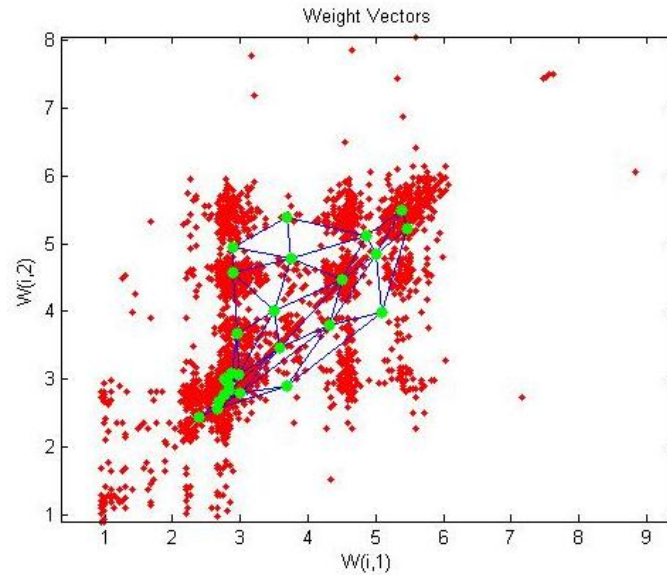


Figure 7: SOM for the 1st and 2nd frequencies from Glass/Polyester 0 °.

As Figure 7 shows the data (pairs of 1st and 2nd frequencies) form clusters (dots of the mesh). These clusters may be subsequently reduced in number while the clustering process continues and more pairs of frequencies are clustered together. The lines represent the connecting vectors between clusters whilst the scattered dots are the plot of the 1st vs. 2nd frequency used as a reference of the location for the classes. The scattered dots plot is similar to that reported by Papadakis , where the X axis corresponds to the first frequency and the Y axis corresponds to the second frequency.

9.5 Test Results

The SOM and FCM clustering processes are usually unsupervised. They will result in the best solution after a number of iterations. The data from 7 different Glass/Polyester fiber orientations shown in Figure 8 were passed through SOM and then FCM, as follows:

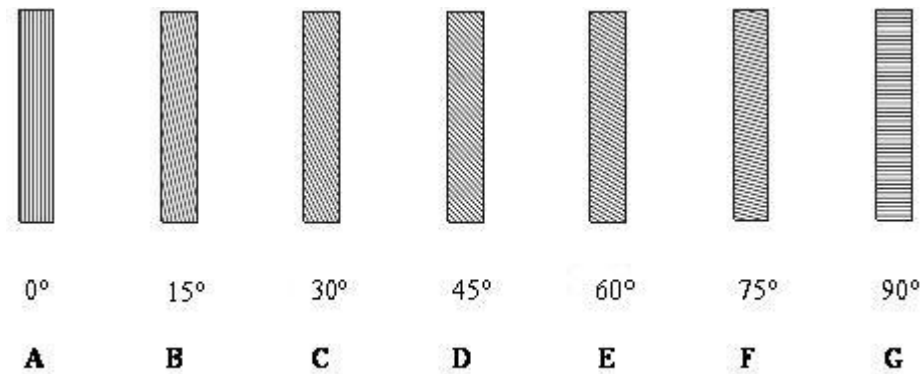


Figure 8: Diagram of fiber orientations.

The SOM was constructed to have 9 output neurons into which the results are clustered. From these 9 output neurons a five class value distribution was formed. The way these classes were defined was based on the region into which each of the output neurons fell. For some tests some of the regions did not capture any of the output neurons, therefore that class presented no values to contribute to the total number of events. The bar plot for each test is shown next to the SOM output map where the clusters for each corresponding class has been drawn.

The FCM algorithm starts with an initial guess for the cluster centers, which is intended to mark the mean location of each cluster. The initial guess for these cluster centers is most likely to be incorrect. Additionally, FCM assigns every data point a membership grade for each cluster. By iteratively updating the cluster centers and the membership grades for each data point, FCM iteratively moves the cluster centers to the right location within this dataset. This process is based on minimizing an objective function that represents the distance from any given data point to a cluster center weighted by that data point's membership grade.

We will now consider Glass/Polyester response at 0°, 15°, 30°, 45°, 60°, 75° and 90°.

9.5.1 0°Fiber orientation

Figure 9 shows the SOM output map with the 9 neurons positioned on their correspondent locations. Notice that some of the neurons are closer to each other (i.e. neurons 4, 5, 6, 7, 8 and 9), these neurons are eventually grouped into a single class by the neural network. Therefore the values that had been related to each neuron by the network will be considered to belong to that class. Once those neurons that are in the same vicinity are identified by the network as belonging to the same class, then a plot with the number of events corresponding to each neuron is created.

The FCM algorithm converges after 13 iterations. Figure 10 shows three clusters in the map which were all represented by a cluster center. The first cluster lies around 2.5k Hz, while the second and the third lie around 4.5k Hz and 5.5k Hz respectively. Each cluster consists of some events (with the first cluster 5761 events, the second and the third cluster 457 and 834 events respectively) which amount to 7052 events in total.

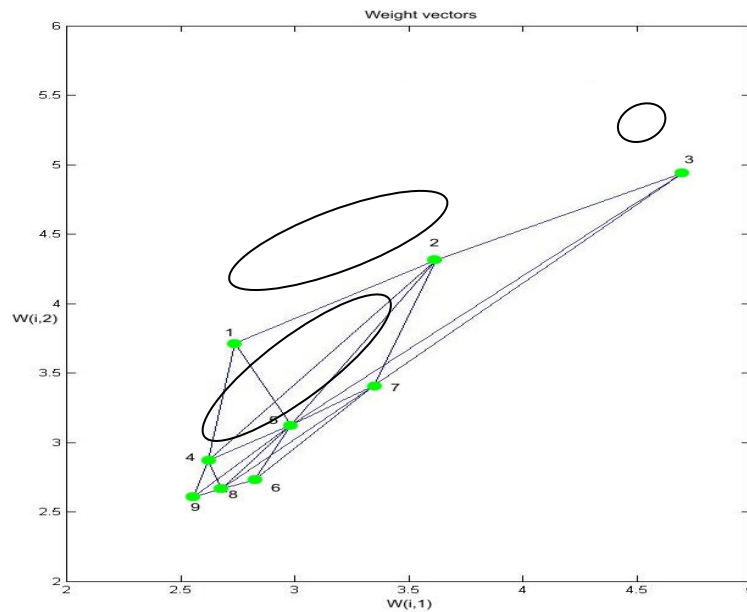


Figure 9: SOM output for 0°Glass/Polyester.

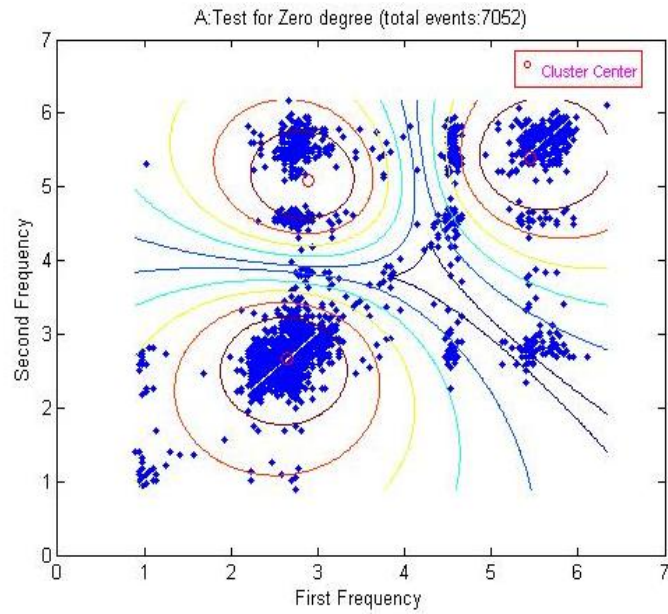


Figure 10: FCM output for 0 °Glass/Polyester.

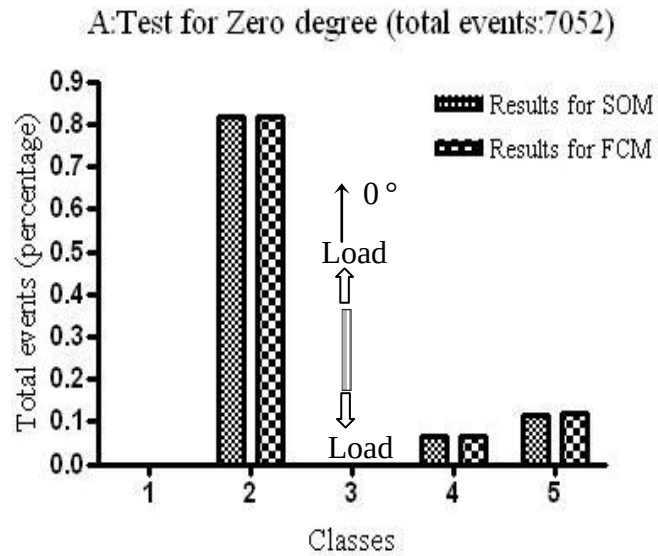


Figure 11: Normalized events per class for 0 °Glass/Polyester.

In order to compare the consistency of these two algorithms, we plotted a vertical bar graph showing the percentage of the events in each class for each algorithm. Figure 11 shows the number of events per class. The X axis accounts for the 5 different classes found during the training process whilst the Y axis has been normalized against total number of events for that test. This is done so that it is easier to compare the relative population for each class per test across all the tests.

9.5.2 5 °Fiber orientation

Figure 12 shows the SOM output map for 15 ° Glass/Polyester. The output neurons have been positioned by the network in different locations compared to those found for 0 ° Glass/Polyester, see Figure 9.

For this test a new class has appeared (Class 1) corresponding to neuron 1. Class 2 contains the values gathered using neurons 2, 3, 4, 5, 6 and 7 accounting for almost 60% of the total events of the test, see Figure 12.

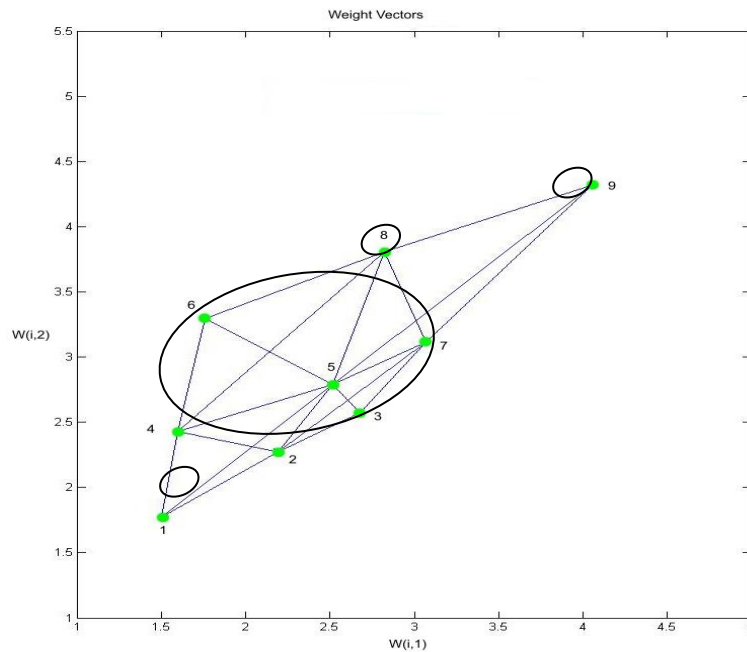


Figure 12: SOM output for 15 ° Glass/Polyester.

Figure 13 shows the FCM output after it converges through 23 iterations. There are clearly four clusters in the map which were all represented by a cluster center. As in the case of the SOM, the cluster which consists of 3027 events dominates the classes while the other three consists of 806, 768, 850 events respectively.

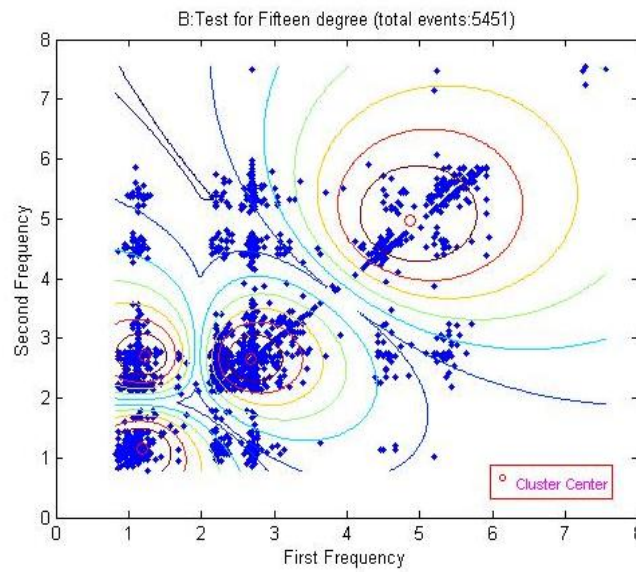


Figure 13: FCM output for 15 °Glass/Polyester.
B:Test for Fifteen degree (total events:5451)

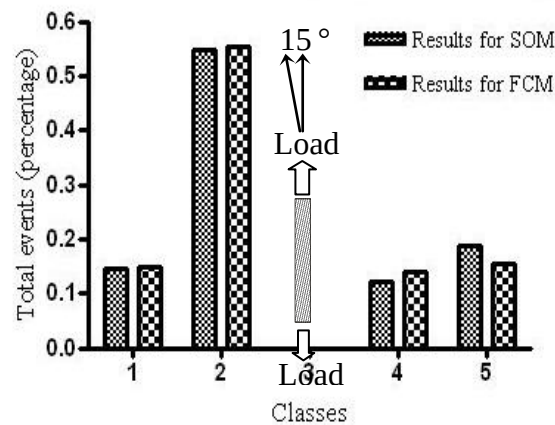


Figure 14: Normalized events per class for 15 °Glass/Polyester.

Figure 14 shows a bar graph of the percentage of events per class for 15° Glass/Polyester. Although there are some slight differences between FCM and SOM, again Class 2 accounts for the majority of the events while the other three classes (Class 1, Class 4 and Class 5) account for about 15% each for both algorithms.

9.5.3 30° Fiber orientation

Figure 15 and 16 show the SOM and FCM output map for 30° Glass/Polyester. Four classes were found in both figures; in Figure 13, neuron 3 corresponded to Class 1, neurons 7 and 8 to Class 2, neurons 1, 2, 5 and 6 to Class 3 and neurons 4 and 9 to Class 5. FCM converges after 41 iterations. And both algorithms indicate that the second class dominates all the events. The distribution of events across the classes in Figure 17 for the 30° fiber orientation, showed a similar tendency to that for the 15° bar chart showed in Figure 14. Nevertheless the total number of events is significantly different for both tests; the 15° fiber orientation has almost 5 times more events (5451) than the 30° test (1251).

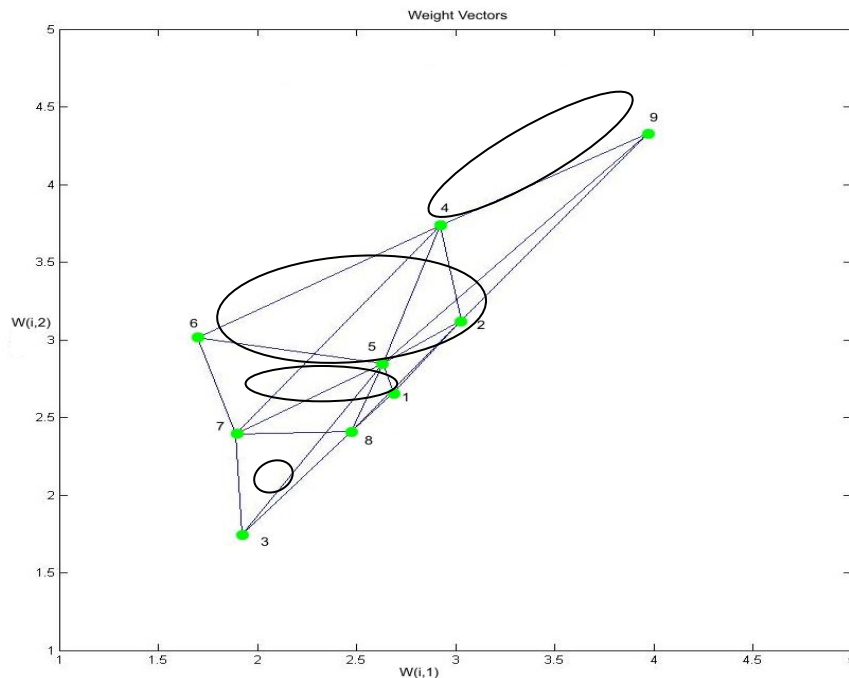


Figure 15: SOM output for 30° Glass/Polyester.

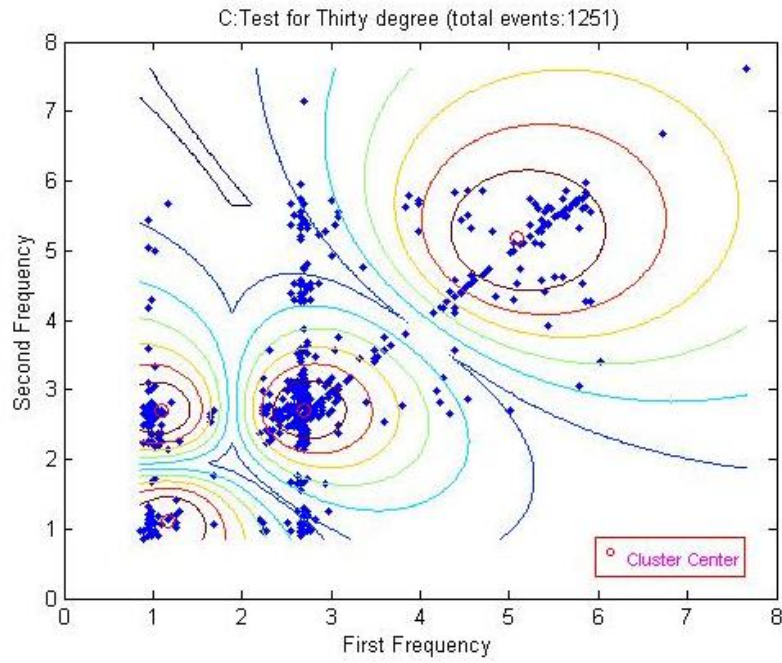


Figure 16: FCM output for 30 °Glass/Polyester.

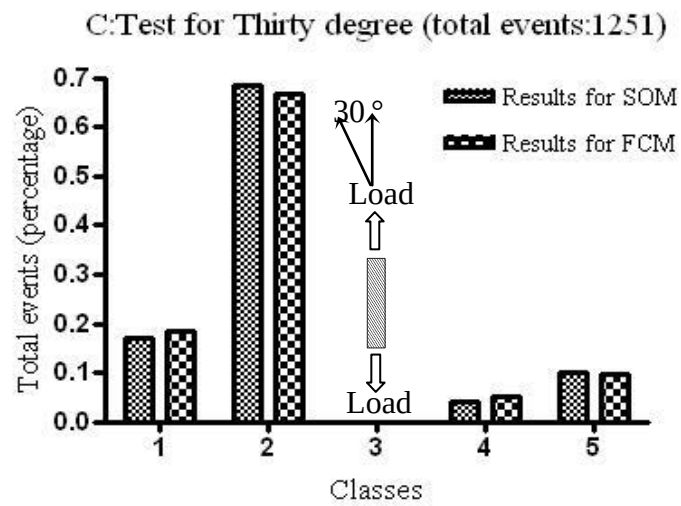


Figure 17: Normalized events per class for 30 °Glass/Polyester.

Figure 17 shows the normalized number of events corresponding to each class from the SOM and FCM. It can be seen that in both algorithms Class 3 has no events for 30° fibbers orientation and class 2 contributes with almost 70% of the events. The others classes contributes less than 40% for the rest of the events.

9.5.4 45° Fiber orientation

One special fiber configuration is the 45° against the direction of the load; its SOM output and FCM output are shown in Figure 18 and Figure 19 respectively. 4 Classes are identified for the 45° test, which is a natural output following the previous fiber orientation test results (Figure 11, Figure 14, and Figure 17) as they rotate away from the direction of the load. The number of events for each class for the 45° fiber orientation test results showed a more even distribution than the previous ones, 20% for Class 1, 36% for Class 2, 18% for Class 4 and 26% for Class 5 in SOM and 20% for Class 1, 40% for Class 2, 18% for Class 4 and 22% for Class 5 in FCM.

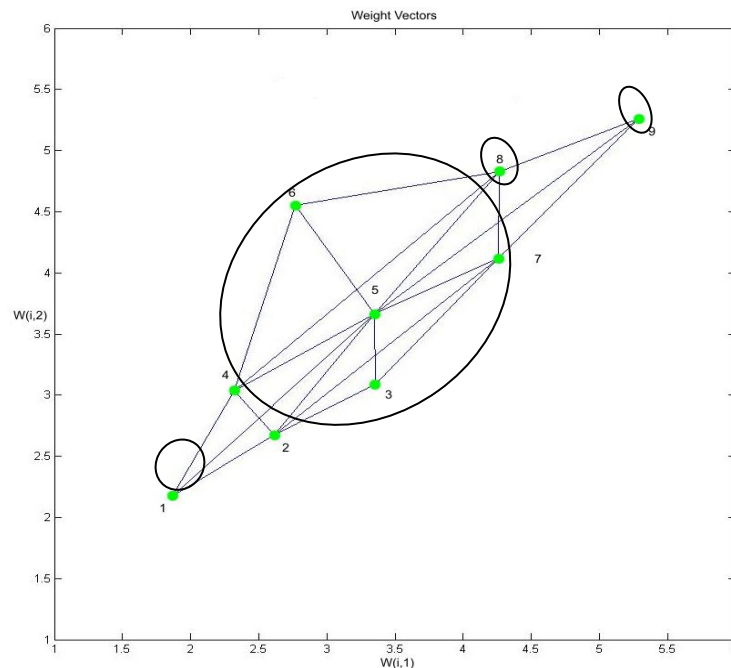


Figure 18: SOM output for 45° Glass/Polyester.

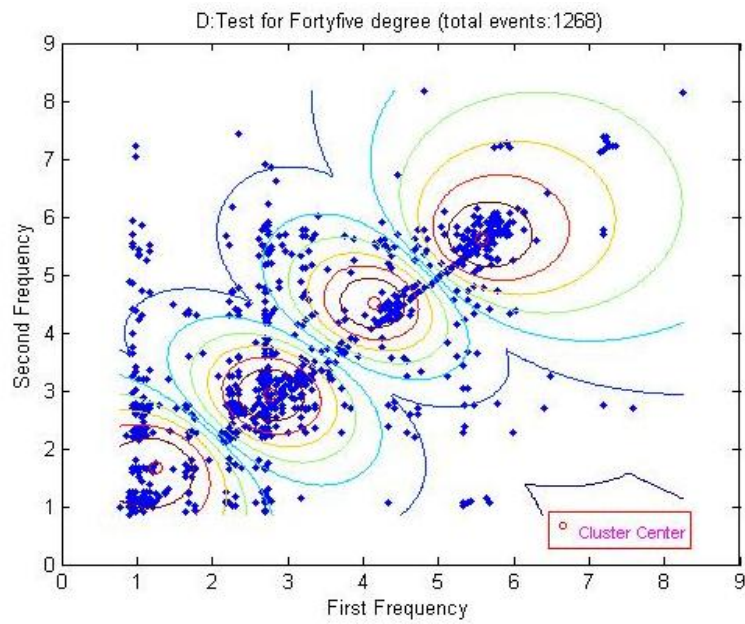


Figure 19: FCM output for 45 °Glass/Polyester.

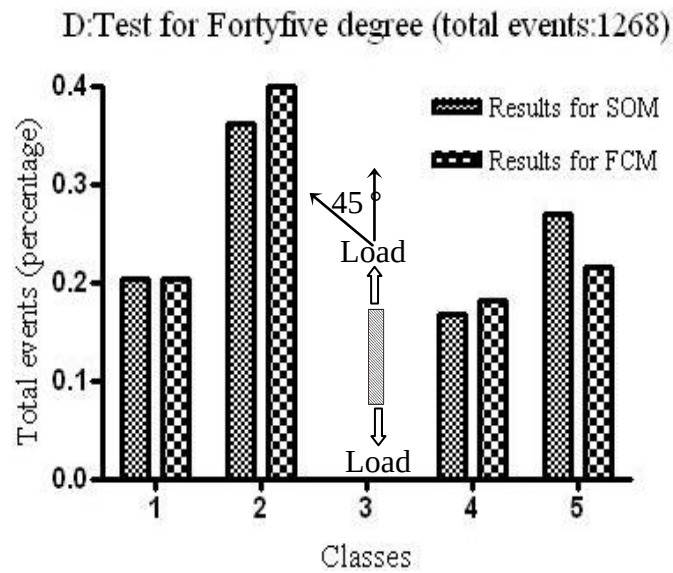


Figure 20: Normalized events per class for 45 °Glass/Polyester.

Figure 20 shows the classes that were constructed from the 45° Glass/Polyester SOM neuron map and FCM clustering. The micro-mechanical events appearing during this test will be subjected to tensile and shear stresses applied to the material evenly, due to the orientation of the fibers.

Between 15° and 45° fiber orientations, all the classes were represented except for Class 3. For the 0° fiber orientated material Class 1 was also missing. Consideration of the phenomenon related to their micro-mechanical failure events will be described in the following sub-sections.

9.5.5 60° Fiber orientation

From the test for 60° and up to the 90° Classes 4 and 5 were not represented and instead Class 3 appeared. As can be seen in Figure 21 the SOM of output neurons and in Figure 22 the FCM clusters for 60° show a clustering of neurons in classes where Classes 4 and 5 are not present.

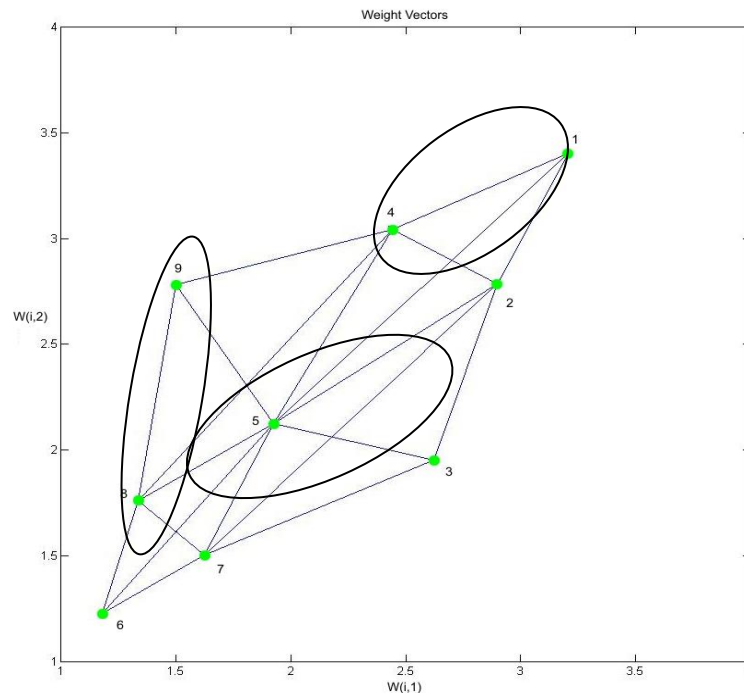


Figure 21: SOM output for 60° Glass/Polyester.

In the SOM algorithm, neurons 6, 8 and 9 are clustered in Class 1; neurons 3, 5 and 7 are in Class 2 and neurons 1, 2 and 4 are grouped into Class 3. In the case of FCM (Figure 22) it converges to three typical classes after 54 iterations. As indicted above and displayed here, starting from this fiber orientation and up until the 90° configuration Class 3 appeared and showed an increase in the number of events related to it as the fiber rotates away from the direction parallel to the load. On the other hand for this same offset of the fibers, Class 2 has gradually reduced in size as tests proceed from 60° to 90°.

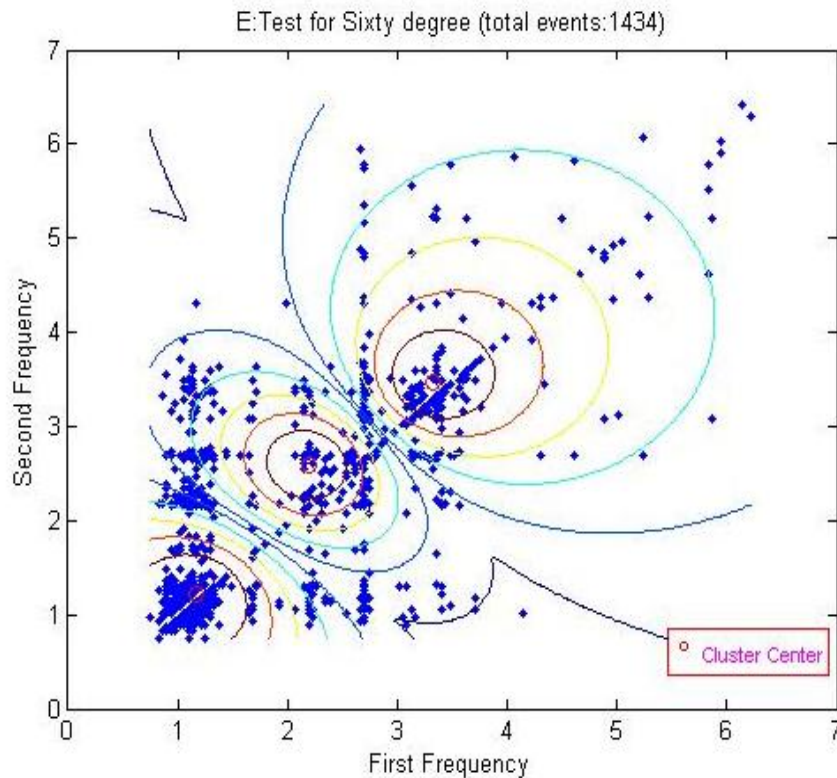


Figure 22: FCM output for 60° Glass/Polyester.

Figure 23 shows a normalized events bar chart for 60° fiber orientation of Glass/Polyester for both algorithms. It is noticeable that for this fiber configuration there are no events that could be related to either Class 4 or Class 5. Class 1 does not show any obvious pattern in terms of the change of its size as the fiber orientation turn around away from the direction parallel to the load.

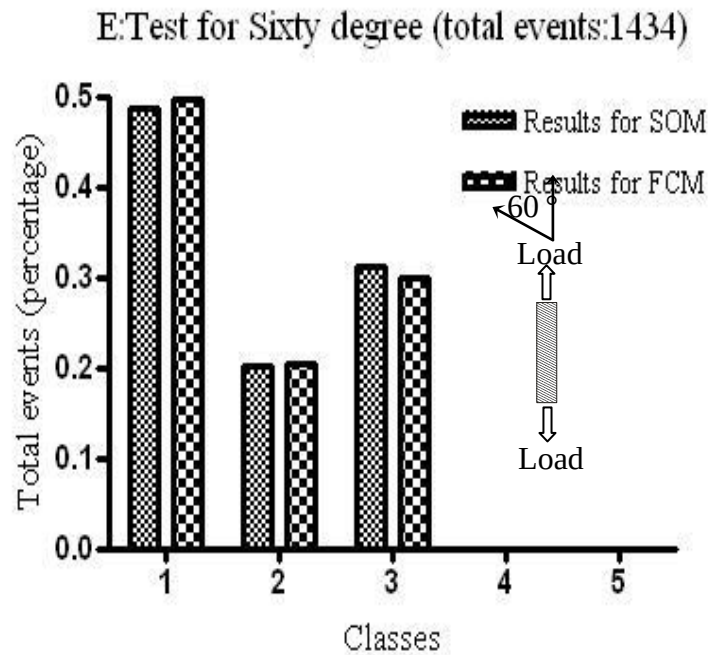


Figure 24: Normalized events per class for 60° Glass/Polyester.

9.5.6 75° Fiber orientation

Figure 25 shows the SOM output neurons for 75° Glass/Polyester along with the circled clusters obtained from these neurons. Figure 26 plots three cluster centers which were represented by cluster centers after FCM converged. As described previously Class 3 has appeared now and the number of events associated with it increases as the fiber orientation rotates away from the direction parallel to the load for the 60° , 75° and 90° tests. The number in Class 1 appears to increase in the first few configurations (between 0° and 60°) after this fiber configuration as we will see in the next orientation.

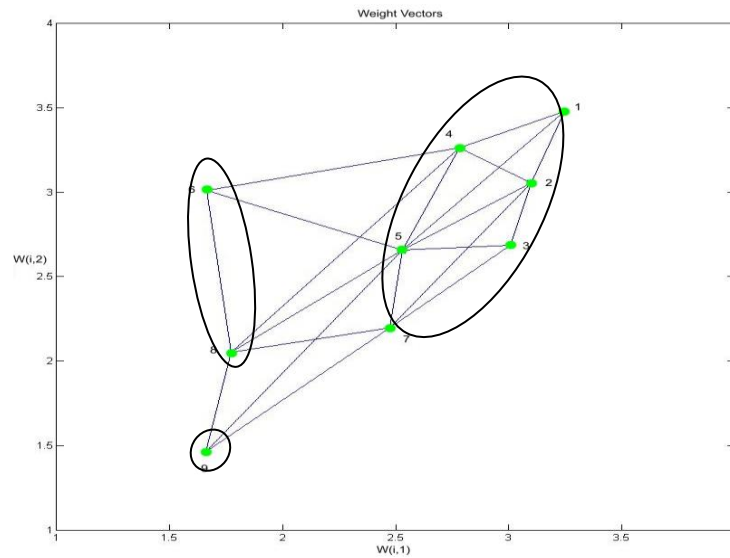


Figure 25: SOM output for 75 °Glass/Polyester.
F:Test for Seventyfive degree (total events:534)

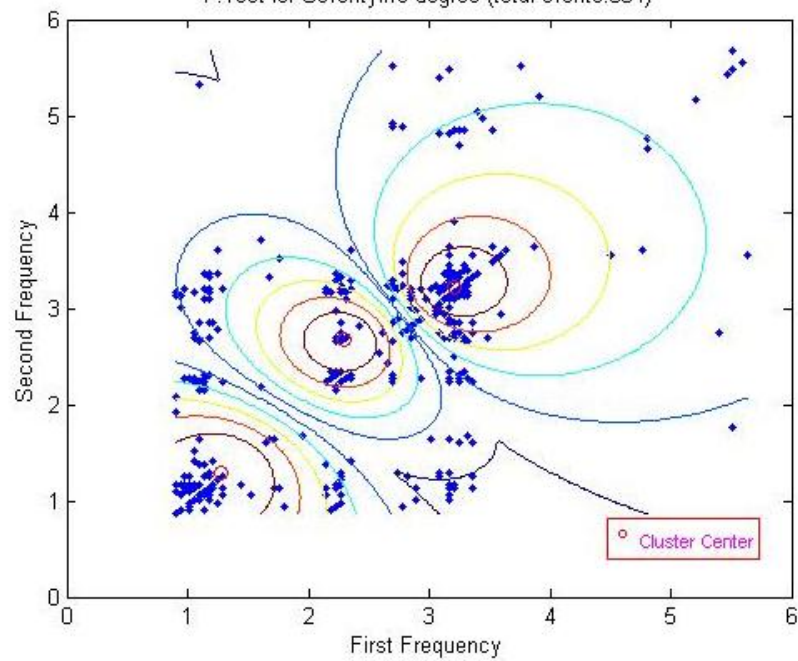


Figure 26: FCM output for 75 °Glass/Polyester.

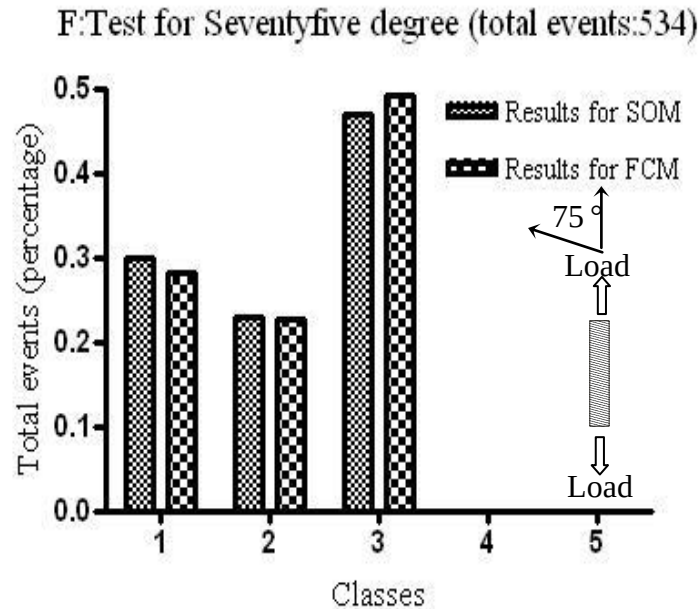


Figure 27: Normalized events per class for 75 °Glass/polyester.

Figure 27 shows the class normalized population for the 75 °Glass/Polyester. Class 3 continues presenting an increase in the number of events compared to previous fiber configurations. Class 4 and Class 5 remain empty and the number in Class 1 continues to decrease as the fibers keep rotating away from the direction parallel to the load.

9.5.7 90 °Fiber orientation

Figure 28 shows the SOM output neurons for the 90 ° fiber orientated Glass/Polyester. Class 1 is formed by neuron 1, Class 2 is populated by neurons 2, 4 and 5, and finally Class 3 embraces neurons 3, 6, 7, 8 and 9. FCM also converges to three clusters, after 30 iterations, with all the cluster centers less than 4.5k Hz. This last configuration with the fiber in a direction perpendicular to the direction of the load (90 °) is another special case in this present analysis. This is because in this configuration it is expected that we will obtain an entire collection of events with characteristics different to those found in the test where

the fiber are parallel (0°) to the direction of the load, as shown in Figure 11. The tests with steps in between are expected to present a combination of events that gradually drift from those existing at 0° , at one end of the spectrum to 90° , the other end.

The location of the output neurons as well as the cluster centers and subsequent classification of events across the entire set of tests suggests the correlation of these classes to different events that appear in more or less quantity for each fiber configuration.

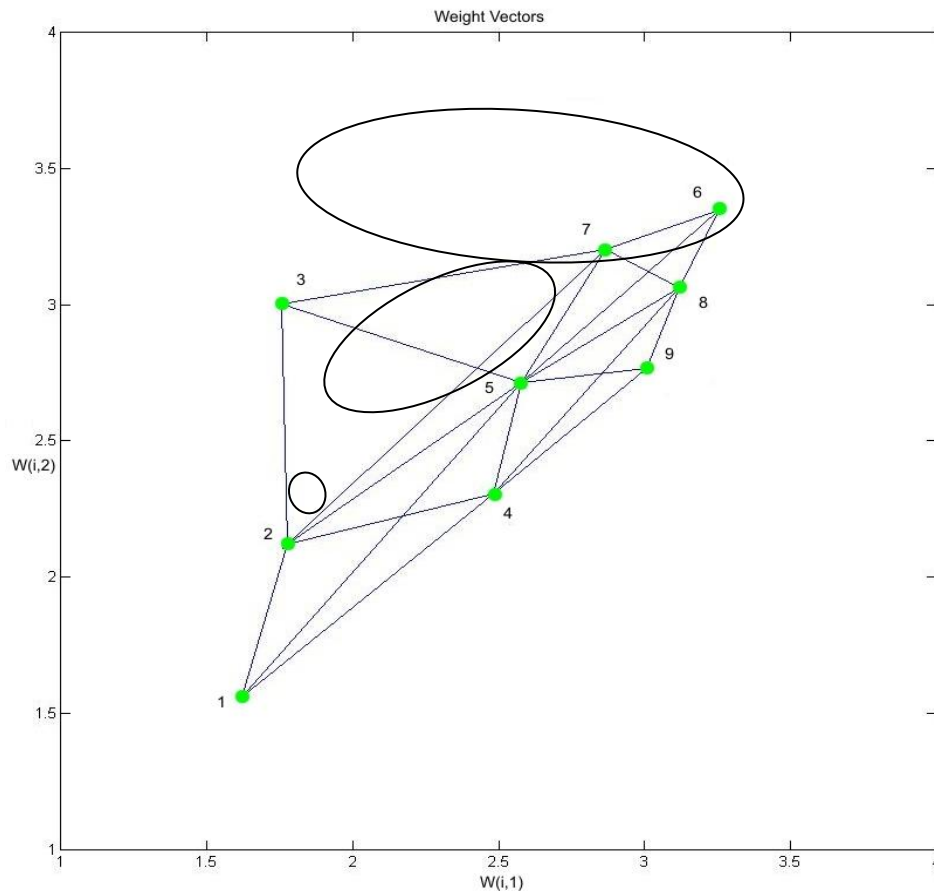


Figure 28: SOM output for 90° Glass/Polyester.

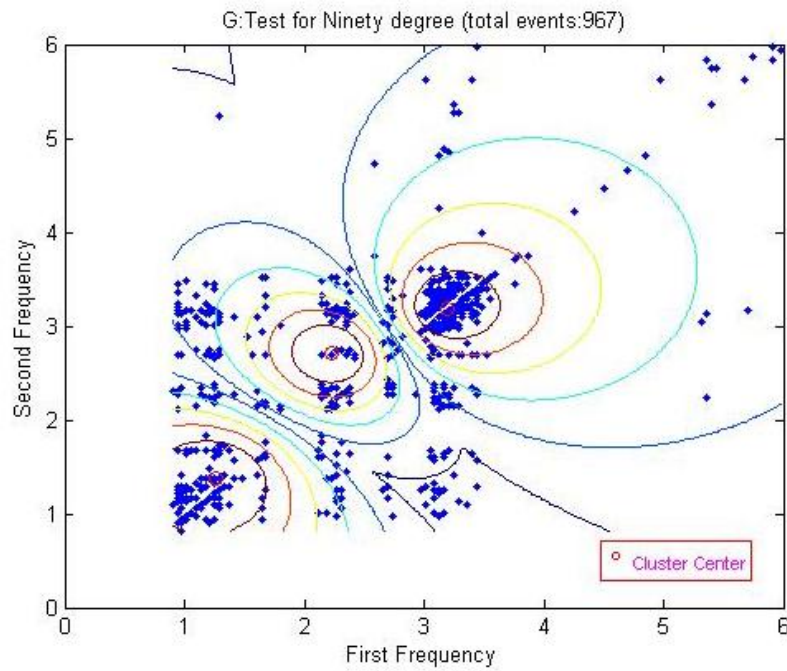


Figure 29: FCM output for 90 °Glass/Polyester.

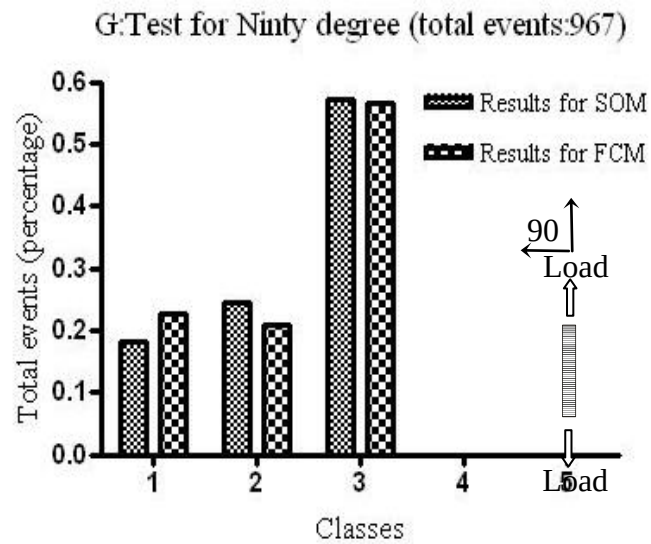


Figure 30: Normalized events per class for 90 °Glass/Polyester.

Bar graph Figure 30 shows the normalized event distribution for the classes found in both algorithms for 90 °Glass/Polyester. Notice that Class 3 presents the largest contribution to the total number of events in the test with nearly 60%, which is consistent for the two algorithms.

Also, it can be seen in Figure 30 that Classes 4 and 5 are not present; Class 2 and Class 1 both contribute around 20% of the total number of events.

The overall transition in both SOM and FCM of the relative contribution per class for each test can be seen in Figure 31. Class 2 shows a progressive decrease in its contribution to the total number of events. Class 3 on the other hand appears from the 60 ° fiber orientation and increases towards the transverse orientation.

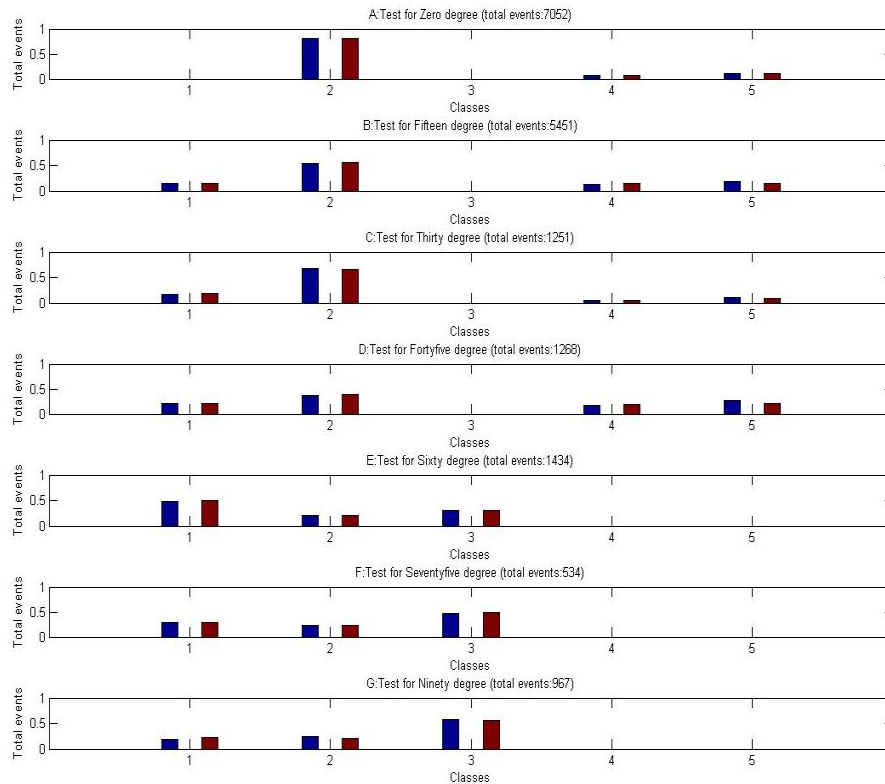


Figure 31: Comparison of results for the 7 tests.

It can also be seen in Figure 31 that Classes 4 and 5 exist only for configurations below 45 °, whereas they are not present for 60 °, 75 ° and 90 °. Class 1 appears in all the tests with a variable contribution to the total events except for the 0 ° configuration where it does not appear at all.

9.6 Discussion and Conclusions

After carrying out an analysis on only the 1st frequencies it was observed that more information was required for a proper description of the failure mechanism phenomenon. This information came from the same FFT previously obtained from all the AE recorded from the tensile tests. After a careful characterization of the recording system in order to assure the quality of the recorded data the clustering process was handled by intelligent system techniques.

This classification process has resulted in a new set of graphs where the events were classified into 5 different classes by their relative closeness in both SOM and FCM. After analysis from a material property perspective, Class 1, Class 2 and Class 3 can be related to matrix cracking, fiber/matrix debonding and fiber/matrix pull out respectively, and Class 4 and 5 can be related to fiber breakage which has been verified by two tests.

In order to verify the correlation between classes and micro-mechanical failure events described in the previous section, a test with 0 °/90 ° Glass/Polyester was carried out. This type of configuration is known to present all the types of events described previously, fiber breakage, fiber/matrix debonding, fiber/matrix pull out and matrix cracking.

Figure 32 shows the SOM output neurons for the 0 °/90 °. Class 1 contains neurons 4, 8 and 9, Class 2 include neurons 5, 6 and 7 corresponds to Class 3, Class 4 is populated by neurons 1 and 2 and finally Class 5 contains the events associated with neuron 3. FCM converges after 46 iterations. Figure 33 clearly shows all five clusters in the map which were all represented by a cluster center. From this figure we can see that the cluster which is around 2.5k Hz dominates all the classes. Each cluster consists of some events which amount to 3015 in total.

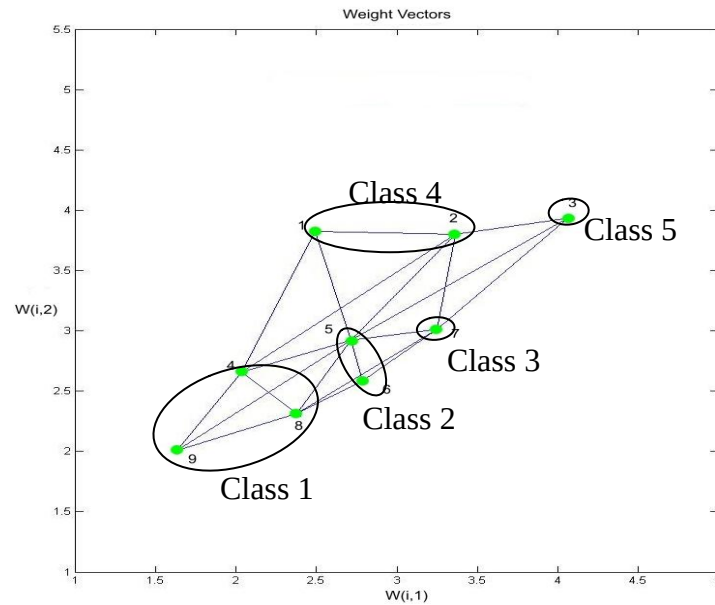


Figure 32: SOM output for 0/90 °Glass/Polyester.

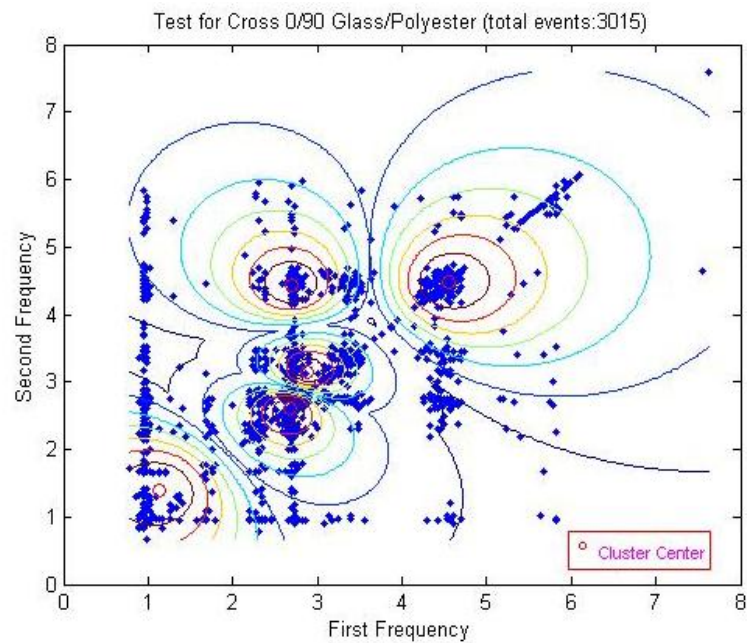


Figure 33: FCM output for 0/90 °Glass/Polyester.

Test for Cross 0/90 Glass/Polyester (total events:3015)

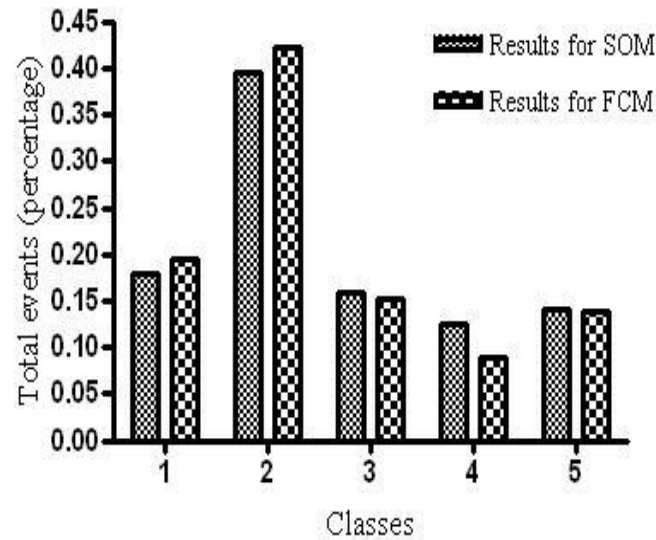


Figure 34: Normalized events per class for 0/90 °Glass/Polyester.

The same process described earlier was followed for the verification testing and as expected, the resulting plot (Figure 34) showed all 5 Classes which are just slightly different for the two algorithms. This complies with the hypothesis proposed in the present document since the 0°/90° configuration being a combination of the two configurations, is expected to present micro-mechanical failure events (and their corresponding AE events) from the two fiber configurations.

Conclusions can be drawn as follows:

- (1) Frequency based analysis from AE technologies can be used to correlate micro-mechanical failure events in composite materials to their corresponding AE signature.
- (2) Two intelligent system engineering techniques methods, SOM and FCM, have been used as a reliable classification and clustering tool for AE events recorded from damage in composite materials.

(3) From the plots obtained from the SOM and FCM classification of AE events, Class 1, Class 2 and Class 3 have been found to be related to matrix cracking, fiber/matrix debonding and fiber/matrix pull out respectively, and Class 4 and 5 have been found to be related to fiber breakage.

9.7 Future Work

Possible areas of future work include:

- (1) Joint research with the manufacturer for an in depth characterization of the response of the recording system.
- (2) Following the methodology explained in the current chapter, there is scope for damage source location for large size components, i.e. plates or actual components.
- (3) Based on the current results there would be scope for analyzing the behavior of sandwich structures under different types of loads, tensile, compression and flexural.

References

- Babuska, R. (1998). Fuzzy Modelling for Control. *Kluwer Academic Publishers*, The Netherlands.
- Bohse J. (2000). Acoustic emission characteristics of microfailure processes in polymer blends and composites. *Compos Sci Technology*, 60: pp 1213-1326.
- Boilot, P. (2003). Novel intelligent data processing techniques for electronic noses: Feature Selection and Neuro-Fuzzy Knowledge Base, Engineering Department,. PhD in Engineering dissertation at Warwick University.
- de Groot PJ et al. (1996). Real-time frequency determination of acoustic emission for different fracture mechanisms in carbon/epoxy composite. *Composite Science and Technology*, 55: p.

- 405-412.
- Eduard Llobet, Evor.L.Hines., Julian W Gardner, and Stefano Franco. (1999). Nondestructive banana ripeness determination using a neural network-based electronic nose. *Measurement Science and Technology*, 10: pp 538-548.
- Giordano M et al. (1999). Analysis of acoustic emission signals resulting from fiber breakage in single fiber composites. *Polym Composite*. 20(6): pp 758-770.
- Hughes S et al. (2002). Use of acoustic emission to identify damage modes in glass fiber reinforced polyester. *Compos Sci Technology*. 62: pp 1433-1444.
- Johnson, M. (2002). Waveform based clustering and classification of AE transients in composite laminates using principal component analysis. *NDT & Int*
- Kohonen T. (1990). Self-organized network. *Proceeding of IEEE*, pp 59-69.
- Lampinen, T., Koivisto, Hannu, and Honkanen, Tapani. (2005). *Profiling Network Applications with Fuzzy C-means and Self-Organizing Maps. Classification and Clustering for Knowledge Discovery*. Vol. Volume 4. 15-27.
- N Godin et al. (2005). Integration of the Kohonen's self-organizing map and k-means algorithm for the segmentation of the AE data collected during tensile tests on cross-ply composites. *NDT&E International*, 38: pp 299-309.
- Nata Ativitavas, T.P., Timothy and J. Fowler. (2006). Identification of Fiber-reinforced Plastic Failure Mechanisms from Acoustic Emission Data using Neural Networks. *Journal of Composite Materials*, 40(3).
- N Godin, S.H., R. Gaertner, and L. Salmon. (2003). Clustering of acoustic emission signals collected during tensile tests on unidirectional glass/polyester composite using supervised and unsupervised classifiers. *NDT & Int*.
- Papadakis N. (2004) Relation comparison methodologies of the primary and secondary frequency components of acoustic events obtained from thermoplastic composites laminates under tensile stress. *11th European Conference on Composite Materials* ; May 31- June 3, Rhodes Greece.
- Qi G. (2000). Wavelet-based AE characterization of composite materials. *NDT & Int*, 33: p. 136.
- Ramírez-Jiménez C. et al. (2004). Identification of failure modes in glass/polypropylene composites by means of the primary frequency content of the acoustic emission event. *Composite Science and Technology*, 64: pp 1819-1827.
- Ritaban Dutta, Evor.L.Hines., Julian W Gardner, Dociana D Udrea and Pascal Boilot. (2003). Non-detective egg freshness determination: an electronic nose based approach. *Measurement Science and Technology*, 14: pp 190-198.

Section III: Intelligent Systems and Telecommunications

Chapter 10

Optimization of Cellular Radio Network using Genetic Algorithm and Fuzzy Logic

Jie An, Evor L. Hines and Mark S. Leeson

10.1 INTRODUCTION

The design objective of the early mobile radio systems was to achieve a large coverage area by using a single, high powered transmitter with an antenna mounted on a tall tower. Although this approach achieved very good coverage, it was not possible to reuse the same frequency channels throughout the system because of the interference which resulted (Rappaport 2002,1996).

The cellular concept was a major breakthrough in solving the problem of spectral congestion and user capacity. The cellular radio system comprises mobile stations (MS), base transceiver stations (BS) and a mobile switching center. High quality of mobile communication service is achieved by limiting the coverage of each BS to cells in small geographic areas. In cellular radio networks, the service area is divided into numbers of cells. Each MS is in a cell and is served by the BS in this cell. The MS is assigned the available channels during its movement in or between the cells. This concept results in a number of technical problems, principally channel allocation and call blocking.

Since demand for mobile communication is growing rapidly, the efficient use of the limited radio spectrum is becoming ever more important in cellular radio systems. The reuse of available channels in each cell of the cellular radio network is implemented by channel allocation algorithms. Each MS in a cell is assigned a channel via the channel allocation strategy. This problem is formulated as a signal-to-interference ratio (SIR) maximization problem that is implemented using

Genetic algorithms (GAs) (Haupt and Haupt 2004). The aim of the GA is to allocate the available frequency channels in such a way that the average quality of the signals that the mobile station receives is maximized.

Call blocking is defined as the situation in which a new call is denied access the network. This may be the case because there are no available channels or the channel allocation algorithm decided that to allow the new call to access any of the available channels would cause increased interference, which might lead to loss of new call(s) in progress. This problem is addressed by using a Fuzzy Logic (FL) (Lin and Lee 1996) together with a simple channel borrowing technique. The latter makes it possible for a new arriving mobile station to borrow an available channel from one nearby cell if all the channels in the cell the MS arrived at are presently in use. FL is used to make the decision as to which cell the channel will be borrowed from.

10.2 BACKGROUND

10.2.1 Cellular Radio Network

A variety of channel allocation algorithms have been developed to satisfy the frequency reuse constraints (Sarkar and Sivarrajan 2002). These algorithms can be classified as either fixed channel allocation (FCA) or dynamic channel allocation (DCA).

In the fixed strategy, the available channel spectrum is divided into sets of frequencies. A number of channels are allocated to each cell based on the reuse pattern determined by QoS requirements (Chu and Rappaport 1994; Chen and Dow 2002). The distance between the BSs which are assigned the same set of frequencies is referred to as the frequency reuse distance. This distance is chosen to keep the interference within acceptable limits, when the interferers are at least the reuse distance away from each other.

In the dynamic strategy, channels are not allocated to different cells permanently. Instead, each time a call request is made, the serving BS requests a channel from the mobile switching centre (MSC) that has access to system wide channel usage information (Re, Fantacci et al. 1995). This scheme can be classified as centralized or distributed. In the centralized approach (Nie and Haykin 1999), all requests for channels are processed by the MSC. In the distributed dynamic allocation strategy (Jiang, T. Lai et al. 2002), the decision

regarding the channel is taken by the BS which is being served based on information from the surrounding cells.

Fixed allocation, although simple, has no ability to adapt to traffic patterns and therefore cannot enhance the QoS. Compared with FCA, DCA has an obvious advantage in terms of channel efficiency. In the dynamic strategy, all the channels are kept in a central pool which is controlled by the mobile switch. Once a new call requests a channel, the channel in which the lowest interference occurs is selected from the channel pool by the switch to serve the new MS. Moreover, the mobile switch only allocates a given frequency if that frequency is not presently in use in the cell or any other cell which falls within the minimum restricted distance of frequency reuse, to avoid co-channel interference.

In dynamic allocation, a BS request for a channel initiates a selection strategy based on a cost function. The function could be a combination of call blocking probability, interference, and other channel reuse constraints. However, this strategy may lead to low efficiency and high computational complexity. The improvement of this allocation process was a major motivating factor for this work.

Clearly, a practical cellular radio system is not infinite in its capacity to admit new calls. The call blocking probability is the probability that a new call is not admitted to any mobile service because of a lack of network resources (Walrand and Varaiya 1996). Using a dynamic channel allocation scheme, new calls are assigned to one of the available channels in the channel pool with the channel reuse constraints satisfied. If no channel can be allocated without violating any reuse constraint, the call is blocked.

10.2.2 Genetic Algorithms

Genetic algorithms (GAs) are optimization and search techniques based on the principles of genetics and natural selection (Haupt and Haupt 2004). One of the key principles of the GA is that the fittest chromosomes survive via competitions within a population. GAs work by evolving a population of solutions and improving it in each generation through the means of selection, mating, and mutation. An evaluation function specific to the problem under consideration is used to decide which members of the population should survive.

A simple GA is described by Goldberg (Goldberg 1989) and is used here to illustrate the basic components of a GA. Figure 1 shows the basic structural composition of a GA.

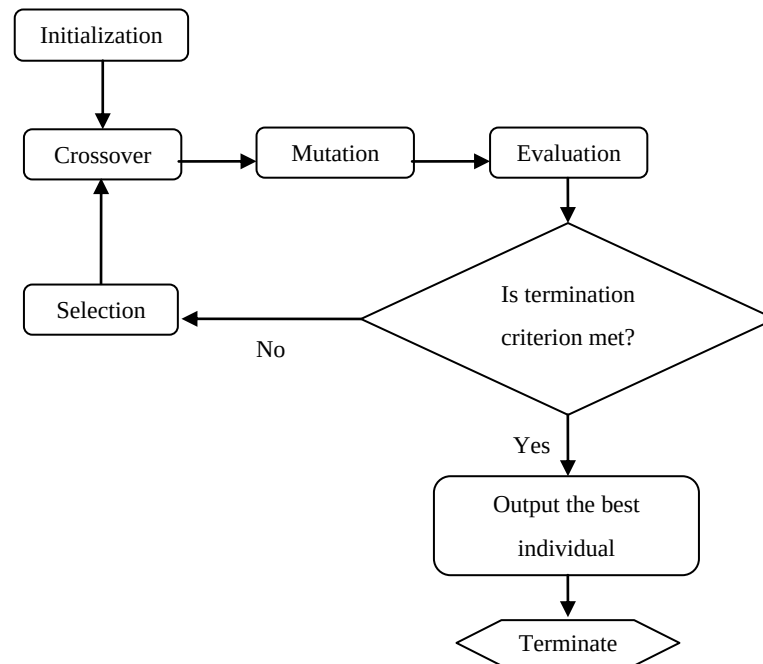


Figure 1: The structural composition of a genetic algorithm

10.2.2.1 Population representation and initialization

GAs operate simultaneously on a number of potential solutions, called a population, consisting of some encodings of the parameter set. In most GAs, a single-level binary string is used to describe the parameters for each solution. Each variable in the parameter set is encoded as a binary string and these are concatenated to form a chromosome. For example, a problem with three variables, x_1 , x_2 and x_3 may be mapped onto the chromosome structure in the following way:

Although binary-coded GAs are most commonly used, alternative encoding strategies, such as integer and real-valued representations (Zalzala and Fleming 1997) are also used. An example of a GA application is the Traveling Salesman

Problem. Here the paths of the traveling salesman are used as the chromosomes in the design of GAs, each integer corresponding to a city during the salesman's journey.

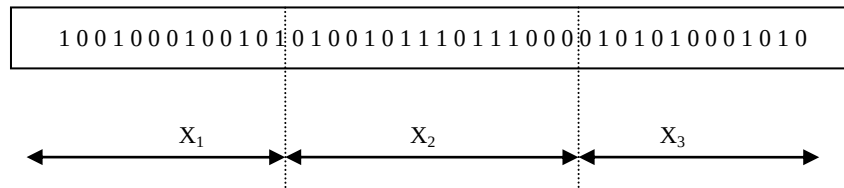


Figure 2: Binary representation of the chromosome structure

Typically, a population is composed of between 30 and 100 individuals (Zalzala and Fleming 1997). The larger the population pool defined, the more the computing space which is required for the GA system to operate in, and the more the computing time which is required for each generation.

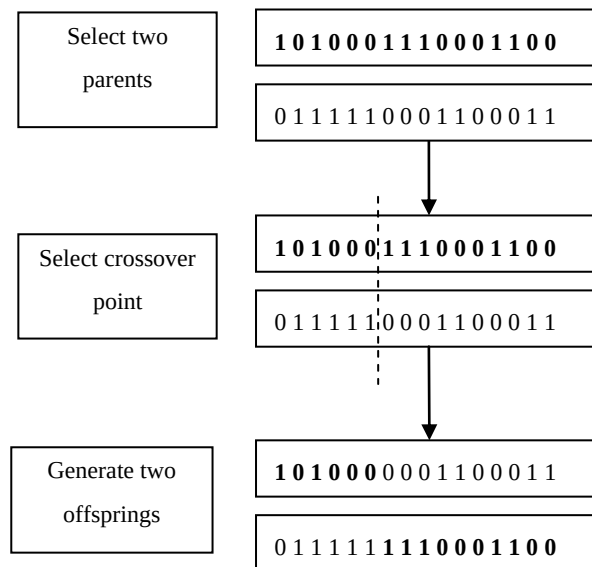


Figure 3: Crossover procedure

10.2.2.2 Crossover

The crossover operation is one of the genetic operators used in producing new individuals using the features of existing chromosomes. This procedure consists of three steps:

- (1) Two individual strings, selected at random from the population, are considered as the two parent chromosomes in the crossover operation.
- (2) The crossover point(s) are randomly selected, usually so that the distribution from which the points are drawn from is uniform.
- (3) Two new individuals are generated by exchanging the substrings that come after the crossover point(s) in both parents.

The mechanism is illustrated in Figure 3.

10.2.2.3 Mutation

In GAs mutation is the operator that causes new individuals to be generated in the population. When mutation is applied to a bit string it sweeps down the list of bits, replacing each by a randomly selected bit if a probability test is passed (Davis 1991). A single point mutation changes a 1 to 0, and vice versa. Increasing the numbers of mutations increases the algorithms' freedom to search outside the current region of the variable space.

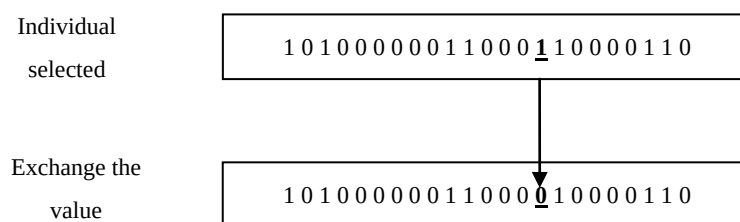


Figure 4: Mutation procedure

Figure 4 is an example of the mutation operation. Firstly, one individual to be mutated is selected from the population. Then the value at a randomly selected

point in the chromosome, called the mutation point, is replaced by another value that is selected from all possible values.

10.2.2.4 Evaluation and fitness function

The evaluation process is driven by the fitness measurement. Each chromosome is assigned a value which is evaluated using the fitness function. A simple GA can be realized by specifying the search space and then identifying the fitness function (Vose 1999). Therefore, this process is the key factor in the design of the GA system. The chromosomes for each generation are selected based on the fitness value.

The fitness function may for example be a mathematical function, an experiment, or a game. The function takes a chromosome as an input and returns a measurement of its performance on the problem we intend to solve. With the initial input, namely, the problem to be solved, encoded solutions to the problem, and an evaluation function of its solutions, we can apply the GA to a population of possible solutions (Davis 1991). In the case of minimization problems (e.g. the traveling salesman problem), the most fit individual will have the lowest numerical value of the associated fitness function.

10.2.2.5 Selection

Selection is the process of determining the numbers of times, or trials, a particular chromosome is chosen for reproduction and, thus, the number of offspring that an individual will produce. The selection of individuals can be viewed as two separate processes (Zalzala and Fleming 1997):

- Determination of the number of trials an individual can expect to receive;
- Conversion of the expected number of trials into a discrete number of offspring.

There are a variety of selection methods in GAs (Haupt and Haupt 2004):

- Pairing from top to bottom. Start at the top of the list and pair the

chromosomes two at a time until the number of selected individuals has reached the selection rate.

- Random pairing. The selected chromosomes are generated randomly.
- Weighted random pairing. The probabilities assigned to the chromosomes in the population pool are inversely proportional to their cost.
- Tournament selection. A subpopulation of the chromosomes is picked at random. The individual from this subpopulation with the highest fitness wins the tournament.

10.2.3 Fuzzy Logic concepts

Fuzzy logic formalizes the use of linguistic terms to represent information in addition to numbers. For example, a length could be described as “short”, “medium” or “long”, where, say, “long” means 80% or more of the maximum length. Fuzzy sets allow many degrees of membership, as opposed to crisp sets allowing only $\in$ or $\notin$.

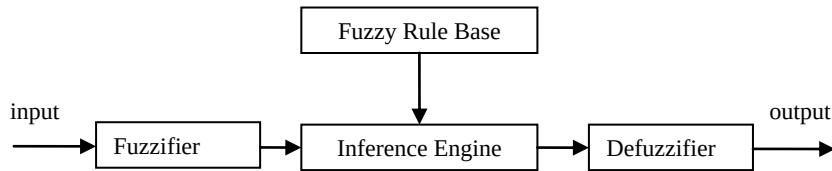


Figure 5: Basic architecture of a FLS

A crisp set is a collection of distinct objects. It is defined in such a way as to dichotomize the elements of a given universe of discourse into two groups: members and nonmembers. A fuzzy set may be viewed as a generalization of the concept of an ordinary set (Yan, M. Ryan et al. 1994). For a given universe of discourse U , a fuzzy set is determined by a membership function μ_F which maps members of U onto a membership range which is usually the interval $[0, 1]$. Namely, $\mu_F : U \rightarrow [0, 1]$.

A fuzzy logic system (FLS) is an inferencing system based on fuzzy set theory, fuzzy logic and fuzzy reasoning. The basic idea behind FLS is to incorporate the “expert experience” of a human operator in the design of the controller in controlling a process whose input-output relationship is described by a collection

of fuzzy rules (e.g., IF-THEN rules) involving linguistic variables rather than using a conventional dynamic model (Lin and Lee 1996).

Some of the popular FLS configurations include Mamdani's fuzzy system, Sugeno's fuzzy system and Tsukamoto's fuzzy system (Jang, C.T. Sun et al. 1997). The components of the FLS proposed by Mamdani are fuzzifier, fuzzy rule base, fuzzy inference engine, and defuzzifier as shown in Figure 5.

Let us consider each in turn:

10.2.3.1 Fuzzifier

The fuzzifier maps the elements from an input crisp set into a fuzzy set. Since the FLS is based on fuzzy set theory, this process is carried out by using a membership function for each fuzzy set converted from the input crisp set.

10.2.3.2 Fuzzy rule base

Fuzzy system rules are characterized by a collection of fuzzy IF-THEN rules. This collection of fuzzy rules characterizes the system's simple input-output relationship. The general form of the rules in the case of a multi-input-single-output system (MISO) is:

$$R^{(l)} : \text{IF } x_1 \text{ is } F_1 \text{ and } \dots \text{ and } x_n \text{ is } F_n, \text{ THEN } y \text{ is } G$$

where F and G are fuzzy sets, and where x and y are the linguistic values and the control variables, respectively. The variables $x_1 \dots x_n$ are the input fuzzy variables and y is the output variable. F and G are corresponding fuzzy sets, which could for example be "low" and "high". Here, l ranges from 1 to n with n representing the total number of rules.

10.2.3.3 Inference engine

The inference engine is used to model human decision making within the conceptual framework of fuzzy logic and approximate reasoning. This mechanism has two basic tasks (Passino and Yurkovich 1998):

- Determining which rules are relevant to the inputs.

- Drawing the conclusion using the current inputs and the information from the fuzzy rule base.

There are various fuzzy inference methods which may be utilized in the FLS, the most commonly used are the following (Yan, M. Ryan et al. 1994):

- The point-valued MAX-MIN fuzzy inference method.
- The point-valued MAX-DOT fuzzy inference method.

For example, we have a fuzzy rule base with two rules:

IF x is A_1 and y is B_1 THEN z is C_1

IF x is A_2 and y is B_2 THEN z is C_2

For two inputs x and y ,

$$a_1 = \mu_{A_1}(x) \wedge \mu_{B_1}(y)$$

$$a_2 = \mu_{A_2}(x) \wedge \mu_{B_2}(y)$$

- MAX-MIN fuzzy inference

In MAX-MIN fuzzy inference, Mamdani's minimum operation rule (Lin and Lee 1996) is used for fuzzy implication. Thus, the membership of the inferred consequence C is given by

$$\mu_C(z) = (a_1 \wedge \mu_{C_1}(z)) \vee (a_2 \wedge \mu_{C_2}(z))$$

- MAX-DOT fuzzy inference

In MAX-DOT fuzzy inference, Larsen's product operation rule (Lin and Lee 1996) is used as a fuzzy implication function. Therefore, the membership of the inferred consequence C is given by

$$\mu_C(z) = (a_1 \bullet \mu_{C_1}(z)) \vee (a_2 \bullet \mu_{C_2}(z))$$

10.2.3.4 Defuzzifier

The defuzzifier maps fuzzy sets into a crisp point. In the FLS, two commonly used defuzzification strategies are the mean of maximum and the center of area (Yan, M. Ryan et al. 1994):

- Mean of maximum method

This strategy generates a control action which represents the mean value of all local control actions whose membership functions reach the maximum.

- Center of area method

This strategy generates the center of gravity of the possible distribution of a control action.

10.3 PROBLEM FORMULATION

The proposed algorithm is applied to 16 hexagonal cells, as shown in Figure 6. The locations of the mobile stations are chosen randomly from a distribution that is uniform over the corresponding hexagonal cell. The numbers of the mobile stations in each cell may be changed in the simulation to evaluate the different performances of the different system capacities.

There are 12 frequency channels in total, and each of these is allowed to serve any mobile user in the system under the dynamic channel allocation constraints. Each frequency only serves a MS in one cell if the frequency is not presently used in the cell, in order to avoid the interference. The call is blocked if there is no available channel in the cell which the call is assigned to.

In this simulation, only co-channel interference is taken into account in order to simplify system modelling. The QoS is used as the performance criterion for the system and this is determined by RSS (Received Signal Strength), SIR (Signal to Interference Ratio) of a mobile station, average traffic difference, the number of dropped calls during handoff and the number of blocked calls during the arrival of new calls in the system. The simulation model used in this work is a modified version of the models used in (Tripathi 1997).

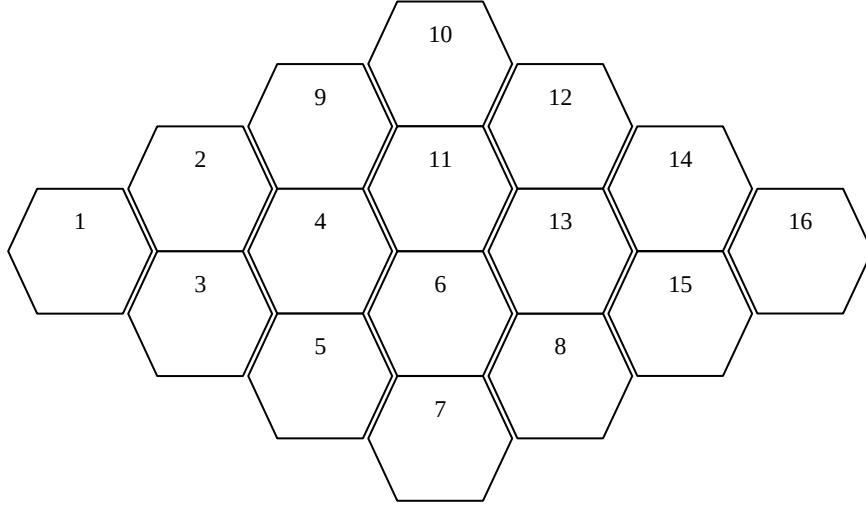


Figure 6: The topological design of the network

10.3.1 Propagation modelling

The RSS is a simple and widely used criterion. The RSS in dBs, P_{Rx} , is obtained from the transmitted power in dBs, P_{Tx} , by subtracting the mean path loss (L_u) and the shadow loss(s) resulting from fluctuations from the mean loss calculated from the Hata-Okumura propagation model (Okumura, Ohmori et al. 1968; Hata 1980) described below. The shadow fading s , is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (Rappaport 2002,1996).

$$P_{Rx} = P_{Tx} - L_u - s \quad (1)$$

where P_{Tx} is the transmission power.

In the Hata-Okumura model, the formula for the median dB path loss in urban areas, L_u , takes the form:

$$L_u = 69.55 + 26.16\log(f_c) - 13.82\log(h_{te}) - c(h_{re}) + \{44.9 - 6.55\log(h_{te})\}\log(d) \quad (2)$$

where f_c is the carrier frequency in MHz, h_{te} is the effective BS antenna height in metres, h_{re} is the effective MS antenna height in metres, d is the transmitter-receiver separation distance in kilometres and $c(h_{re})$ is a correction factor for different sizes of coverage area. The correction term used here is that which is appropriate to a suburban area (Rappaport 2002,1996):

$$c(h_{re}) = \{1.1\log(f_c) - 0.7\}h_{re} - \{1.56\log(f_c) - 0.8\} \quad (3)$$

Assuming in this work that the BS antenna has a height of 30 meters, the height of the MS antenna is 3 meters and that the carrier frequency f_c is assumed to be 900 MHz as an example then, we get the total path loss:

$$L_u = 96 + 35.2\log(d) \quad (4)$$

Interference in a cellular radio network may arise from co-channel interference, adjacent-channel interference and background noise. The first of these is caused by two MSs using the same channel, the second by crosstalk from channels whose frequencies are near each others; and the third by antenna thermal noise. Here, we consider only co-channel interference since the other two may be minimised by appropriate circuit and system design. Considering a set I of active MSs, the SIR of the MS $j \in I$ is then determined from (DeFatta, Lucas et al. 1988):

$$SIR_j = \frac{P_{Rx,j}}{\sum_{i \neq j} P_{Rx,i} + N_0} \quad (5)$$

where $P_{Rx,j}$ is the received power in Watts at j , $P_{Rx,i}$ is the received power in Watts at MS i , which is at the minimum distance for a satisfactory signal to noise ratio, the reuse distance. N_0 is the thermal noise power in Watts.

SIR possesses the advantage as an optimisation parameter that it is common to voice quality, system capacity and the rate of call dropping. The average SIR is the main factor in the fitness function design of the GAs optimization system for the channel allocation problem. This measurement is also considered as an important criterion for the quality of the signal received by the MS.

10.3.2 Mobility modelling

Initially, each base station serves four mobile stations and they are randomly allocated in each cell. The MS moves from one cell to another at a constant speed and their velocities are in the range between 0 and 0.2 km/step. We are assuming the average duration of the existing calls is random in 0-8 simulation steps. If the call is ended, the MS is removed from the network and the channel assigned to this call is released back to the channel pool.

The MS direction and the velocity remain constant throughout the whole journey.

10.3.3 Traffic modelling

Traffic is quantified by the number of calls in a cell. The average differences in the numbers of the calls assigned to each BTS is considered to be another important parameter in this simulation. Uniformity of traffic distribution is captured via the difference in traffic levels over the links between the cells. Since there are $n(n-1)/2$ possible links between n cells, we can define the average traffic difference as (Tripathi 1997)

$$\overline{Tr} = \frac{2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n (Tr_i - Tr_j)}{n(n-1)} \quad (6)$$

where Tr_i and Tr_j represent the number of MSs connected to BTSs i and j respectively.

The interarrival of calls is assumed to be random and its distribution is exponential with mean arrival rate λ per unit time t . The arrival process is assumed to be by Poisson distributed (Trivedi, Ma et al. 2003). The probability that the number of arrivals during a given time period, denoted by $A(t)$ equals n , is given by:

$$\Pr\{A(t) = n\} = e^{-\lambda t} \frac{(\lambda t)^n}{n!} \quad (7)$$

In this simulation, we assume the mean of the Poisson distribution to be 16 and the holding time of the new incoming call is set to be 3 simulation steps. The calls ended or failed are removed at the end of each step and the simulation model continues for several hundred steps. This is based on the idea that each call is finished after a time interval and then releases the channel.

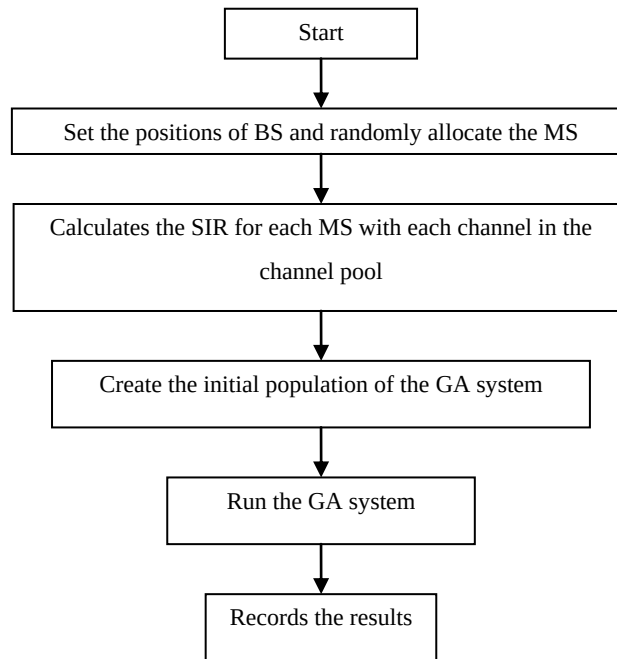


Figure 7: System flowchart of the channel allocation using GA technique

10.4 SYSTEM IMPELMENTATION

10.4.1 Channel allocation via genetic algorithm

There is no simple, closed-form relationship between QoS and its determining elements, making the problem a good candidate for optimisation via for example intelligent systems techniques. After a brief review of GAs, their application to channel allocation to maintain acceptable co-channel interference via SIR maximisation will be presented.

The system flowchart for the channel allocation problem using GA is shown in Figure 7 and the GA system is then introduced.

The channel allocation problem becomes important because, given a number N of available channels satisfying the channel reuse constraint and M mobile stations, there are M^N allocations possible. This is an NP -complete problem, and there are no known efficient techniques to solve such problems. The rapid growth of the number of allocations possible means that exhaustive searching rapidly becomes infeasible. GAs offer a solution that will produce workable allocations in an acceptable time. GAs are used to allocate the frequencies to the existing mobile stations in the cellular systems. By this means, the available channels are reused as much as possible and the interference kept to a tolerable level. This GA system is not terminated until it has completed the specified number of generations.

10.4.1.1 Initialisation

Each individual in the population pool is represented by a vector with a length equal to the number of mobile stations. The i^{th} value in the vector determines the frequency channel that the mobile station i is using. For example, the individual is set by assigning channel 1 to the first mobile station and channel 3 to the last one, as in Figure 8.

1 2 6 3

Figure 8: An example of an individual

Since our system is following the dynamic channel allocation rules (Rappaport 2002,1996), each initial individual is set by the random selection of frequency channels, with allocation permitted if and only if that frequency is not presently in use in that cell. This is illustrated in Figure 9.

Channel number				
1	2		6	3

3	5		5	8

m individuals

Figure 9: GA population pool

10.4.1.2 Crossover

The m individuals resulting from the evaluation and selection in the last generation are sent to the crossover module. By pairing individual n with individual $m-n+1$, the m surviving individuals are separated into pairs as parents of new individuals.

Uniform crossover is the method of choice in this work. Each bit of the offspring is created by copying the corresponding bit from one or the other parent selected at random with equal probability, as shown in Figure 10. Uniform crossover has advantage that ordering of bits is entirely irrelevant because there is no linkage between adjacent bits. Multipoint crossover takes half of the material from each parent in alternation, while uniform crossover decides independently which parent to choose. When the population has largely converged, the exchange between two similar parents leads to a very similar offspring. This is less likely to happen with uniform crossover particularly with small population sizes, and so, gives more robust performance (Vose 1999).

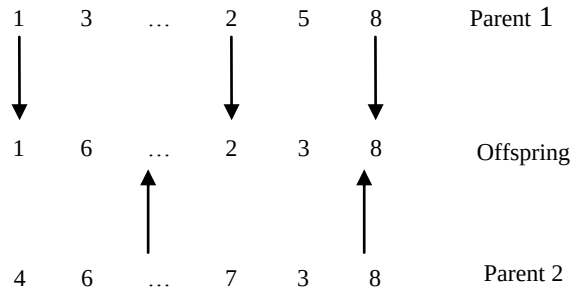


Figure 10: Crossover process

10.4.1.3 Mutation

After reproduction and crossover, one element on one individual changes in the range of possible frequency channels with a mutation probability. Figure 11 below shows an example of the mutation operation. The value at the mutating

point is exchanged with an available channel which is randomly selected from the channel pool.

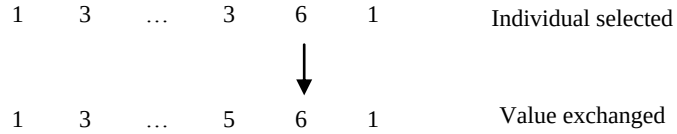


Figure 11: Mutation process

10.4.1.4 Evaluation

Two characteristics are considered for the evaluation procedure in this work. Firstly, the average SIR of mobile stations must be as high as possible. On the other hand, the minimum SIR requirement for all users must be met. The interference between the mobile stations using the same group of channels will be too high if the average SIR is below the minimum value, since the more mobile stations that reuse the same channel the more interference occurs and hence a smaller value of SIR for the corresponding mobile station. The requirement is set at 9 dB (Tuomaala 2002) in this system.

The fitness function in this case thus has the following form:

$$fitness = SIR_{avg} + P(SIR) + J \quad (8)$$

where SIR_{avg} is the average SIR in dB, $P(SIR)$ is a penalty function which decreases the fitness if the value of SIR is lower than 9dB, and J adds randomness. Here the form of $P(SIR)$ and J used were respectively (Tuomaala 2002):

$$P(SIR) = -5 \times (9 - \min(9, \min(SIR))) \quad (9)$$

$$J = \frac{U}{10000} \quad (10)$$

where U is a random number drawn from a uniform distribution between 0 and 1.

10.4.1.5 Selection

A simple selection scheme is utilised, in which the better half of the individuals is retained and the worse half is eliminated. The selected individuals will contribute the next generation and the process is repeated until the process is completed.

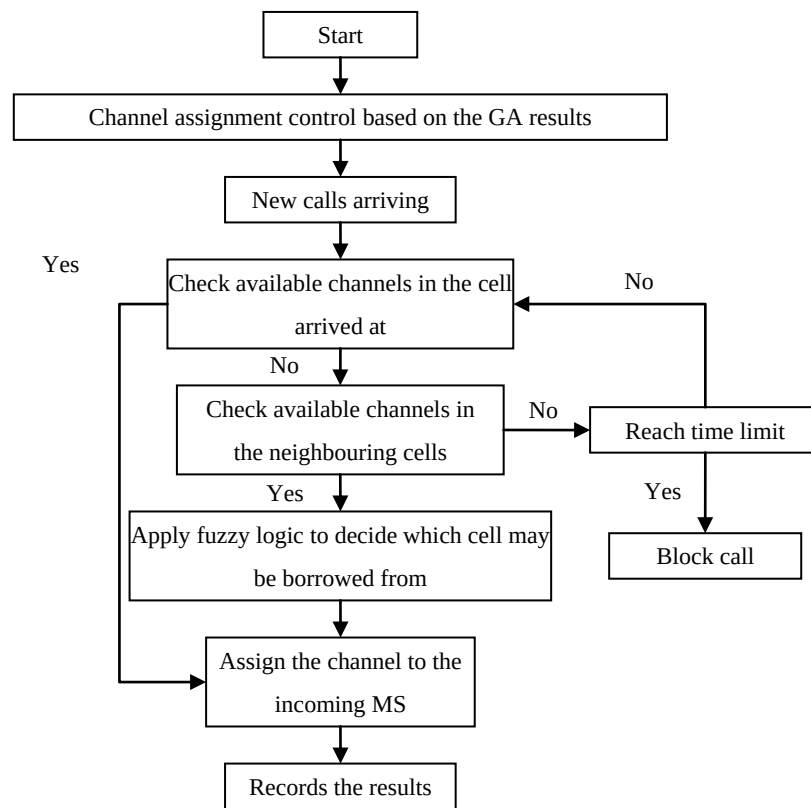


Figure 12: System flowchart of call blocking process using our fuzzy logic and the channel borrowing technique

10.4.1.6 Termination

The number of generations is set in the program as described in the simulation section. The larger the number of generations of the GA system, the more programming recourses (e.g. time, memory etc.) is required. In general, the fitness

value of the fittest individuals selected from the population pool is reaching a constant level during the generations. In order to identify the optimum number of generations for the GA system, a parameter evaluation step of the program needs to be taken; this will be described in the simulation section.

Cell Number (16)	1	1	0	...	0	1	0
	0	1	1	...	1	0	0
	...	...	...	...	...	...	...
	1	1	0	...	0	0	0
	0	0	1	...	1	1	1

Figure 13: Space matrix representation

10.4.2 Call blocking problem using a fuzzy logic and channel borrowing technique

Clearly, a practical cellular radio system is not infinite in its capacity to admit new calls. The call blocking probability is the probability that a new call is not admitted to any mobile service because of a lack of network resources (Walrand and Varaiya 1996). Using a dynamic channel allocation scheme, new calls are assigned to one of the available channels in the channel pool with the channel reuse constraints satisfied. If no channel can be allocated without violating any reuse constraint, the call is blocked.

Figure 12 shows a system flowchart of the call blocking process using our fuzzy logic and the channel borrowing technique.

10.4.2.1 Channel assignment control

The optimum channel allocation solution from the GA stage may be represented by space F , a 16×12 binary matrix containing the frequencies used in each cell of the system. The matrix is updated whenever a new incoming call is assigned an available channel or an existing call ends releasing the channel. An element, f_{jk} , in the matrix is assigned the value 1 when channel j is assigned the mobile station

in cell k and the value 0 when it is not. This gives a typical space matrix of the form shown in Figure 13.

Once a new call is waiting for a frequency channel, simply assign the channel with the highest fitness value if there are available channels which will not be used in the corresponding cell. Based on the principle of dynamic channel allocation, we say the cell is busy if all the frequency channels have been used to serve the MSs in the cell. If a new incoming call is waiting in a busy cell, this MS will be either blocked or it will borrow an available channel from a nearby cell. During this process, the combined fuzzy logic and channel borrowing algorithm is triggered whenever a waiting call in a busy cell is going to assign an available channel from a nearby cell and fuzzy logic is used to decide which cell is chosen.

10.4.2.2 Fuzzy Logic System

The fuzzy logic system used in the implementation is based on the Mamdani FLS, the same system as used in the handoff problem. The combination of the received signal strength RSS_c from the candidate BS, SIR_c of the available channel with the highest fitness value in the candidate cell and the numbers of the MSs connected in the cell, Tr_c , is used as the input to the FLS using which the value of the membership function for each candidate BS, mem_c , is calculated. The BS with the highest value of mem_c will then be chosen as the cell which the waiting MS will assign the channel from. The structure is schematically represented in Figure 14.

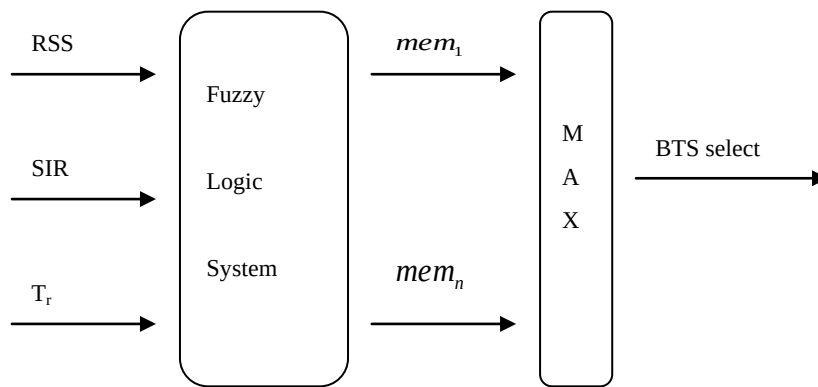


Figure 14: A schematic representation of the fuzzy logic system for call blocking control

10.4.2.3 Membership Function

The universe of discourse for each input variable is classified into three fuzzy sets, “high”, “normal” and “low”. To get sufficient accuracy for output classification, the output variables are divided into seven fuzzy sets, “highest”, “higher”, “high”, “normal”, “low”, “lower”, “lowest”, according to different combinations of input fuzzy sets defined by the rule base. Since the output fuzzy variables are used for comparison, and hence have no real physical meaning, the centre of the membership function for each output fuzzy set can be set as shown in Table 1. The centre of the membership function for each input fuzzy set is presented in Table 2 in terms of the maximum values permitted during the operation of the program.

Table 1: Fuzzy output sets

	<i>Lowest</i>	<i>Lower</i>	<i>Low</i>	<i>Normal</i>	<i>High</i>	<i>Higher</i>	<i>Highest</i>
mem_c	2	4	6	8	10	12	14

Table 2: The centres of input fuzzy membership

	<i>Low</i>	<i>Normal</i>	<i>High</i>
<i>RSS</i>	RSS_{min}	$(RSS_{max}+RSS_{min})/2$	RSS_{max}
<i>SIR</i>	SIR_{min}	$(SIR_{max}+SIR_{min})/2$	SIR_{max}
<i>Traffic</i>	MS_{min}	$(MS_{max}+MS_{min})/2$	MS_{max}

10.4.2.4 Fuzzy Logic Rule Base

In our system, the fuzzy rule base is summarised in Table 3.

10.5 SYSTEM EVALUATION VIA SIMULATION

The system contains 16 base stations, each of them is serving 6 mobile users and there are 12 frequencies in total. Each of the channels is allowed to service any mobile user in the system under the dynamic channel allocation constraints. A typical system model is shown as in figure 15.

Table 3: The fuzzy logic rule base for our call blocking problem

Rule No.	RSS_c	SIR_c	Tr_c	mem_c
1	High	High	High	High
2	High	High	Normal	Higher
3	High	High	Low	Highest
4	High	Normal	High	Normal
5	High	Normal	Normal	High
6	High	Normal	Low	Higher
7	High	Low	High	Low
8	High	Low	Normal	Normal
9	High	Low	Low	High
10	Normal	High	High	Normal
11	Normal	High	Normal	High
12	Normal	High	Low	Higher
13	Normal	Normal	High	Low
14	Normal	Normal	Normal	Normal
15	Normal	Normal	Low	High
16	Normal	Low	High	Lower
17	Normal	Low	Normal	Low
18	Normal	Low	Low	Normal
19	Low	High	High	Low
20	Low	High	Normal	Normal
21	Low	High	Low	High
22	Low	Normal	High	Lower
23	Low	Normal	Normal	Low
24	Low	Normal	Low	Normal
25	Low	Low	High	Lowest
26	Low	Low	Normal	Lower
27	Low	Low	Low	Low

10.5.1 Tuning GA parameters

Initially, a random channel is selected from the channel pool of the 12 frequency channels and the channel is only allowed to serve one mobile user in one cell once under the dynamic channel constraints. 12 individuals are randomly selected to form the population pool and each is represented by a vector with a length equal to the number of the mobile stations.

10.5.1.1 Effect of Crossover probability

In order to achieve the best performance, the GA system was run with different crossover probability (P_c) in the range 0.2 to 1.0. For this test, the mutation probability is set to 0.2 and generation number is equal to 500; 500 having been found to be sufficiently large for the GA to stabilize. The results showed that a P_c of 0.6 produced the most rapid rise and reaches the highest fitness value after 250 generations. Therefore, the value of 0.6 is chosen to be the optimum crossover probability in this simulation.

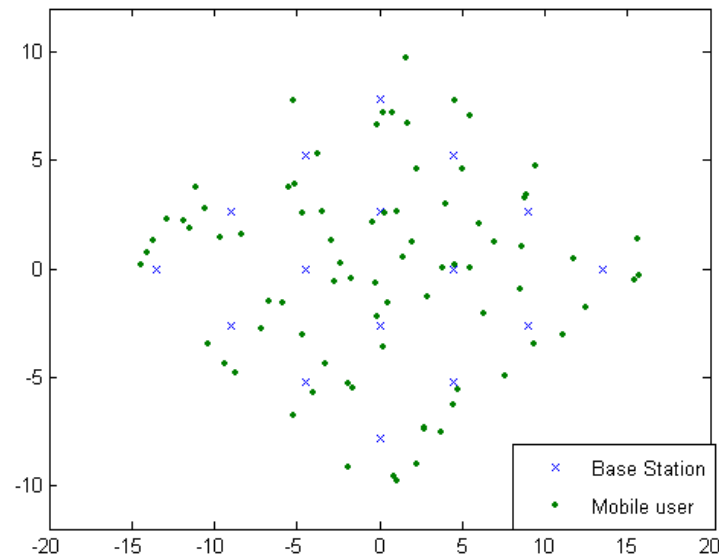


Figure 15: The graphic design of the cellular radio system model

10.5.1.2 Effect of Mutation probability

The effect of mutation probability, P_m is tested in a similar manner to crossover probability. The GA was run with different mutation probability in the range 0.2 to 1.0. The crossover probability was set to 0.6 and the generation number was set to 500. The maximum fitness as a function of generation results showed that the evolution of the fitness values of the GA system significantly depends on the mutation probability. With high mutation probability, the fitness values showed

more variability. This may be caused by the fact that some individuals with the better fitness values are eliminated by the high rate of mutation and the fitness of the population does not stabilize.

With the lower values of mutation rates, the fitness values evolve more gently. From the results, the one with the mutation probability of 0.2 shows the best convergence and reaches the best fitness value in the end.

Therefore, the value 0.2 is chosen as the mutation probability in this simulation.

10.5.1.3 Effect of Number of generations

The previous results showed that with the crossover probability = 0.6 and the mutation probability = 0.2 the GA produces the best performance. With those two values, the fitness values stabilize in less than 300 generations.

In order to use the system for the call blocking and handoff problems, its capacity needs to be adjustable. In other words, changes in the number of MSs and the available frequencies are necessary in future simulations to evaluate the different performances in respect of different network capacities. In order to identify the optimum number of generations, tests are carried out with different numbers of mobile stations in each cell.

The GA tests were carried out with different system capacities and each of them run for 1000 generations. The results were that all the fitness values achieved their maximum values in less than 600 generations. Therefore, 600 generations is assumed to be a reasonable number of generations to use in the call blocking and handoff problem.

10.5.2 Performance evaluation

Here a GA is applied to the problem of optimizing the channel allocation scheme in cellular radio networks. In this scenario, 16 base stations are considered and each cell has a radius of 3 km. This network is optimized using a GA to allocate the available channels to the existing mobile stations in each cell, in order to keep the interference to a tolerable level. Obviously, the larger the number of frequency channels, or the fewer the number of MSs, the better the network performs.

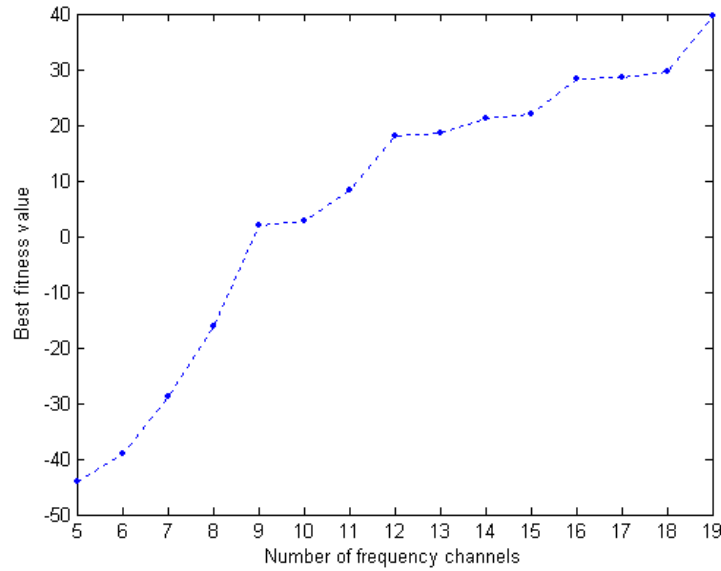


Figure 16: The best fitness value as a function of the number of frequency channels allocated to the network

In Figure 16, the system with 4 MS/cell is allocated using the frequency channels from a range of 5 to 19. In the case where the number of frequency channels is less than 9, the result is a really low fitness value and thus poor performance. The fitness function of this GA system is based on the average SIR value of the network. In order to meet the minimum requirement which was set at 9 dB in the system, the number of frequency channels must be at least 12 channels for this 4×16 network (e.g. 16 cells and 4 mobile stations per cell).

10.5.3 Call blocking problem using a combined fuzzy logic and simple channel borrowing technique

In this section, the simple channel borrowing technique is applied to deal with the call blocking problems for the new call arrivals, and fuzzy logic is used to decide which cell the channel is borrowed from. The performance metrics include the cumulative distribution functions (CDFs) of SIR, RSS, traffic differences and the number of the blocked calls.

Initially, the frequency channels are optimally allocated to the existing mobile users by GA before the new calls arrive. The algorithms are applied to 4, 6, 8, and 10 MS/cell network models as samples to show how the system performs for different capacities of the cellular network.

The new call is only assigned a channel if the frequency is not being used in the cell which the call arrived in; otherwise the call is blocked based on the principle of dynamic channel assignment. In this simulation, a comparison is made between the proposed combined fuzzy logic system and a conventional dynamic channel assignment method.

10.5.4 Performance evaluation

The results for the number of blocked calls and CDF of blocked calls when using our fuzzy logic system was then compared with the numbers of the blocked calls made by the conventional dynamic channel allocation technique over 200 runs of the 4 networks models. The results were that a reduction of over 70% blocked calls was achieved by using the fuzzy logic system.

The average values of SIR, RSS, traffic differences and the numbers of blocked calls achieved by using the two different algorithms in each of the 4, 6, 8, and 10 MS/cell network models are shown in Table 4.

Table 4: The results of the system performance for the call blocking problem

	SIR(dB)		RSS(dB)		Tr(Call No.)		Blocked calls	
4	10.7	12.7	-85.6	-84.7	3.6	3.7	23	162
6	10.3	11.6	-86.0	-85.0	3.6	3.7	52	174
8	10.3	11.4	-85.3	-85.0	3.6	3.7	40	170
10	9.9	11.1	-86.3	-85.0	3.5	3.6	38	172
	Fuzzy	Con.	Fuzzy	Con.	Fuzzy	Con.	Fuzzy	Con.

The table shows that these two algorithms did not affect the values of the average values of the RSS and traffic differences. However there was a slight reduction in the average SIR values in the case of the fuzzy logic system. This was caused by the large numbers of blocked calls made by the conventional algorithms. Since there were much fewer mobile users who were served in the network, the average SIR values were better than that for a high capacity network. The goal here is to try to reduce the numbers of blocked calls as much as possible while keeping

the average SIR at a tolerable level in the network. In this simulation, the tolerable SIR value is set as 9dB as we used in the fitness function of GA system.

Analyses were performed on the results for the average SIR for the different algorithms in 4 network models. The results show that the difference in performance decreased as the network's capacity increases.

10.6 CONCLUSION

The objective of this chapter was to apply intelligent systems to achieve best QoS performance in cellular radio systems. This chapter has addressed this objective by optimizing the following three main problems:

- channel allocation problem: how to allocate the available frequency channels to the existing mobile stations in the network and minimize the interference caused by the reused channels
- call blocking problem: how to reduce the numbers of blocked calls during the arrival of calls

In this chapter, genetic algorithm and fuzzy logic are applied to the problems. Each of the components of the proposed implementation is studied and tested using the Matlab environment.

The main contributions of this chapter are:

- Genetic Algorithm optimized channel allocation
- This problem has been formulated as a SIR maximization problem. The performance of each component of the algorithm have been evaluated with different parameters. With the tested parameter settings, the genetic algorithm optimized channel allocation scheme shows that the algorithm works relatively consistently in optimizing the network; the resulting performance of the network increases with available resources (frequency channels) as expected.
- Fuzzy logic and channel borrowing technique applied to call blocking problem

Call blocking probability is the probability that a new call is not admitted to any mobile service because of a lack of network resources. Under the dynamic channel allocation scheme, a new call is blocked if no channel can be allocated without violating any reuse constraint. The combination of fuzzy logic algorithm and the simple channel borrowing technique is applied to reduce the number of blocked calls during the arrival of new calls. The fuzzy logic system resulted in a reduction of over 70% blocked calls compared to a conventional dynamic channel allocation technique. The penalty being that the average SIR value is less than 2 dB.

10.7 FURTHER STUDIES

10.7.1 Genetic Programming (GP)

GP is an extension of GAs into the area of computer program induction by evolutionary search. According to the research in (Chou and Wang 1998), GP has been successfully applied to the channel assignment problem in wireless networks. A comparison of GP and GA based methods for the channel allocation problem could be conducted as further research work. A brief introduction of GP is presented below as follows.

In GP, the individuals in a population are computer programs instead of binary strings, each of which is extended to determine its performance. The problem itself in GP does not represent a solution, but the solution is realised by running these programs and collecting results.

Compared with GAs, GPs have inherited most of the features of GAs, including crossover, mutation and fitness. A GP starts with an initial population of randomly-generated computer programs, composed of functions and terminals appropriate to the problem domain.

10.7.2 Channel borrowing strategy

In this research, a simple channel borrowing technique is applied to reduce the numbers of blocked calls during the arrival of new calls, and this method performed quite well. This method can also be applied to the handoff problem. An

alternative channel borrowing strategy called BDCL (borrowing with directional channel locking) is introduced in (Yeung, T.P. Yum et al. 1996).

This strategy includes:

- FCA with BDCL for handoff calls
- BDCL with channel reservation
- FCA with BDCL for handoff calls allows a handoff call to borrow a channel using BDCL strategy if no free nominal channel in the call arrival cell is available. In BDCL with channel reservation, both newly arriving calls and handoff calls can use a borrowed channel. But a fixed number of nominal channels in a cell are reserved for exclusive use of handoff calls. Our simulation work can be further extended with this strategy to reduce the numbers of dropped and blocked calls during both the handoff and newly arriving call process.

REFERENCES

- Chen, J. S. and C. R. Dow (2002). "Fixed channel allocation scheme performance enhancement for cellular mobile system " IEE Proc. Commun. 149(4): 232-236.
- Chou, L. and S. Wang (1998). "Channel assignment using genetic programming in wireless networks." Global Telecommunications Conference 5(1998): 2664-2668.
- Chu, T. and S. S. Rappaport (1994). "Generalized Fixed Channel Assignment in Microcellular Communication Systems." IEEE Trans. Veh. Technol. 43(3): 713-712.
- Davis, L. (1991). Handbook of genetic algorithms, Van Nostrand Reinhold.
- DeFatta, D. J., J. G. Lucas, et al. (1988). Digital signal processing: A system design approach, John Wiley and Sons, LTD.
- Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning, Addison

- Wesley Longman, Inc.
- Hata, M. (1980). "Empirical Formula for Propagation Loss in Land Mobile Radio Services." IEEE Trans. Veh. Technol. 29(3): 317-325.
- Haupt, R. L. and S. E. Haupt (2004). Practice genetic algorithms - Second Edition, John Wiley & Sons, Inc.
- Jang, J. S. R., C.T. Sun, et al. (1997). Neuro-fuzzy and soft computing : a computational approach to learning and machine intelligence, Prentice-Hall, Inc.
- Jiang, J., T. Lai, et al. (2002). "On Distributed Dynamic Channel Allocation in Mobile Cellular Networks." IEEE Trans. Parallel and Distributed Systems 13(10): 1024-1037.
- Lin, C. T. and C. S. G. Lee (1996). Neural Fuzzy Systems - A Neural-fuzzy Synergism to Intelligent Systems, Prentice Hall, Inc.
- Nie, J. and S. Haykin (1999). "A Q-Learning-Based Dynamic Channel Assignment Technique for Mobile Communication Systems." IEEE Trans. Veh. Technol. 48(5): 1676-1687.
- Okumura, T., E. Ohmori, et al. (1968). "Field Strength and Its Variability in VHF and UHF Land Mobile Service." Review of the Electrical Communication Laboratory 16(9-10): 825-873.
- Passino, K. M. and S. Yurkovich (1998). Fuzzy Control, Addison Wesley Longman, Inc.
- Rappaport, T. S. (2002,1996). Wireless Communication - Principles and Practice, Prentice Hall, Inc.
- Re, E. D., R. Fantacci, et al. (1995). "Handover and Dynamic Channel Allocation Techniques in Mobile Cellular Networks." IEEE Trans. Veh. Technol. 44(4): 229-237.
- Sarkar, S. and K. N. Sivarrajan (2002). "Channel Assignment Algorithms Satisfying Cochannel and Adjacent Channel Reuse Constraints in Cellular Mobile Networks." IEEE Trans. Veh. Technol. 51(5): 954-967.
- Tripathi, N. D. (1997). Genetic Adaptive Handoff Algorithm Using Fuzzy Logic and Neural Networks. Electrical and Computer Engineering, Virginia Polytechnic Institute and State University.
- Trivedi, K. S., X. M. Ma, et al. (2003). "Performability modelling of wireless communication systems." Int. J. Commun. Syst. 16: 561-577.
- Tuomaala, E. (2002). Applying a Genetic Algorithm in Cellular Radio Network Optimization.
- Vose, M. D. (1999). The Simple Genetic Algorithm, Massachusetts Institute of Technology.
- Walrand, J. and P. Varaiya (1996). "High-performance communication networks." Morgan-Kaufmann, San Francisco.
- Yan, J., M. Ryan, et al. (1994). Using fuzzy logic - Towards intelligent systems, Prentice Hall International (UK) Limited.
- Yeung, K. L., T.P. Yum, et al. (1996). "Prioritized Handoff Strategies Using Channel Borrowing-based Dynamic Channel Assignment." IEEE Indoor and Mobile Radio Communications 3(15-18): 1230-1234.
- Zalzala, A. M. S. and P. J. Fleming (1997). Genetic algorithms in engineering systems, The Institution of Electrical Engineers.

Chapter 11

Design of Survivable Dense-Wavelength-Division-Multiplexing Optical Mesh Transport Networks Using Genetic Algorithms

Yousef S. Kavian, W. Ren, Evor L. Hines, Mark S. Leeson, Habib F. Rashvand and Majid Naderi

11.1 MODERN TRANSPORT NETWORKS

Transport networks provide bulk carriage for client networks including asynchronous transfer mode (ATM) and Internet protocol (IP) networks, and synchronous optical networking (SONET) and synchronous digital hierarchy (SDH), (SONET/SDH) networks. Telephony networks consisting of twisted pair copper wires were the original transport networks. Whereas modern wavelength-routed optical transport networks use fiber optic cables, and nodal devices including optical cross-connect switches (OXC) and optical add-drop multiplexers (OADMs) to connect end users (Doucette, 2005; Murthy and Gurusamy, 2004). An OXC can route an optical signal from an input fiber to an output fiber without performing optoelectronic conversion and an OADM is employed to add or drop some part of traffic between access points.

Modern transport networks employ the Dense-Wavelength-Division-Multiplexing (DWDM) technique as a new transport network technology due to its ability to greatly increase network bandwidth (Kartalopoulos, 2000); using the DWDM technique a single fiber can operate at multiple Tb/s transmission rates. In wavelength-routed DWDM optical transport networks, a variety of services including voice, data and video are multiplexed together and routed from origin to

destination over a common infrastructure called lightpaths or an all-optical communication channel without any optical-electronic-optical conversion and buffering at the intermediate nodes; this is known as wavelength routing (Mukherjee, 2000).

DWDM optical transport networks are being increasingly deployed in next generation wide area, metropolitan, and local area network infrastructures. Figure 1 shows different service layer schemes that utilize DWDM optical networks as backbones. Each sub-network can be operated separately as if it had its own dedicated transmission system, but in reality they can be thought of as various service layers within the underlying transport networks. Service layers can be found in a variety of different stacking arrangements, and each acts as a transport network for the next higher layer (Doucette, 2005). In first model, IP packet data served by an ATM layer (Tanenbaum, 2003) that carries its packet data payloads and this in turn is served by a SONET/SDH layer that is itself served by a lower DWDM layer. In other models, the header payloads, complexity and cost are decreased as a result of the integration of IP over DWDM networks (Ghani et al., 2003; Tomsu and Schmutzer, 2002).

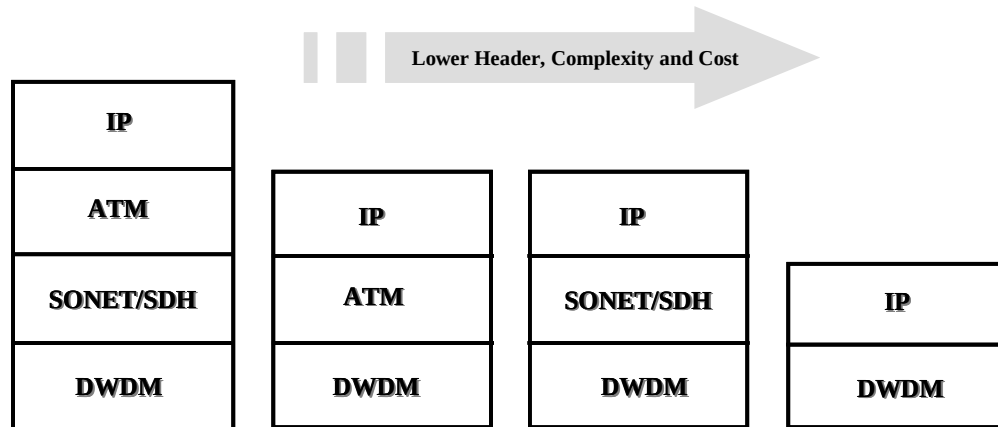


Figure 1: A summary of the different service layer schemes.

11.1.1 Survivability

Nowadays it is necessary to implement optical transport networks with 100% traffic restoration against network component failures. DWDM optical networks are prone to failures and unexpected problems with network components (such as link and node failures) at the optical layer may result in multiple failures at client layers such as the IP layer (Medard and Lumetta, 2001; Mohan and Murthy, 2000). Furthermore, aggregation bandwidths of the order of several terabits-per-second per fiber, using the DWDM technique, causes enormous data and revenue loss in the event of network failure. Therefore maintaining a high level of resilience is an important and crucial issue in the design of fault tolerant DWDM optical networks. This means there is a need for the network to have the ability to restore traffic upon failure (Zhou and Subramaniam, 2000).

Traffic restoration may be provided at the optical layer or the higher client layer; each of which has its own merits. Restoration at the optical layer has several advantages, such as shorter restoration time, efficient resource utilization, and protocol transparency, over that at the client layer. A network with restoration capability requires redundant capacity or spare resources to cover the failures. In case of single link failure, during design of a survivable network, in addition to setting up a working lightpath between each original and destination node pair, P_{od}^w , to satisfy traffic demand, another spare lightpath, P_{od}^s , needs to be established between the node pair, too. The spare lightpath is employed as the means to restore the traffic demand in the event of failure of a working lightpath. For example as shown in Figure 2 the spare lightpath $P_{od}^s : D \rightarrow A \rightarrow B \rightarrow C$ protects working path $P_{od}^w : D \rightarrow E \rightarrow C$ against a single link failure on link E-C.

Integration of survivability into the network core at the optical layer to design fault tolerant networks is an extremely complicated problem, which is known to be NP-hard (Ramamurthy et al., 2003). One of the most important problems in designing survivable optical networks is the routing and wavelength assignment (RWA) problem (Ramaswami and Sivarajan, 1995) in that for each connection request between original-destination node pairs in the demand matrix, a path and appropriate wavelengths are allocated to that request according to an objective function and constraints (Zang et al., 2000; Kennington et al. 2003).

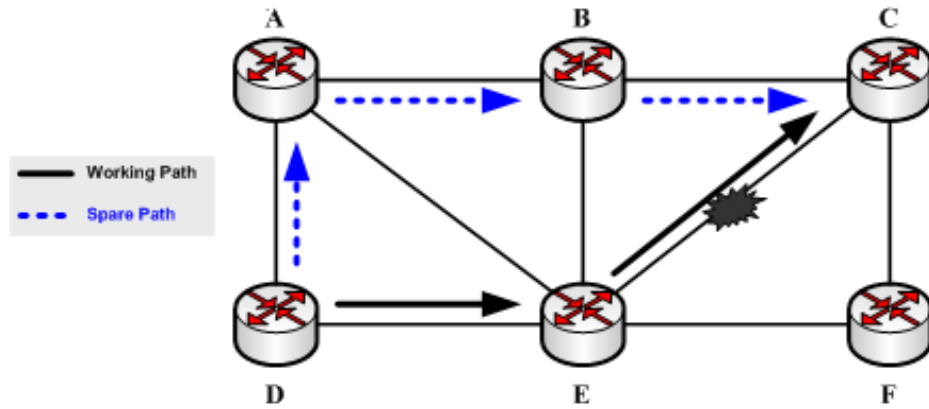


Figure 2: Schematic layout of a DWDM survivable optical mesh network.

11.1.2 Survivability Mechanisms

Two failure recovery mechanisms called *protection* and *restoration* provide the basis of the different schemes in the design of survivable optical mesh networks (Bhandari, 1999). Protection architectures (Maier et al., 2002) establish working and spare lightpaths for arrival requests during network configuration, whereas restoration architectures (Ramamuthy and Mukherjee, 1999) use network state variables to assign spare lightpaths in the event of failure. Whilst the former is static and offers a faster restoration time, the latter is dynamic with greater bandwidth efficiency. The protection architecture is classified into two major categories, *dedicated* protection and *shared* protection. In dedicated protection, backup resources along spare lightpath are specifically dedicated to that lightpath and cannot be utilized by other spare lightpaths; in the shared protection scheme redundant resources can be shared between several spare lightpaths (Ho, 2004). Dedicated protection is faster than shared protection, but shared protection is more efficient for resource utilization.

Both protection and restoration architectures can be further divided into *path* and *link* survivability schemes. In the path survivability scheme, the working and spare

lightpaths must be link disjoint so that in the event of failure of a working lightpath the traffic is rerouted through its spare lightpath. Xin and Rouskas (2004) studied the application of a path protection scheme in the design of large-scale optical networks. In the link survivability scheme, the traffic is rerouted only around the failed link. Whilst the former provides efficient utilization of redundant resources and lower propagation delay, the latter provides lower switching delay (Zhang and Mukherjee, 2004).

Figure 3 shows the various DWDM optical mesh transport network protection and restoration schemes.

The design of fault tolerant optical core networks with protection and restoration schemes has received considerable attention in recent years, with for example the design principles and state of the art being described by Ho and Mouftah (2004). The basic concepts, architectures, models and mechanisms of fault management in survivable optical networks were addressed by Park (2004).

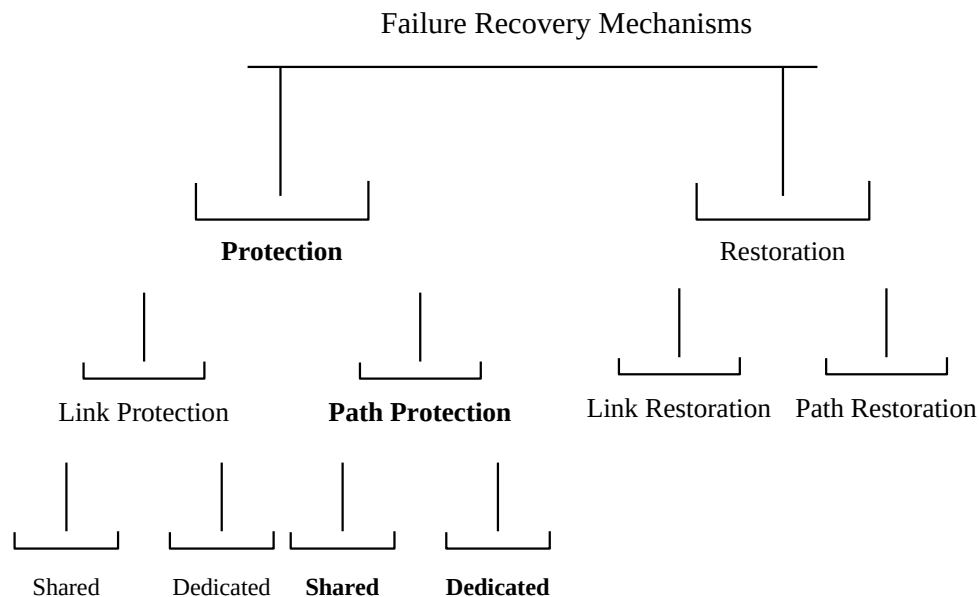


Figure 3: Protection and restoration schemes.

The integer linear programming (ILP) approach has been extensively employed for modeling different survivability schemes in the literature with surveys by Kennington et al. (2006) and Ramamurthy et al. (2003). ILP works well for small problems with linear objective functions and constraints and for small networks where the number of decision variables increases dramatically by increasing the size of networks. However the limitations of ILP models have led to the introduction of diverse heuristic algorithms.

Shenai and Sivalingam (2005) presented hybrid survivability approaches for optical WDM mesh networks. Zhang et al. (2004) presented a heuristic algorithm for the design of survivable WDM networks. Zang et al. (2003) addressed the routing and wavelength assignment (RWA) problem in a network with path protection under duct-layer constraints. Liu et al. (2005) focused on the routing of backup paths and spare capacity allocation in the network to guarantee seamless communication services under failure. Yang et al. (2005) presented resource efficient provisioning solutions in WDM mesh networks, which achieved the objective of maximizing resource sharing. The problem of survivable lightpath provisioning using shared path protection in optical mesh networks employing WDM was investigated by Ou et al. (2004). In addition, Ho (2004) has considered mesh WDM optical networks with partial wavelength conversion capability.

11.2 GENETIC ALGORITHMS

A genetic algorithm (GA) is a stochastic global search method based on natural biological evolution and genetics (Holland, 1975) that may be employed to solve optimization problems that are not well suited for standard optimization algorithms, including those where the objective function is discontinuous or non differentiable. Unlike traditional optimization techniques, GAs works with coding of parameters, rather than the parameters themselves. The decision variables are encoded into finite length strings called *chromosomes*. In contrast to traditional search methods, GAs work with a population of candidate solutions and good solutions are distinguished from bad solutions using an *objective* function that may be a mathematical model, a computer simulation or a logical decision called the *fitness function* (Goldberg, 1989). Figure 4 shows the flowchart of a typical GA's search for solutions.

The GA repeatedly modifies a population of individual solutions and at each step selects individuals from the current population to be parents to produce children for

the next generation. Selection rules are used to choose parents; crossover rules combine two parents to form children and mutation rules apply random changes to parents to form children.

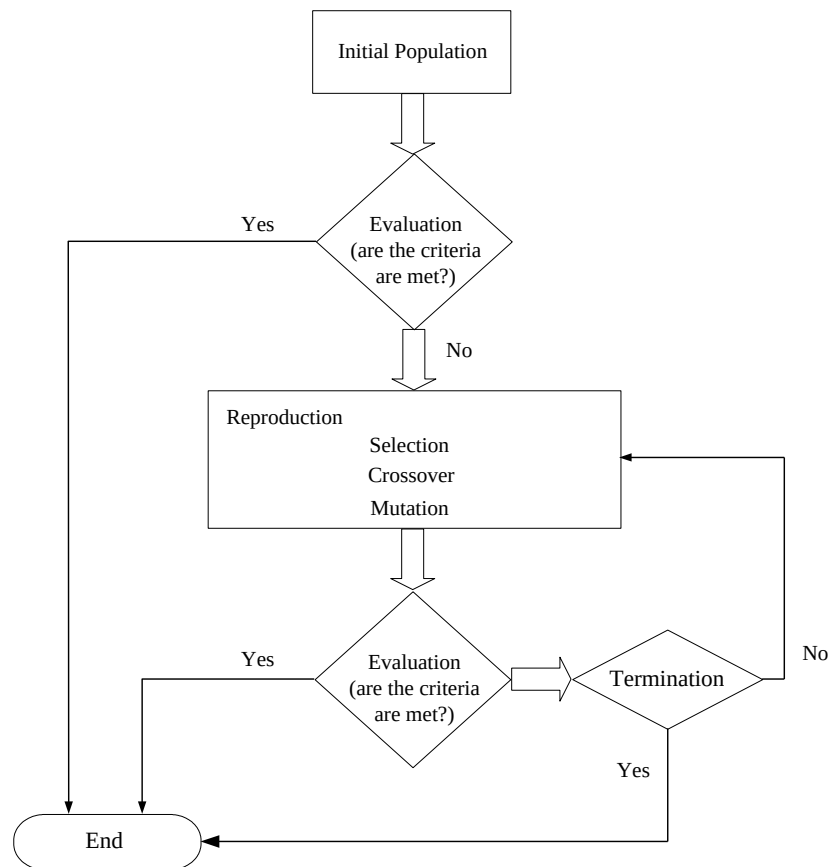


Figure 4: The flowchart of genetic algorithm.

11.3 MATHEMATICAL MODELING

Essentially the RWA problem is to determine the wavelength routed working and spare paths for a given demand matrix, while minimizing the wavelength channels used. It is assumed that the network topology is represented as a directed graph $G(N, L)$, where $N = \{n_1, n_2, \dots, n_N\}$ is the set of nodes and $L = \{l_1, l_2, \dots, l_M\}$ is the set of connecting links in the network. The demand matrix $D[d_{(o,d)}]_{N \times N}$ aggregates demand between origin and destination node pairs (o, d) . It is given in terms of the number of wavelengths, taken for all links from the set $\Psi = \{\lambda_1, \lambda_2, \dots, \lambda_W\}$, required for each connection, $d_{(o,d)}$. The sets of eligible working paths, κ_{od}^w , and spare paths κ_{od}^s , between each node pair before and after of the event of failure are pre-computed using the κ -shortest paths algorithm.

Let us now consider some of the key components:

(1) Objective function

The proportion of working capacity (number of wavelengths required) allocated to working (spare) lightpaths is denoted by f_l^w (f_l^s) for link l . The minimization of the wavelengths utilized by working and spare lightpaths to service a given demand matrix may be written as;

$$f_C = \min \left\{ \sum_{l=1}^M (f_l^w + f_l^s) \right\} \quad (1)$$

$$f_l^w = \sum_{(o,d)} \sum_{P_l^{w,k}} \sum_w w_w^{k,od}, \quad \forall l \in L, \quad \forall D, \quad (2)$$

$$f_l^s = \sum_{(o,d)} \sum_{P_l^{s,k}} \sum_w s_w^{k,od}, \quad \forall l \in L, \quad \forall D \quad (3)$$

Link l is traversed by a set of k th working (spare) paths $P_l^{w,k}$ ($P_l^{s,k}$). The indicator $w_w^{k,od}$ ($s_w^{k,od}$) is set to 1 if the k th working (spare) path between node pair (o, d) uses wavelength w , and to 0 otherwise.

(2) *The link-capacity constraint.* The total number of occupied wavelengths, working and spare, on each link is bounded by the number of wavelengths per link W , which is the product of the number of channels per link, n_{ch} , and the number of wavelengths per channel, ω_{ch} .

$$f_l^w + f_l^s \leq W, \forall l \in L \quad (4)$$

(3) *The satisfaction constraint.* Each link of the working and spare paths that is assigned for a connection request between each node pair (o, d) must satisfy the demand between that node pair.

$$\begin{aligned} \sum_{w=1}^W w_w^{k,od} &= d_{(o,d)}, \forall l \in L, \forall (o,d) \in D \\ \sum_{w=1}^W s_w^{k,od} &= d_{(o,d)}, \forall l \in L, \forall (o,d) \in D \end{aligned} \quad (5)$$

(4) *The wavelength utilization constraint.* Each wavelength can be utilized only by working paths or by spare paths.

$$w_w^{k,od} + s_w^{k,od} \leq 1 \quad (6)$$

(5) *Disjointness.* The working path and the spare path, $(P_{(o,d)}^w, P_{(o,d)}^s)$, between each node pair (o, d) must be link disjoint to cover single link failure. This constraint guarantees that in the event of failure only one of the working path and spare path will fail, and hence the failure will be accommodated by a spare path.

$$P_{(o,d)}^w \cap P_{(o,d)}^s = \phi, \forall (o,d) \in D \quad (7)$$

This constraint is satisfied if and only if $\kappa_{od}^w \cap \kappa_{od}^s = \phi, \forall (o,d) \in D$.

(6) Failure Modeling

It is assumed that only single link failures, Figure 5, occur during any time period of network operation for both DPP and SPP; this being the most realistic failure model for DWDM networks (Shenai and Sivalingam, 2005), where link failures are assumed to be independent of each other and the probability of a single link failure is the same for all links and has a uniform distribution function given by $f(e) = 1/M, \forall l \in L$ (Saradhi et al. 2004). Therefore, the quality of protection (QoP) against the failures is the same for all links in the network.

(7) $\mathcal{K}$ -Shortest Paths Problem

The $\mathcal{K}$ -Shortest path procedure is often needed to design reliable communication networks (Wei and shong, 1999). The $\mathcal{K}$ -Shortest path problem consists of the determination of a set of shortest paths between a given (o,d) node pair of nodes, $\Gamma_{od}(\kappa) = \{P_1, \dots, P_\kappa\}$, when the objective function of the shortest path problem is considered. That is, not only is the shortest path is to be determined, but so do the second shortest, the third shortest, and so on up to the $\mathcal{K}$ -th shortest path.

Lawler (1972) proposed the best known algorithm in computation order $O(\kappa(m + n \log n))$ in an undirected graph with n nodes and m links. While the best known bound is $O(\kappa n(m + n \log n))$ in directed graphs (Kato et al., 1982). The $\mathcal{K}$ -shortest paths algorithm is an extension of the Dijkstra shortest path algorithm (Mittal, and P. Mirchandani, 2004). The algorithm is outlined using pseudo code in Figure 6, where κ_{od} is the final set of n shortest paths found between node pair (o,d) and P is the set of possible paths. For each path p of P , $L(p)$ is the set of links of p in increasing cost order.

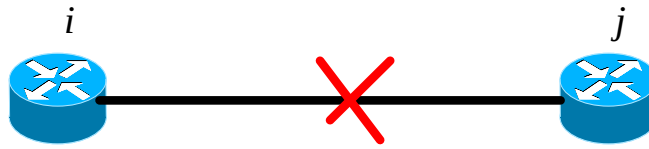


Figure 5: Single link failure.


```

 $\kappa_{od} = \phi$ 
 $P = \phi$ 
For each node  $o \in N$ 
    For each node  $d \in N - \{o\}$ 
        Find the shortest path  $SP$  between  $o$  and  $d$  using Dijkstra Algorithm
         $P = P \cup \{SP\}$ 
         $n = 1$ 
        While ( $n < \kappa$ ) and ( $P \neq \phi$ )
            Take the first path  $p$  of  $P$ 
             $P = P - \{p\}$ 
             $\kappa_{od} = \kappa_{od} \cup \{p\}$ 
            Search  $L(p)$ 
            While ( $L(p) \neq \phi$ ) and ( $n < \kappa$ )
                Take the lowest cost link  $l$  of  $L(p)$ 
                 $L(p) = L(p) - \{l\}$ 
                Remove the link  $l$  of the network and
                Search the new shortest path  $SP'$  between  $o$  and  $d$ 
                (Dijkstra Algorithm)
                If  $SP'$  is found
                     $P = P - \{SP'\}$ 
                     $\kappa_{od} = \kappa_{od} \cup \{SP'\}$ 
                     $n = (n + 1)$ 
                End If
                Reinsert  $l$  in the network
            End While
        End While
    End For
End For
    
```

Figure 6: The pseudo code for the K-Shortest paths algorithm.

For a single link failure, the weight of connecting link between node pair (i, j) is assigned to infinity in the event of failure and to fiber length in kilometers, $\bar{l}_j$, otherwise.

$$\omega(l_j) = \begin{cases} \infty & \text{In the event of failure} \\ \bar{l}_j & \text{Otherwise} \end{cases} \quad (8)$$

11.4 GENETIC ALGORITHM BASED MODEL

The complexity of the survivable network design problem and the ability of GAs to search for solutions for the complex problem in large-scale search space, lead to the provision of a GA based model for this problem. The main reasons for applying a GA to this problem can be summarized as:

(1) Design of survivable DWDM optical transport networks is an NP-hard problem.

(2) The GA is employed in a dedicated manner to model dedicated path protection and shared path protection that are offline survivability approaches, where all the calculations to establish working and spare lightpaths are realized during the process of network set-up; before the network starts to operate.

(3) The GA does not have the limitations of integer linear programming (ILP) approaches including the linearity of constraints and objective functions and it is possible to design survivable DWDM optical transport networks according to nonlinear constraints and objective function including quality of service (QoS) criteria.

(4) GAs have been successfully employed for designing communication networks (Ahn and Ramakrishna, 2002; Arabas and Kozdrowski, 2001; Inagaki and Haseyama, 2005; Chi-Chun and Wei-Hsin, 2000; He et al., 2004; Yang et al. 2003; Pierre and Legault, 1998) and solving NP-hard problems (De Jong et al., 1989).

(5) Furthermore Drenzer and Salhi (2002) compared the performance of descent algorithms, tabu search, simulated annealing and GAs to find the best network configuration for the one-way and two-way network problem, where the objective was minimizing the total travel time of all users by minimizing the total vehicle

miles traveled. It was found that for realistic network sizes the GA outperformed the other algorithms in finding the best the best solution, while taking a longer computation time than other methods (Ukkusuri et al., 2007).

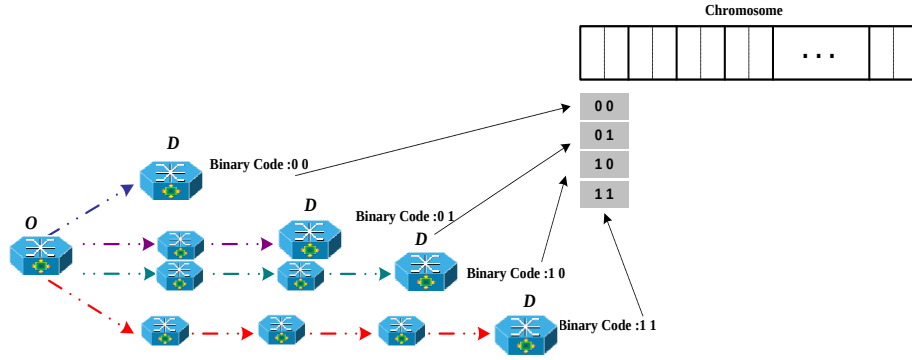


Figure 7: Encoding of lightpaths into variable length chromosomes.

11.4.1 Encoding lightpaths into variable length chromosomes

The key point of connection between GA and an optimization problem is the mapping of the decision variables into chromosomes that affect the accuracy of possible solutions found by the GA. In the problem of survivable DWDM optical mesh network design; the ultimate outcome is a set of working and spare lightpaths between each (o, d) to satisfy a demand request, in the context of the objective function, Eq.(1), and the constraints Eq.(4)-Eq.(7). The chromosome is defined by assigning the integer l to each link with corresponding wavelength set $\{\lambda_{l,1}, \lambda_{l,2}, \dots, \lambda_{l,W}\}$. Then each path of the K -shortest paths between each node pair (o, d) is assigned a binary code and is encoded to a string, $STR_{(o,d)} = \{P_1, \dots, P_K\}$. The chromosome is a binary string formed by concatenation of the strings so that the length of each chromosome is a function of K and the number of requests, Q , in the demand matrix. Figure 7 is a schematic overview of the encoding of lightpaths into variable length chromosomes. As shown in the figure, the set of K - shortest paths between (o, d) constructs a sub-graph of the network graph; i.e. $g_{od} \subset G(N, L)$.

For encoding, the least significant value code is assigned to the shortest path of the sub-graph g_{od} and the most significant value code is assigned to the longest one. Therefore the sub-graph g_{od} is encoded into a variable length binary string formed by concatenation of the binary codes as a chromosome.

- 1- Evaluate the fitness, f_i , of each individual in the population.
- 2- Compute the probability (slot size), p_i , of selecting each member of the population: $p_i = f_i / \sum_{k=1}^n f_k$, where n is the population size.
- 3- Calculate the cumulative probability, q_i , for each individual: $q_i = \sum_{j=1}^i p_j$.
- 4- Generate a uniform random number, $r \in (0,1]$.
- 5- If $r < q_1$ then select the first chromosome, x_1 , else select the individual x_i such that $q_{i-1} \leq r \leq q_i$.
- 6- Repeat steps 4–5 n times to create n candidates in the mating pool.

Figure 8: The steps of implementation of roulette wheel.

11.4.2 Creating the next generation

Crossover: This operation produces new, fitter chromosomes having some parts of the genetic material from both parents. In this work, path crossover involves the exchange of K -short lightpaths between (o,d) node pairs to traverse demand traffic. As illustrated in Figure 9(a), to accommodate the load demanded between (A, E) and (D, E), one wavelength and two lightpath parents, {[A-B-C-E], [D-E]} and {[A-B-E], [D-A-E]}, are needed. After the crossover operation, the two offspring will be {[A-B-E], [D-E]} and {[A-B-C-E], [D-A-E]}. The former is less costly and so is probably selected for addition to the next generation.

Selection: A particular individual is selected for reproduction based on its fitness value. In this work, a virtual roulette wheel (Kwong et al., 2000) is employed to select fitter parent chromosomes. Each chromosome in the population is associated with a sector in this wheel and the area of each sector is proportional to the fitness value of its chromosome, increasing the probability that the fitter chromosomes are selected. Figure 8 indicates the steps in the implementation of a roulette wheel (Goldberg, 1989).

Mutation: Mutation is the random adjustment of one part of the chromosome and often enables the recovery of good genetic material that may be lost through the generations. In this case, a binary mutation operates on a gene (bit) of an element (binary path code) of a chromosome and complements it. The application of mutation to one of the parents from Figure 9(a), {[A-B-C-E], [D-E]} is shown in Figure 9(b). After the mutation operation on each light path, the new offspring (lightpath) is {[A-B-E], [D-A-E]}.

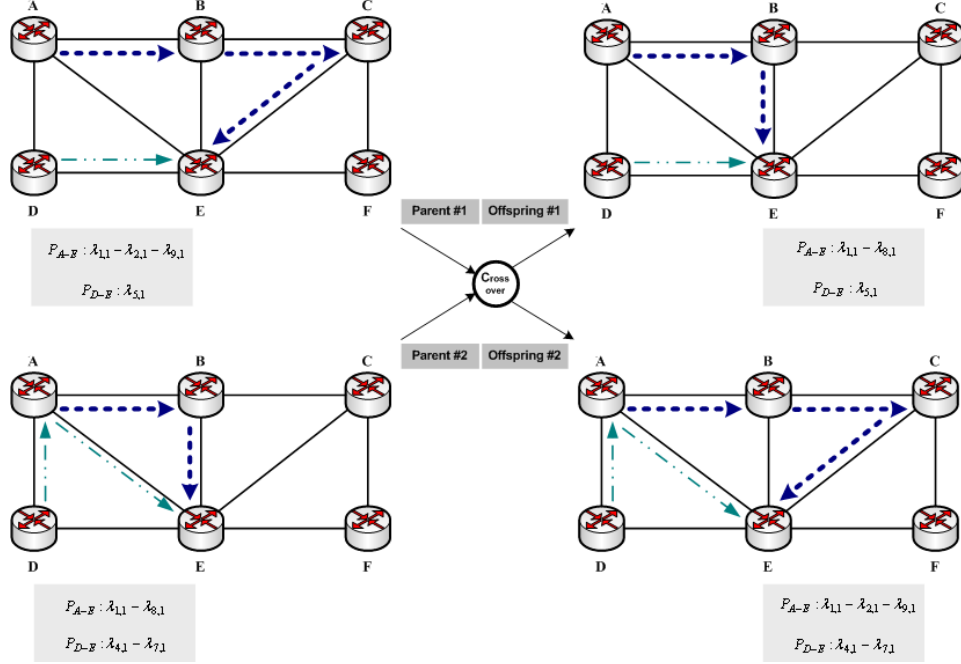


Figure 9: (a) The crossover operation.

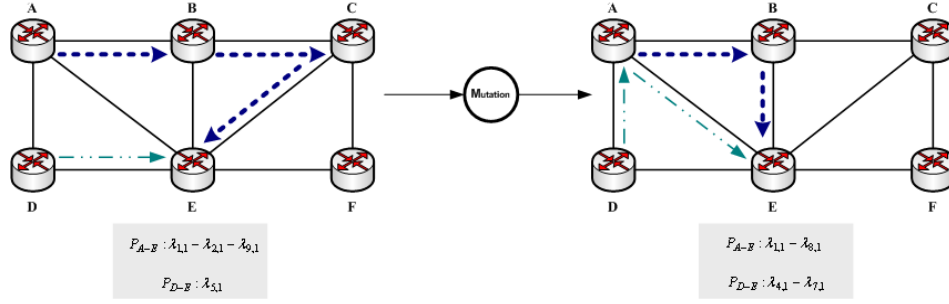


Figure 9: (b) The mutation operation.

11.4.3 Fitness Function

The fitness is a measure of the relative performance of individuals in GAs and here is a weighted sum, with weights w_w and w_s for the working and spare capacity respectively thus

$$fitness(w) = w_w \cdot \sum_{l=1}^M f_l^w + w_s \cdot \sum_{l=1}^M f_l^s \quad (9)$$

11.4.4 The Initial Population

The nature of the initial population created and its size affects the speed and the accuracy of the GA in its convergence to solutions. Large population sizes produce generally better solutions but are very costly in terms of time and memory usage and finding an efficient population size is dependent on the problem considered. The initial path between (o, d) can be generated at random by choosing a path between each (o, d) from the K -shortest paths of sub-graph g_{od} in a random manner for the initial path.

$$P_{od}^i = P_J, \quad 1 \leq J \leq \kappa, \quad \forall (o, d) \in D \quad (10)$$

In other words, the initial path between each (o, d) is the J th path, where J is a random integer. In this work, testing revealed that a better approach was to adapt a heuristic method in which the initial paths were chosen to be the shortest paths from the K -shortest paths for all requests, Q requests, in the demand matrix.

$$P_{od}^i = P_1, \quad \forall (o, d) \in D \quad (11)$$

That is to say that all bits (genes) of the chromosomes in the initial population are '0'.

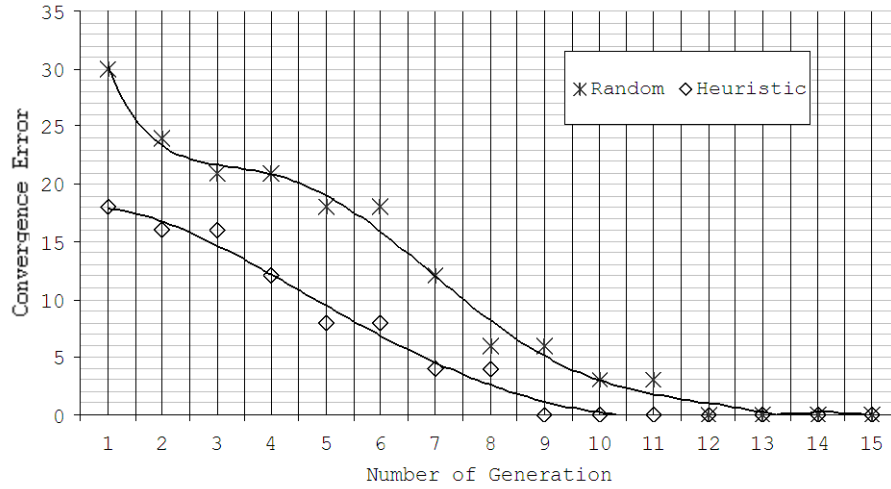


Figure 10: Comparing random initialization and heuristics initialization.

The performance of the random and heuristic initialization schemes has been evaluated by comparing their effects on solving the RWA problem, for the DWDM network shown later in Figure 11. The whole set of results are related to ' $\rho = 3$ ' Erlang load, from each node to all of the other nodes, where the number of

wavelengths per link is $W=10$ and is the same for all links. The behavior of the convergence error was recorded for all solutions. As shown in Figure 10, the initial error for the heuristic approach is less than the error for random initialization, and the heuristic initialization has faster convergence speed than the random ones.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (12)$$

11.4.5 Termination

The stochastic nature of GA searching means that it can be difficult to specify convergence criteria to terminate the evolution cycle. Here, the GA is terminated after a predefined maximum number of generations and/or when the quality of the best lightpaths evaluated according to the objective function is acceptable. If no acceptable solutions are found, the GA is restarted or a fresh search initiated.

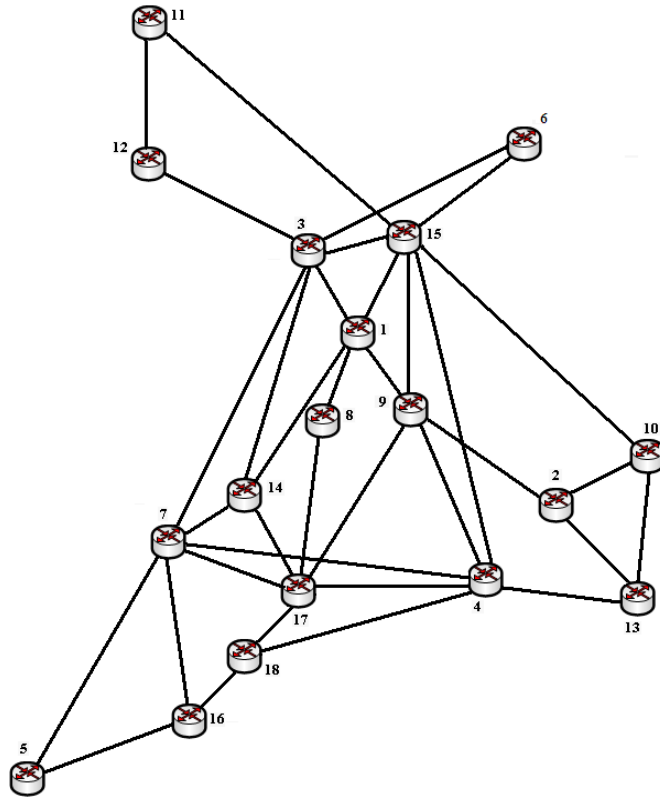


Figure 11: The Pan European network topology.

11.5 EXPERIMENTAL SET-UP AND BENCHMARKING

The Pan European network, shown in Figure 11, (18 nodes, and 35 links) was used as a popular test bench backbone network. The solutions were achieved by considering 40 wavelengths per link, with five channels per link and eight wavelengths per channel. In order to include the network topology in the simulation

set-up, the adjacency matrix, Eq.(12), was utilized. Table 1 was employed to assign integers to network links and to map the topology into a weighted graph using fiber link lengths in kilometers.

All links in the physical layer could carry signals in two opposite directions and all nodes were capable of full wavelength conversion.

11.5.1 Simulation and Processing Set-ups

The simulation set-up was developed using MATLAB 7.1 and run on a parallel processing system 'SUN UNIX SOLAR' Workstation with 24 processors and 24 gigabytes of memory. The block diagram of the simulation set-up consisted of four main blocks; see Figure 12. The first block, called '*Network Topology*', included network topology defined using an adjacency matrix and mapped into a directed graph according to the length of the links. In the second block the demand matrix, including node pairs and volume of traffic between them in terms of wavelengths, was defined and the block is called '*Demand Matrix*'. The third block, called the '*K-Shortest Paths*' block, computed K-shortest path sets for requests in the demand matrix. The fourth block was the '*GA Solver*' that established working and spare lightpaths between each node pair in the demand matrix.

11.5.2 Demand Matrix Modeling

Although the DWDM optical mesh transport networks may actually contain a mixture of services such as IP, ATM and DS-n, all such higher level service requirements can be viewed in aggregate and distilled down to be represented by an overall demand for wavelengths between (o,d) node pairs. Therefore in the demand matrix the incoming lightpath traffic was regulated by a predefined traffic distribution. This model was a more realistic model to design optical transport networks. The demand matrix used was $D = [(1,11,10); (2,7,6); (3,4,7); (6,4,5); (5,17,8); (6,11,9); (17,10,6); (11,4,11); (13,8,13); (18,12,7)]$, where each element of this matrix represents an $(o,d,demand)$.

Table 1: Link encoding and weight assignment.

Node Pair	Link Assignment	Length of link (km)	Capacity (wavelengths)
(1,3)	1	309	40
(1,8)	2	215	40
(1,9)	3	211	40
(1,14)	4	680	40
(1,15)	5	332	40
(2,9)	6	802	40
(2,10)	7	588	40
(2,13)	8	631	40
(3,6)	9	844	40
(3,7)	10	1471	40
(3,12)	11	1323	40
(3,14)	12	569	40
(3,15)	13	415	40
(4,7)	14	1504	40
(4,9)	15	699	40
(4,13)	16	1203	40
(4,15)	17	1155	40
(4,17)	18	359	40
(4,18)	19	912	40
(5,7)	20	1251	40
(5,16)	21	1506	40
(6,15)	22	429	40
(7,14)	23	791	40
(7,16)	24	889	40
(7,17)	25	1312	40
(8,17)	26	766	40
(9,15)	27	500	40
(9,17)	28	959	40
(10,13)	29	530	40
(10,15)	30	1826	40
(11,12)	31	650	40
(11,15)	32	2388	40
(14,17)	33	900	40
(18,16)	34	376	40
(18,17)	35	409	40

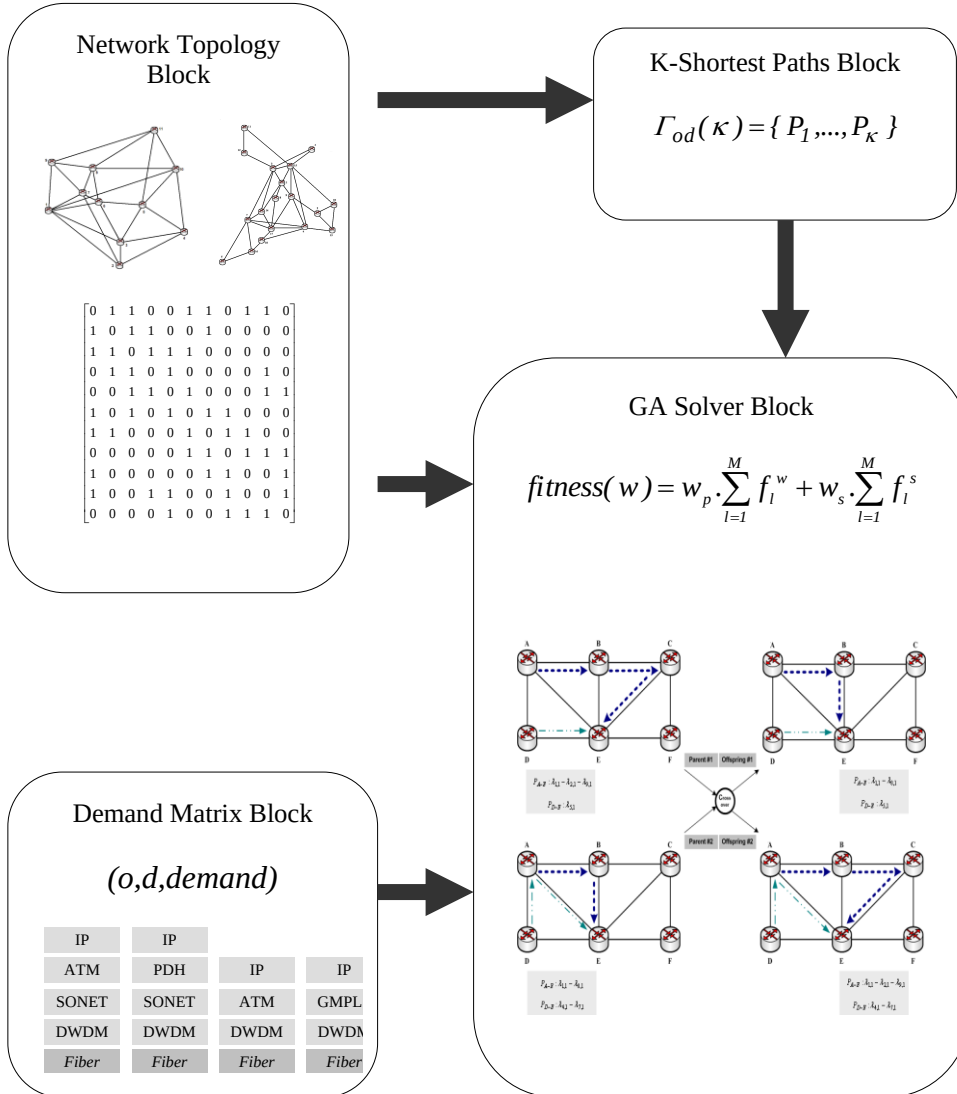


Figure 12: Block diagram of simulation set-up.

11.5.3 Genetic Algorithm Parameters

The setting of the GA parameters is an important issue that affects the accuracy of the solution, time and complexity of searching and memory space consumption. The GA model was run for 100 generations with 50 chromosomes population size, relative accuracy tolerance was $\varepsilon = 10 \exp(-10)$ and the crossover and the mutation probabilities were $\rho_c = 0.90$ and $\rho_m = 0.01$. The value of the GA parameters used is summarized in Table 2.

Table 2: Genetic algorithm parameters.

Population Size	100
Generations	100
Crossover Probability	$\rho_c = 0.90$
Mutation Probability	$\rho_m = 0.01$
Relative Accuracy Tolerance	$\varepsilon = 10 \exp(-10)$

11.6 SIMULATION RESULTS

11.6.1 Working Lightpaths

The RWA simulation results for the Pan European network with demand matrix D are shown in Table 3. Working wavelengths were assigned to paths using the simple but effective First-Fit strategy (Zang et al., 2000), which chooses the available wavelength with the smallest index. The table shows the connection requests between (o,d) node pairs with their volumes in terms of required wavelengths, the RWA solution and the working wavelength utilization for each connection request.

The total number of wavelengths assigned to working paths is 203, giving an average of 20.3 wavelengths per request. The highest number of assigned wavelengths is 39 wavelengths (assigned to working paths between node pair 13 and

8) and the lowest number of assigned wavelengths is 10 wavelengths (assigned to working paths between node pair 6 and 4).

Table 3: RWA Solution for working paths for the Pan European Network.

Node Pair	Demand	Working Path	Working Wavelength	Wavelength Utilization
(1,11)	10	1-15-11	$(\lambda_1 \dots \lambda_{10})$ /all links in the path	20
(2,7)	6	2-9-4-7	$(\lambda_1 \dots \lambda_6)$ /all links in the path	18
(3,4)	7	3-15-4	$(\lambda_1 \dots \lambda_7)$ /all links in the path	14
(6,4)	5	6-15-4	$(\lambda_1 \dots \lambda_5)$ / (6-15), and $(\lambda_8 \dots \lambda_{12})$ / (15-4)	10
(5,17)	8	5-7-17	$(\lambda_1 \dots \lambda_8)$ /all links in the path	16
(6,11)	9	6-15-11	$(\lambda_6 \dots \lambda_{14})$ / (6-15), and $(\lambda_{11} \dots \lambda_{19})$ / (15-11)	18
(17,10)	6	17-9-2-10	$(\lambda_1 \dots \lambda_6)$ /all links in the path	18
(11,4)	11	11-15-4	$(\lambda_1 \dots \lambda_{11})$ / (11-15), and $(\lambda_{13} \dots \lambda_{23})$ / (15-4)	22
(13,8)	13	13-4-17-8	$(\lambda_1 \dots \lambda_{13})$ /all links in the path	39
(18,12)	7	18-4-15-3-12	$(\lambda_1 \dots \lambda_7)$ /all links in the path	28
				203

11.6.2 The Effect of K -Shortest Paths on Working Lightpaths

The effect of the number of shortest paths, K , on the RWA problem for working lightpaths was studied by increasing the number of shortest paths. As shown in Figure 13, by increasing the value of K from 1 to 8 the number of assigned wavelengths decreases from 284 wavelengths to 203 wavelengths for the Pan European network.

The latter value remains stable for $K \geq 4$. Therefore to decrease the complexity, the length of chromosomes, and consequently to decrease the consumed memory and time, the best value for K is 4 and more than this produced minimal solution improvements.

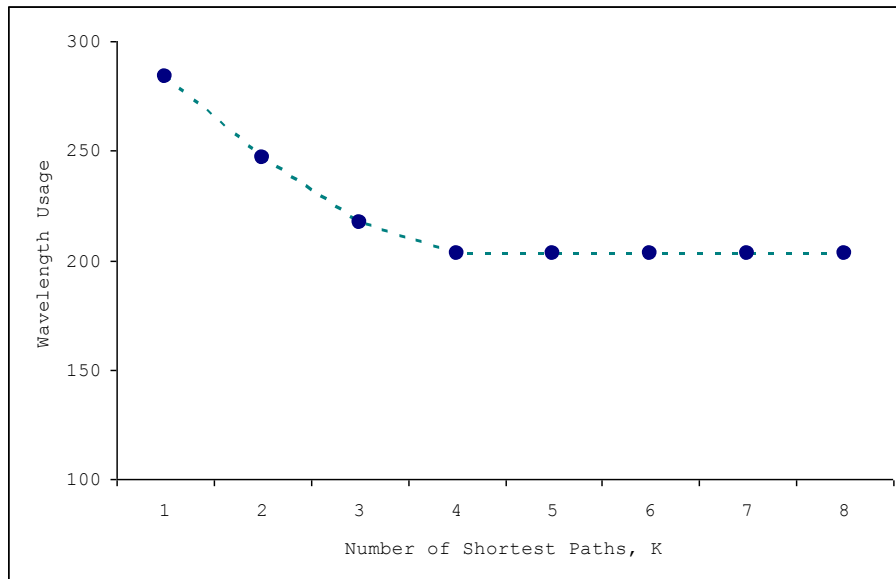


Figure 13: The effect of the number of shortest paths.

Table 4: RWA Solution for Spare Paths by DPP for Pan European Network.

Node Pair	Demand	Spare Path	Spare Wavelength	Wavelength Utilization
(1,11)	10	1-3-12-11	$(\lambda_1 \dots \lambda_{10}) / \{(1-3)-(12-11)\}$, and $(\lambda_8 \dots \lambda_{17}) / (3-12)$	30
(2,7)	6	2-13-4-17-7	$(\lambda_{13} \dots \lambda_{19}) / \{(13-4)-(4-17)\}$, and $(\lambda_1 \dots \lambda_6) / (17-8)$	24
(3,4)	7	3-14-17-4	$(\lambda_1 \dots \lambda_7) / \text{all links in the path}$	21
(6,4)	5	6-3-1-9-4	$(\lambda_1 \dots \lambda_5) / \{(6-3)-(3-1)-(1-9)\}$, and $(\lambda_7 \dots \lambda_{12}) / (9-4)$	20
(5,17)	8	5-16-18-17	$(\lambda_1 \dots \lambda_8) / \text{all links in the path}$	24
(6,11)	9	6-3-12-11	$(\lambda_6 \dots \lambda_{14}) / (6-3)$, $(\lambda_{18} \dots \lambda_{26}) / (3-12)$, and $(\lambda_{11} \dots \lambda_{19}) / (12-11)$	27
(17,10)	6	17-4-13-10	$(\lambda_8 \dots \lambda_{13}) / (17-4)$, and $(\lambda_1 \dots \lambda_6) / \{(4-13)-(13-10)\}$	18
(11,4)	11	11-12-3-7-4	$(\lambda_1 \dots \lambda_{11}) / \text{all links in the path}$	44
(13,8)	13	13-10-15-1-8	$(\lambda_7 \dots \lambda_{19}) / (13-10)$, and $(\lambda_1 \dots \lambda_{13}) / \{(10-15)-(15-1)-(1-8)\}$	52
(18,12)	7	18-17-9-15-11-12	$(\lambda_1 \dots \lambda_7) / \{(18-17)-(9-15)\}$, $(\lambda_7 \dots \lambda_{13}) / (17-9)$ and $(\lambda_{20} \dots \lambda_{26}) / (15-11)$, $(\lambda_{12} \dots \lambda_{18}) / (11-12)$	35
				295

11.6.3 Spare Lightpaths: Dedicated Path Protection (DPP)

The DPP RWA solutions for the spare lightpaths are shown in Table 4. The working and spare lightpaths of all requests are link disjoint, the scheme thus protects against any single link failure because at most one of the two working and spare lightpaths will fail. The number of assigned wavelengths to spare lightpaths is 295 wavelengths, which means 145% redundancy.

11.6.4 Spare lightpaths: Shared Path Protection (SPP)

In the SPP architecture, a link can be shared between spare paths if the working paths that protect the proposed spare paths are link disjoint. Let $S_l^s = \{(P_1^s, P_1^w, d^1), (P_2^s, P_2^w, d^2), \dots, (P_j^s, P_j^w, d^j)\}$ be a set of spare paths, working paths and demand requests that spare paths are common on link l . This link can be shared between spare paths if $P_1^w \cap P_2^w \cap \dots \cap P_j^w = \emptyset$ and the amount of spare resource, r^l , is reduced by the minimum of the common demand requests. The total amount of shared resource is:

$$R = \sum_l \sum_{S_l^s} (j-1).r^l, \forall l \in L \quad (13)$$

The RWA problem was solved using SPP with the same resultant number of working wavelengths as DPP and this is shown in Table 4. Details of the sharing strategy employed by SPP are shown in Table 5. As shown, spare paths that have common links but their working paths are not mutually link disjoint cannot share wavelengths on their common link while other spare paths whose working paths are mutually disjoint can share wavelengths on their common link. The amount of reduced capacity by wavelength sharing is the minimum demand volume traversed on that link.

The total number of shared wavelengths needed was 24, reducing the spare wavelengths by 8.1% from the DPP case, which needed 295, this being the number of spare wavelengths.

Table 5: Sharing Strategy for Spare Paths by SPP for Pan European Network.

Link	Path Pairs	Working Paths Situation	Reduced Capacity
3-12	(1,11)-(6,11)	Joint	0
12-11	(1,11)-(6,11)	Joint	0
17-4	(3,4)-(17,10)	Disjoint	6
6-3	(6,4)-(6,11)	Disjoint	5
13-10	(17,10)-(13,8)	Disjoint	6
11-12	(11,4)-(18,12)	Disjoint	7
			R=24

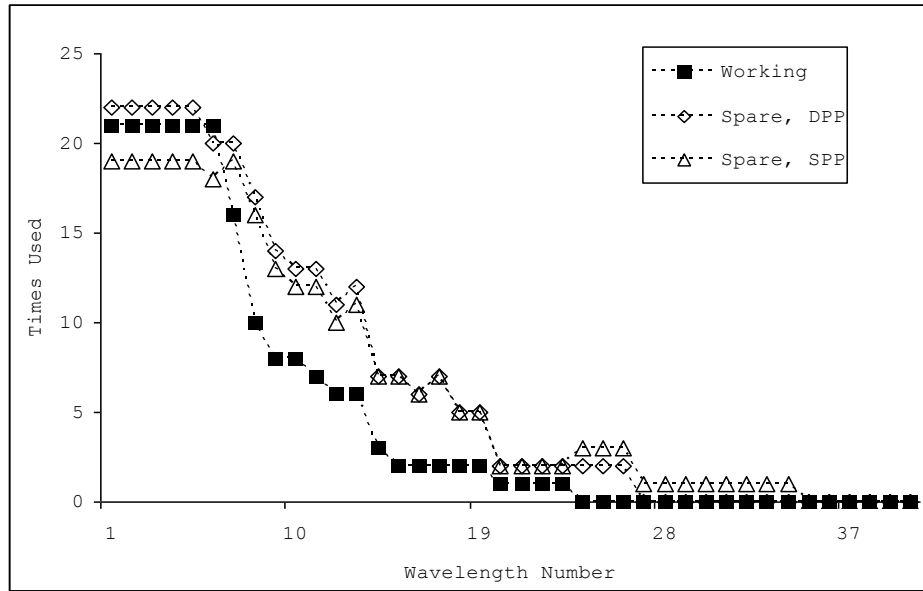


Figure 14: Wavelength distribution for the Pan European network.

11.6.5 Behavior Validation of First-Fit Algorithm

The first-fit algorithm chooses the available wavelength with the smallest index. The behavior of this algorithm has been validated. Figure 14 shows the frequency of the wavelength assignments and it may be observed that the first fit algorithm assigned the lowest numbered of wavelengths as expected, with the working wavelengths making particular use of the first six wavelengths.

11.7 CONCLUSIONS

DWDM optical networks carry large traffic volumes and thus the need to provide a high level of resiliency is an important and critical issue in their deployment. This is particularly true if they are to be employed as backbones for the provision of future IP services with enhanced QoS. This work has addressed the provision of spare capacity via a novel GA model based on variable length chromosomes. The RWA problem has been solved using the GA method and the results of applying it to the Pan European network have been investigated in the context of a single link failure. The optimum number of shortest paths was found to be four, with a greater number producing greatly diminished returns. The assignments using both DPP and SPP were investigated, with the latter producing a saving of 8.1%. The results indicate that GAs are a promising approach to tackle RWA problems in DWDM networks.

References

- Ahn, C. W. and Ramakrishna, R.S. (2002). A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations, *IEEE Transaction on Evolutionary Computation*, 6, No. 6, pp. 566-579.
- Arabas, J. and Kozdrowski, S. (2001). Applying an Evolutionary Algorithm to Telecommunication Network Design, *IEEE Transactions on Evolutionary Computation*, 5, No. 4, pp. 309-323.
- Bhandari, R. (1999). *Survivable Networks: Algorithms for Diverse Routing*, Norwell, MA: Kluwer.
- Chi-Chun, L., and Wei-Hsin, C. (2000). A multiobjective hybrid genetic algorithm for the capacitated multipoint network design problem, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 30, no. 3, pp. 461 -470.
- De Jong, K. A. and Spears, W. M. (1989). Using Genetic Algorithm to Solve NP-Complete Problems, *Proceeding of Third International Conference on Genetic Algorithms (ICGA)*.
- Doucette, J. E. (2005). *Advances on Design and Analysis of Mesh-Restorable Networks*, PhD Thesis, Edmonton, Alberta.
- Drezner, Z. and Salhi, S. (2002). Using hybrid metaheuristics for the one-way and two-way network design problem, *Naval Research Logistics*, 49, pp. 449-463.
- Ghani, N., Dixit, S. and Wang, T. S. (2003). On IP-WDM Integration: A Retrospective, *IEEE Communications Magazine*, pp. 42-45.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley.
- He, L., Botham, C. P., and O'Shea, C.D. (2004). An Evolutionary Design Algorithm for Ring-based SDH Optical Core Networks, *BT Technology Journal*, 22, pp. 135-144, 2004.
- Ho, P. H. (2004). Segment Shared Protection in WDM Networks With Partial Wavelength Conversion, *IEEE Communications Letters*, 8, No. 6, pp. 294-296.
- Ho, P. H. and Mouftah, H. T. (2004). Shared Protection in Mesh WDM Networks, *IEEE Communications Magazine*, pp. 70-76.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.
- Inagaki, J. and Haseyama, M. (2005). A Genetic Algorithm for Routing with an Upper Bound Constraint, *IEICE Transaction of Information and System*, E88-D, No. 3, pp. 679-681.
- Kartalopoulos, S. V. (2000). *Introduction to DWDM Technology: Data in a Rainbow*, IEEE Press, Piscataway, NJ.
- Katoh, K., Ibaraki, T., Mine, H. (1982). An efficient algorithm for K shortest simple paths, *Networks*, 12, pp. 411-427.
- Kennington, J., Olinick, E., Ortynski, A. and Spiride, G. (2003). Wavelength routing and assignment in a survivable WDM mesh network, *Operation Research*, 51, No. 1, pp. 67-79.

- Kennington, J. L., Olinick E. V., and Spiride, G. (2006). Basic mathematical programming models for capacity allocation in mesh-based survivable networks, *International journals of management sciences*, pp. 1-16.
- Kwong, S., Lam, D.W.F., Tang, K.S., Man, K-F. (2000). Optimization of spare capacity in self-healing multicast ATM network using genetic algorithm, *IEEE Transactions on Industrial Electronics*, 47, No. 6, pp. 1334-1343.
- Lawer, E. L. (1972). A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem, *management Science*, 18, pp. 401-405.
- Liu, Y., Tipper, D. and Siripongwutikorn, P. (2005). Approximating Optimal Spare Capacity Allocation by Successive Survivable Routing, *IEEE/ACM Transactions on Networking*, 13, No. 1, pp. 198-211.
- Medard, M. and Lumetta, S. (2001). Architectural issues for robust optical access, *IEEE Communications Magazine*, 39, No.7, pp. 116-122.
- Maier, G., Pattavina, A., Patre, S. D. et al. (2002). Optical network survivability: Protection techniques in WDM layer, *Photonic Network Communication*, 4, No. 3/4, pp. 251-269.
- Mittal, S. and Mirchandani, P. (2004). Implementation of K-Shortest path Dijkstra algorithm used in All-Optical data communication networks, SIE 546 Project Report.
- Mohan, G. and Murthy, C.S.R. (2000). Lightpath restoration in WDM optical networks, *IEEE Network*, 14, No. 6, pp. 24-32.
- Mukherjee, B. (2000). WDM Optical Communication Networks: Progress and Challenges, *IEEE Journal on Selected Areas in Communications*, 18, No.10, pp. 1810-1824.
- Murthy, C. S. R. and Gurusamy, M. (2004). WDM Optical Networks: Concepts, Design, and Algorithms, Prentice Hall.
- Ou, C., Zhang, J., Zang, H., Sahasrabuddhe, L. H., and Mukherjee, B. (2004). New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks, *Journal of Lightwave Technology*, 22, No. 5, pp. 1223-1232.
- Park, J. T. (2004). Resilience in GMPLS path management: model and mechanism, *IEEE Communications Magazine*, 42, No. 7, pp. 128-135.
- Pierre, S. and Legault, G. (1998). A Genetic Algorithm for Designing Distributed Computer Network Topologies, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 28, No. 2, pp. 249-258.
- Ramamurthy, S., Sahasrabuddhe, L. and Mukherjee, B. (2003). Survivable WDM Mesh Networks, *Journal of Lightwave Technology*, 21, No. 4, pp. 870-883.
- Ramamurthy, S. and Mukherjee, B. (1999). Survivable WDM mesh networks: Part I- Protection, *Proceeding of IEEE Infocom Conference*, New York, 2, pp. 744-751.
- Ramamurthy, S. and Mukherjee, B. (1999). Survivable WDM mesh networks: Part II-Restoration, *Proceeding of IEEE ICC Conference*, pp.2023-2030.
- Ramaswami, R. and Sivarajan, K. N. (1995). Routing and Wavelength Assignment in All-optical Networks, *IEEE/ACM Transaction on Networking*, 3, pp. 489-500, 1995.
- Saradhi, C. V., Gurusamy, M. and Zhou, L. (2004). Differentiated QoS for Survivable WDM Optical Networks, *IEEE Communications Magazine*, 42, No. 5, pp. s8-14.
- Shenai, R. and Sivalingam, K. (2005). Hybrid survivability approaches for optical WDM mesh networks, *Journal of Lightwave Technology*, 23, No. 10, pp. 3046-3055.

- Tanenbaum, A. S. (2003). Computer Networks, 4th Edition, Prentice Hall.
- Tomsu, P., Schmutzer, C. (2002). Next Generation Optical Networks: The Convergence of IP Intelligence and Optical Technologies, Prentice Hall, Upper Sadle River, NJ.
- Ukkusuri, S. V., Mathew, T. V., Waller, S. T. (2007). Robust Transportation Network Design Under Demand Uncertainty, *Computer-Aided Civil and Infrastructure Engineering*, 22, pp. 6-18.
- Wei, S. and Shong, C. (1999). A K-Best Paths Algorithm for Highly Reliable Communication Networks, *LEICE Transaction on Communication*, E82-B, No. 4, pp. 586-590.
- Xin, Y. and Rouskas, G. N. (2004). A Study of Path Protection in Large-Scale Optical Networks, *Kluwer Academic Publishers, Photonic Network Communications*, 7, No. 3, pp. 267-278.
- Yang, H. S., Maier, M., Reisslein, M. and Carlyle, W. M. (2003). A Genetic Algorithm-Based Methodology for Optimizing Multi service Convergence in a Metro WDM Network, *Journal of lightwave Technology*, 21, No. 5, pp. 1114 -1133.
- Yang, X., Shen, L. and Ramamurthy, B. (2005). Survivable Lightpath Provisioning in WDM Mesh Networks Under Shared Path Protection and Signal Quality Constraints, *Journal of Lightwave Technology*, 23, No. 4, pp. 1556-1564.
- Zang, H., Jue, J. and Mukherjee, B. (2000). A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks, *SPIE Optical Networks Magazine*, 1, No. 1, pp. 47-60.
- Zang, H., Ou, C. and Mukherjee, B. (2003). Path-Protection Routing and Wavelength Assignment (RWA) in WDM Mesh Networks Under Duct-Layer Constraints, *IEEE/ACM Transactions on Networking*, 11, No. 2, pp. 248-459.
- Zhang, J. and Mukherjee, B. (2004). Review of Fault Management in WDM Mesh Networks: Basic Concepts and Research Challenges, *IEEE Network*, 18, pp. 41-48.
- Zhang, Z. R., Zhong, W. D. and Mukherjee, B. (2004). A heuristic algorithm for design of survivable WDM networks, *IEEE Communication Letters*, 8, pp. 467-469.
- Zhou, D. and Subramaniam, S. (2000). Survivability in Optical Networks, *IEEE Network*, 14, pp. 16-23.

Chapter 12

A Spectral Clustering Approach for Blind Decoding of MIMO Transmissions over Time-Correlated Fading Channels

Steven Van Vaerenbergh and Ignacio Santamaría

12.1 INTRODUCTION

In the last decade, multiple-input multiple-output (MIMO) wireless communication technology has gained considerable attention due to its potential to significantly increase spectral efficiency compared to traditional single-input single-output (SISO) technology. A number of computationally efficient algorithms for reliable symbol detection in time-invariant flat-fading MIMO systems have been developed, such as the V-BLAST architecture (Foschini, Golden et al., 1999).

A direct application of these algorithms in time-varying environments is difficult, however, due to the need for perfect channel state information at the receiver side. Some adaptive equalization algorithms have been proposed to resolve this issue (Choi, Yu, et al., 2005, Rontogiannis, Kekatos et al., 2006, Karami, Shiva, 2006). All of these techniques are supervised equalization algorithms, requiring an initialization phase in which a number of pilot symbol slots are sent.

An alternative approach to equalization can be based on clustering techniques. Since the pioneering work by Chen, Mulgrew et al., (1993) these techniques have been widely applied over the last decade. Clustering algorithms exploit the fact that, for channels excited by signals belonging to a finite alphabet, the noisy observations tend to be clustered around a finite number of

so-called channel states. Once the channel states or cluster centroids have been estimated (typically by applying a clustering technique based on a known training sequence), channel equalization reduces to a classification problem, which can efficiently be solved using a radial basis function network (RBFN) (Chen, Mulgrew et al., 1993, Cid-Sueiro, Artés-Rodríguez, et al., 1994, Lee, Beach et al., 1999), or a support vector machine (Sebald, Bucklew, 2000, Mitchinson, Harrison, 2002, Pérez-Cruz, Navia-Vázquez et al., 2001, Santamaría, González, et al., 2003). From a more general point of view we can say that the problem of separating linear mixtures of discrete-alphabet inputs reduces to the problem of finding and labeling a finite (and known in advance) number of clusters.

Most of the clustering-based equalization techniques proposed so far consider time-invariant SISO systems. Moreover, they need a training sequence which is used to estimate and label the cluster centers in a single step. The main drawback of these techniques is that the number of clusters to be estimated grows exponentially as M^L , where M is the cardinality of the source alphabet (i.e., the constellation size) and L is the channel length. During recent times a lot of work has been directed towards reducing the number of clusters to be estimated. This reduction can be achieved by estimating only those cluster centers that lie close to the decision border (Lee, Beach et al., 1999, Santamaría, González et al., 2003), or by exploiting the symmetries of the channel states inherited from the symmetries of the input constellation (Filho, Dorizzi et al., 2002, Kopsinis, Theodoridis, 2003). Also, some extensions to multiple-input single-output channels have been recently proposed by Diamantaras (2006), Diamantaras, Papadimitriou, (2006).

In this chapter we discuss in detail the case of time-varying multiple-input multiple-output (MIMO) channels. The time-varying nature of the mixing matrix provokes a movement of the cluster centers and, consequently, the clusters adopt non-convex shapes and overlap each other. In this situation, conventional clustering algorithms such as k-means, fuzzy k-means and expectation-maximization (EM) learning, which typically require well-separated clusters, would fail. A promising alternative is the recently proposed spectral clustering technique (Ng, Jordan et al., 2002), which is capable of clustering non-convex datasets. Here, a spectral analysis of point-to-point statistics is performed by calculating the eigenvectors of a similarity (or “affinity”) matrix. This reduces the original hard clustering problem to an easier clustering problem which can be

solved by conventional clustering methods in the eigenspace of the affinity matrix.

Although spectral clustering algorithms provide excellent results for non-convex and intertwined datasets, they do not work for overlapped clusters. Another drawback is the exponential growth of the number of clusters with the constellation size and number of transmit antennas. Therefore, some new ideas must be added to the technique to tackle these problems. We show that by appending the time variable to the vector of observations, it is possible to untangle the clusters corresponding to different inputs. We also present a procedure that exploits the symmetries of the cluster centers to reduce somehow the number of clusters, therefore reducing the computational cost and extending the applicability of the method. The proposed clustering technique is fully unsupervised in that no knowledge of a training sequence is required (i.e., a blind technique). We only need a few pilots (known symbols) to label the clusters and to decode the transmitted data. Some preliminary results have been presented in Van Vaerenbergh, Estébanez et al., (2007). The proposed technique achieves better results than state-of-the-art MIMO decoding techniques for time-correlated channels such as the generalized decision feedback equalizer proposed by Choi et al. (2005).

Some extensions of this idea will also be discussed. For instance, the use of orthogonal space-time block coded (OSTBC) MIMO schemes, such as the popular Alamouti encoding (Alamouti, 1998), allows us to exploit the structure imposed by the code in the clustering stage. For fast time-varying channels, up to certain Doppler spreads, we obtain better results than a space-time differential code.

This chapter is organized as follows. In Section 2 a short introduction to the equalization of fast time-varying MIMO communications with flat-fading channels is given. Section 3 discusses the basic ideas behind the most common spectral clustering algorithms. Consecutively, spectral clustering is extended and used to derive an algorithm to decode fast time-varying MIMO channels in Section 4. Finally, a number of test results are given in Section 5 followed by the concluding remarks in Section 6.

12.2 PROBLEM FORMULATION

MIMO systems are used in wireless communications to enhance signal diversity, or spectral efficiency, or both. In a typical MIMO flat-fading system with N_t transmit and N_r receive antennas, the received vector $\mathbf{x}[i] = [x_1[i], \dots, x_{N_r}[i]]^T$ at time i is expressed as

$$\mathbf{x}[i] = \mathbf{H}[i]\mathbf{d}[i] + \mathbf{v}[i] \quad (1)$$

where $\mathbf{H}[i]$ is the $N_r \times N_t$ channel matrix whose elements represent independent flat-fading SISO channels, $\mathbf{d}[i]$ contains the N_t (in general, complex) symbols transmitted by the N_t antennas at time i , and $\mathbf{v}[i]$ represents both spatially and temporally white complex zero-mean Gaussian noise.

In MIMO systems with *block fading* channels, variations of the channel during the transmission of one block of symbols are so small that they can be ignored. Hence the channel matrix $\mathbf{H}[i] = \mathbf{H}$ is considered constant during transmission of one block of symbols. This is not the case for MIMO systems with *fast time-varying* channels, where the channel matrix changes from symbol to symbol due to the Doppler spread caused by the movement of the transmitter and/or receiver. In time-varying MIMO systems, depending on the Doppler spread, the channel matrices $\mathbf{H}[i]$ are temporally correlated. The variations can be modeled for instance by the Clarke-Gans model (Rappaport, 2001) which states that if a vertical $\lambda/4$ antenna with uniform power distribution is used to transmit a single tone, the received spectrum is

$$S_{E_z}(f) = \frac{1.5}{\pi f_m \sqrt{1 - \left(\frac{f - f_c}{f_m}\right)^2}}, \quad (2)$$

where f_c and f_m are the carrier frequency and the maximum Doppler shift, respectively.

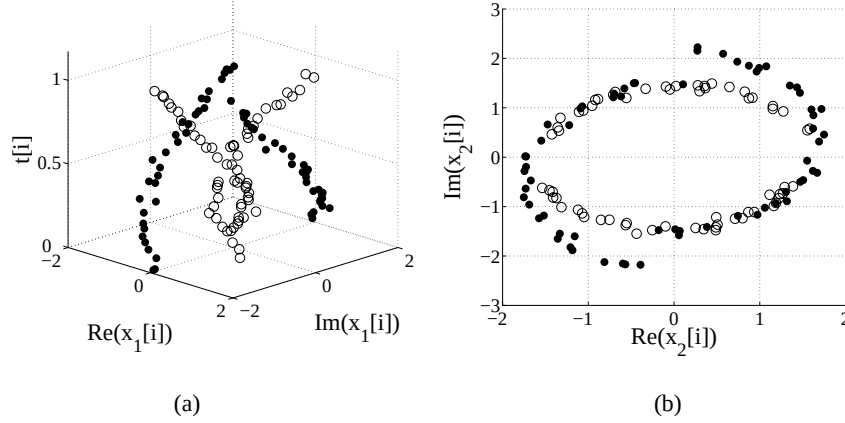


Figure 1: Scatter plots of the data received by the two antennas of a 2×2 BPSK MIMO system with fast time-varying channels. The points corresponding to the symbols $[+1, +1]^T$ and $[-1, -1]^T$ are represented by circles and the points corresponding to $[+1, -1]^T$ and $[-1, +1]^T$ by black dots, emphasizing the symmetry of the constellation used. Note that no such information is available at receiver side.

The goal of blind decoding consists of estimating the symbols $\mathbf{d}[i]$ given the received datapoints $\mathbf{x}[i]$. This problem is illustrated in Figure 1, which shows typical scatter plots of the complex data $x_1[i]$ and $x_2[i]$ received by the two antennas in a time-varying 2×2 MIMO system with binary phase-shift keying (BPSK) modulation, for which the basic constellation points are $\mathbf{d}[i] \in \{+1, -1\}$. Classical clustering algorithms that operate directly on the data from these scatter plots will fail due to overlapping of the clusters. In the next section we give a short introduction to spectral clustering, a clustering technique that is able to solve more complicated clustering problems and that we will extend in Section 4 to decode time-varying MIMO systems.

12.3 SPECTRAL CLUSTERING

Clustering is an important problem in unsupervised learning. It consists of the partitioning of data into dissimilar groups of similar data. Common efficient clustering algorithms include k-means, fuzzy c-means and expectation

maximization (EM) learning. Yet when the datapoints form complex shapes, these classical clustering algorithms fail.

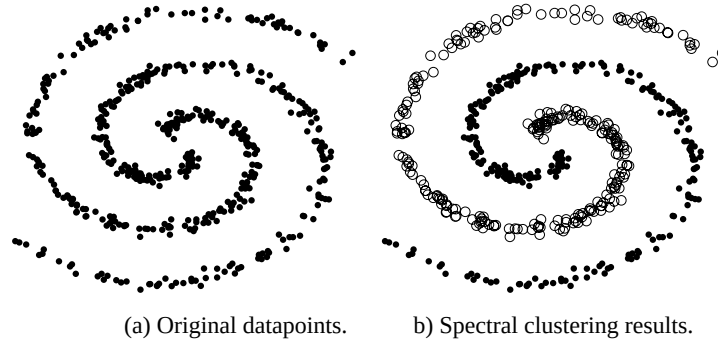


Figure 2: (a). Two sets of intertwined datapoints, difficult or impossible to cluster with conventional clustering algorithms. (b) Spectral clustering easily divides the points into two separate clusters, based on the principle that two points should be in the same cluster if they are close to each other.

A promising alternative is the recently proposed spectral clustering technique (Ng, Jordan et al., 2002), which is capable of clustering non-convex datasets as illustrated in Figure 2. Here, a spectral analysis of point-to-point statistics is performed by calculating the eigenvectors of the kernel matrix (also called *similarity* or *affinity* matrix). This reduces the original hard clustering problem to an easier clustering problem which can be solved by conventional clustering algorithms in the eigenspace of the kernel matrix. Originally rooted in graph theory and kernel methods (Chung, 1997, Shi, Malik, 2000, Weiss, 1999), spectral clustering can be derived as a relaxation of the classical graph bipartitioning problem (Chung, 1997). A good introduction to different forms of spectral clustering can be found in Alzate, Suykens, (2006). In the following we present a summary of this analysis.

12.3.1 Deduction from graph theory

In graph theory, a set of datapoints $D = \{\mathbf{x}_i\}_{i=1}^N$ can be represented as an undirected graph. The datapoints are the nodes of this graph, and an edge is created between each pair of nodes. A weight can be associated to each edge, representing the similarity between the points in each node pair. A widely used

weight function is the *Gaussian kernel*, which uses the Euclidean distance between two points $\mathbf{a}$ and $\mathbf{b}$ as a distance measure:

$$w(\mathbf{a}, \mathbf{b}) = \exp\left(-\frac{\|\mathbf{a} - \mathbf{b}\|^2}{\sigma^2}\right), \quad (3)$$

where σ is a suitable *kernel width*.

Given an undirected graph $G = (V, E)$ where V is the set of N nodes and E is the set of edges, the problem of graph bipartitioning consists of separating the graph into two sets A , B by eliminating edges connecting the two sets. The sets should be disjoint such that: $A \cup B = V$ and $A \cap B = \emptyset$. The total weight of the edges that have to be eliminated is called the cut:

$$\text{cut}(A, B) = \sum_{\mathbf{a} \in A, \mathbf{b} \in B} w(\mathbf{a}, \mathbf{b}). \quad (4)$$

13.1.1.1 Mincut

One of the most elementary criteria to perform graph bipartitioning is the *mincut* criterion (Fiedler, 1975), which is formulated as follows:

$$\min_q J_{\text{mincut}} = \text{cut}(A, B) = \frac{1}{2} \sum_{i,j} w(i, j)(q_i - q_j)^2 \quad (5)$$

where q_i is a cluster membership indicator:

$$q_i = \begin{cases} 1, & \text{if } i \in A \\ -1, & \text{if } i \in B. \end{cases} \quad (6)$$

The original mincut problem is NP-hard (Chung, 1997) due to the constraint on q_i . However, a suboptimal solution can be found by relaxing this constraint and letting q_i take real values. Define $\mathbf{D} = \text{diag}(d_1, \dots, d_N)$, where $d_i = \sum_j w(i, j)$ is the sum of the weights for all edges that connect to node i , and define $\mathbf{W}$ as the symmetric kernel matrix with the ij -th entry equal to $w(i, j)$. The solution to the relaxed problem with constraint $\bar{\mathbf{q}}^T \bar{\mathbf{q}} = 1$, where $\bar{\mathbf{q}}$ is a vector containing the elements $\bar{q}_i$, is then given by the following eigenvalue problem:

$$\mathbf{L}\bar{\mathbf{q}} = \lambda\bar{\mathbf{q}}. \quad (7)$$

The matrix $\mathbf{L}$ is the Laplacian of the graph and it is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The suboptimal solution $\bar{\mathbf{q}}$ is the eigenvector corresponding to the second smallest eigenvalue (also called the Fiedler vector). The cluster membership indicator q_i is obtained by binarizing $\bar{q}_i$ using a suitable threshold Θ :

$$q_i = \text{sign}(\bar{q}_i - \Theta), \quad i = 1, \dots, N. \quad (8)$$

Due to the fact that there are no restrictions in (5) related to the cluster size, the mincut criterion tends to separate small sets of points. In the following we discuss a criterion that imposes a simple size restriction on the clusters.

12.3.1.1 Normalized cut

In Shi, Malik, (2000), a graph bipartitioning criterion was introduced that normalizes the cost of the cut relative to total weight of each cluster. The resulting *normalized cut* criterion is defined as:

$$\min_{\mathbf{q}} J_{\text{ncut}} = \frac{\text{cut}(A, B)}{d_A} + \frac{\text{cut}(A, B)}{d_B}, \quad (9)$$

where $d_A = \sum_{i \in A} d_i$. Due to this normalization, the normalized cut criterion penalizes small sets or isolated points. The solution to this problem is NP-hard, but if $\mathbf{q}$ can take real values then J_{ncut} is minimized by the eigenvector corresponding to the second smallest eigenvalue of the following generalized eigenvalue (GEV) problem:

$$\mathbf{L}\bar{\mathbf{q}} = \lambda\mathbf{D}\bar{\mathbf{q}}. \quad (10)$$

12.3.1.2 NJW algorithm

A simple yet effective spectral clustering algorithm was presented in Ng, Jordan et al., (2002). The Ng-Jordan-Weiss (NJW) algorithm performs spectral clustering by applying a normalization criterion directly on the kernel matrix, resulting in the following eigenvalue problem:

$$\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{q}} = \tilde{\lambda}\tilde{\mathbf{q}}. \quad (11)$$

This criterion can be obtained from the normalized cut criterion by premultiplying (10) by $\mathbf{D}^{-\frac{1}{2}}$, which leads to

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{q} = \lambda \mathbf{D}^{-\frac{1}{2}}\mathbf{q}. \quad (12)$$

Defining $\tilde{\mathbf{q}} = \mathbf{D}^{\frac{1}{2}}\mathbf{q}$ leads to

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{q}} = \lambda \tilde{\mathbf{q}} \quad (13)$$

$$(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}})\tilde{\mathbf{q}} = \lambda \tilde{\mathbf{q}} \quad (14)$$

$$\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{q}} = \tilde{\lambda}\tilde{\mathbf{q}}, \quad (15)$$

where $\tilde{\lambda} = 1 - \lambda$.

The NJW algorithm provides an elegant way to cluster datasets into multiple subsets, based on the solutions of the GEV (11). It operates as follows:

- (1) Calculate the kernel matrix $\mathbf{W}$ of the points $\mathbf{x}[i]$ using (3), and set $w(i, i) = 0$ for $i = 1, \dots, N$.
- (2) Obtain $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. This normalization will assure that all clusters have similar size.
- (3) Form the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$ where $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ are the k largest eigenvectors of $\tilde{\mathbf{L}}$ and k is the number of subsets to be retrieved.
- (4) Treat the rows of $\mathbf{V}$ as points in R^k , and normalize them to unit length. These points correspond to the original points $\mathbf{x}[i]$ but form compact clusters now. Cluster them with a conventional clustering algorithm such as k-means.
- (5) Assign the original point $\mathbf{x}[i]$ to cluster j if and only if row i of the matrix $\mathbf{V}$ was assigned to cluster j .

12.3.2 Drawbacks and solutions

Although spectral clustering is a very powerful method, it has a number of difficulties which have not been resolved satisfactorily yet. Its main shortcoming is the computational complexity $O(N^2)$ of the eigenvector calculations, where N is the number of datapoints. Common workarounds consist of exploiting the kernel matrix structure, dividing the clustering problem up into subproblems, or approximating the spectral clustering mechanics by alternative algorithms. Other difficulties are the choice of the kernel parameters and the number of clusters to detect. A number of well-motivated rules of thumb as well as some more

theoretically founded methods have been proposed to solve these issues (Ng, Jordan et al., 2002, Bach, Jordan, 2003).

The choice of the kernel size σ in (3) has a high impact on the clustering quality. It is a measure of whether two points are considered similar, and should be of the same order of the distance between similar points. Some rules of thumb have been proposed to set a value for σ , whereas in other cases this value is set manually (Ng, Jordan et al., 2002, Bach, Jordan, 2003). In the following we discuss a simple criterion to automatically determine a suitable kernel size σ .

When the data contains clusters with different local statistics, there may not be a single value of σ that works well for all the data. In (Zelnik-Manor, Perona, 2005) a “local” scaling parameter σ_i is proposed instead of this global parameter. It allows for “self-tuning” of the point-to-point distances by studying the local statistics of the neighboring points of every point $\mathbf{x}[i]$. This leads to the following extension of (3):

$$\tilde{\kappa}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{d(\mathbf{x}[i], \mathbf{x}[j])}{\sigma_i \sigma_j}\right), \quad (16)$$

where $d(\mathbf{x}[i], \mathbf{x}[j])$ is a distance measure such as the Euclidean distance. If σ_i is chosen as

$$\sigma_i = \|\mathbf{x}[i] - \mathbf{x}_L[i]\|^2, \quad (17)$$

where $\mathbf{x}_L[i]$ is the L -th nearest neighbor of $\mathbf{x}[i]$, then spectral clustering with this kernel will interconnect all points with their closest neighbors that have similar σ_i . The value of L has little influence on the clustering result, and it depends mainly on the data dimension of the embedding space. Apart from improving standard spectral clustering in a number of applications, this self-tuning also allows clustering based on density of points (Zelnik-Manor, Perona, 2005).

12.4 DECODING OF FAST TIME-VARYING MIMO CHANNELS THROUGH SPECTRAL CLUSTERING

As discussed in Section 2, the datapoints in a time-varying MIMO communication system may form non-convex and even overlapping clusters in the scatter plots, impossible to cluster with classical algorithms. Spectral clustering can deal with the non-convexity of these groups, but it will fail when the data clusters overlap. In Van Vaerenbergh, Estébanez et al. (2007) a simple

trick was presented to circumvent this problem, consisting of making use of the temporal index, which is usually discarded for scatter plot analysis. It is discussed in detail in the following.

A second problem arises when the number the number of points per cluster is too small. On the one hand, the clustering algorithm deals with one block of data at a time, containing typically $N = 256$ data symbols. On the other hand, the number of clusters to retrieve is M^{N_t} , where M is the size (the *cardinality*) of the source alphabet and N_t is the number of transmit antennas. Both M and N_t need to be small to guarantee a minimum number of points per cluster, which is usually not the case. However, as shown in Van Vaerenbergh, Estébanez et al. (2007), it is possible to exploit the constellation geometry of M -PSK systems to make the data clusters more dense and consequently easier to retrieve.

These ideas are discussed in detail in the following, and they are extended for various MIMO systems. First we discuss the case of spatial multiplexing MIMO systems, in which N_t different symbols are transmitted by the N_t antennas at a given time instant.

12.4.1 Adding the temporal dimension into the clustering problem

The received data $\mathbf{x}[i]$ in a fast time-varying MIMO system can be preprocessed for spectral clustering by simply adding the temporal dimension to the vector of observations. The combined vector representing one datapoint and its temporal index $t[i]$ can be denoted as

$$\mathbf{x}^0[i] = [\mathbf{x}[i]^T, t[i]]^T, \quad (18)$$

which is a complex vector with $N_r + 1$ elements. When this extra dimension is added to the scatter plots of Figure 1, threads appear due to the temporal correlation between consecutive channel matrices (see Figure 3). Thanks to its non-convex clustering capabilities, spectral clustering should be able to retrieve these different threads from $\mathbf{x}^0[i]$.

The performance of the clustering algorithm depends mainly on two factors. In the first place, the number of datapoints N in one block must be larger than the number of clusters (a rule of thumb is to have at least 10 samples per cluster). And since spectral clustering is a computationally costly procedure, this number of clusters, which is M^{N_t} , should be reasonably low. Taking into

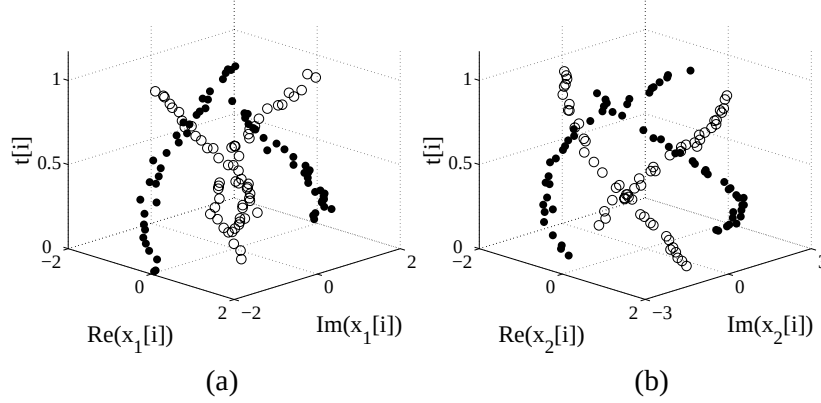


Figure 3: Scatter plots of the data of Figure 1 to which a temporal index was added. Threads of datapoints are now distinguishable in both figures.

account the fact that most commercial MIMO systems use up to $N_t = 4$ transmit antennas, we will only treat BPSK systems ($M = 2$) and QPSK systems ($M = 4$) in this work.

In the second place, clusters should be well connected, i.e., the distance between neighboring points of the same thread should not be larger than the distance between points of different threads. This requires a rescaling of the temporal dimension to match the scale of the spatial dimensions, for instance $t[i] = i/256$, with $i = 0, \dots, 255$ for blocks of 256 symbols. Moreover, this means that if a symbol is not sent during a considerable time, one thread might be incorrectly identified as two separate threads. However, as will be shown in the next section, both difficulties can be reduced by using information derived from the geometric properties of the constellation.

12.4.2 Exploiting the input constellation symmetries

In this section we show that the geometrical symmetries of the transmitted constellation can be used to design a two-phase clustering algorithm, in which only a reduced number of clusters needs to be detected during each phase. Before dealing with the general case of $N_t \times N_r$ M -PSK MIMO systems, the proposed algorithm is illustrated on a simple 2×2 BPSK MIMO system.

12.4.2.1 BPSK MIMO systems

In the noiseless case ($\mathbf{v}[i] = 0$), Eq.(1) can be written as

$$\mathbf{x}[i] = \mathbf{H}[i]\mathbf{d}[i] \quad (19)$$

For a 2×2 BPSK MIMO system, there will be 4 symbol clusters to detect in the data $\mathbf{x}[i]$, corresponding to the transmitted symbol vectors $[+1, +1]^T$, $[+1, -1]^T$, $[-1, -1]^T$ and $[-1, +1]^T$. In Figures 1 and 3 we can observe that for any cluster following a certain trajectory, there is always another cluster following a trajectory symmetric with respect to the origin. This observation is confirmed by (19): since a BPSK system can generate both $\mathbf{d}[i]$ and $-\mathbf{d}[i]$, the datapoint $\mathbf{x}[i]$ as well as its opposite $-\mathbf{x}[i]$ can be received. When transmitted through a time-varying channel, these datapoints lie in clusters that follow symmetric trajectories with respect to the introduced “temporal axis”. This property will be exploited to improve the spectral clustering stage, by first grouping together the datapoints that follow symmetric trajectories.

This geometrical property is not limited to 2×2 systems. In a general $N_t \times N_r$ BPSK MIMO system with fast time-varying channels, for any cluster following a certain trajectory in time, there is always another cluster following the symmetric trajectory. Combining the data of two such clusters will provide a more robust clustering problem. This observation leads to the following two-phase algorithm: In the first phase, groups of symmetric clusters are detected. One clustering problem needs to be solved here to find 2^{N_t-1} clusters. In the second phase, each group of symmetric clusters is separated into two different clusters, representing N_t independent problems.

12.4.2.1.1 Phase 1: Grouping of symmetric clusters.

The following analysis shows how spectral clustering can be extended to find clusters consisting of two symmetric “subclusters” at a time. Similar to (18) we can introduce the “symmetric” datapoint

$$\mathbf{x}^1[i] = \begin{bmatrix} -\mathbf{x}[i]^T, t[i] \end{bmatrix}^T, \quad (20)$$

which contains the point opposite to $\mathbf{x}^0[i]$ with respect to the temporal axis. This new datapoint does not correspond to any real datapoint and can therefore be considered a “virtual” pattern. It will only be used to facilitate the clustering process and the superscript 1 refers to the fact that it is the *first* of these virtual patterns. Other constellations such as QPSK need more virtual patterns per original datapoint, as will be shown below.

Consider the distance measure

$$d(\mathbf{x}[m], \mathbf{x}[n]) = \min \left(\left| \mathbf{x}^0[m] - \mathbf{x}^0[n] \right|^2, \left| \mathbf{x}^0[m] - \mathbf{x}^1[n] \right|^2 \right). \quad (21)$$

This measure is small in two cases: firstly for points that are very close to each other, and secondly for points that are very close to the opposite of each other. If the modified Gaussian kernel (16) is used with this distance measure for spectral clustering, neighboring points as well as opposite points will be grouped together, leading to 2^{N_t-1} clusters. This first phase avoids the incorrect clustering that might occur when some of the threads have a low number of datapoints, by combining the information from symmetric clusters.

12.4.2.1.2 Phase 2: Retrieving the individual clusters.

After having identified the 2^{N_t-1} groups of symmetric clusters, the two individual threads for each group need to be retrieved. Since these subclusters are now separated from the other clusters, the clustering problem is greatly simplified. Problems might occur, however, if one or both clusters contain few datapoints. Luckily, the constellation geometry can be exploited again in this second phase of the problem, making use once more of the symmetry of the subcluster's trajectories.

Specifically, we can increase the number of points in each cluster by adding the symmetric virtual patterns $\mathbf{x}^1[i]$ to the datapoints $\mathbf{x}^0[i]$ that need clustering. As shown above, these symmetric points will lie in one of the two clusters to be retrieved. With each cluster being more dense, the performance of the clustering algorithm will increase. Taking this into account, the second clustering phase consists of the following steps:

- (1) Expand the set of points to cluster, $\mathbf{x}^0[i]$, with their (opposite) virtual points $\mathbf{x}^1[i]$.
- (2) Apply spectral clustering to these points.
- (3) Maintain only the original points $\mathbf{x}^0[i]$ in each group.

Thanks to this procedure, even if one of the clusters has few nearby samples, it tends to have “holes”, or may even be empty, it is still possible to apply spectral clustering to retrieve the two clusters. In the case of an “empty” cluster the second step will correctly leave one of the clusters empty and only identify its position.

12.4.2.2 Generalization to M-PSK MIMO systems

The described procedure to exploit the constellation geometry can be easily extended to M -PSK constellations. In this general case, M^{N_t} clusters need to be retrieved, which boils down to finding M^{N_t-1} clusters in the first phase and M subclusters in the second phase.

In case of BPSK systems the first phase of the clustering algorithm consisted of grouping together clusters that follow symmetric trajectories in time. The main difference for M -PSK systems is that now the clusters to be retrieved should follow trajectories that are rotated over a certain angle α with respect to the introduced temporal axis. For QPSK systems this angle will be $\alpha = \pi/2$, as can be deduced easily from its constellation. The special case of BPSK systems is found for $\alpha = \pi$. In general, we have that $\alpha = 2\pi/M$, where M is the size of the constellation.

The clustering algorithm consists of the same two phases as for BPSK systems. The only difference is that not 1 *symmetric* virtual pattern but $M-1$ *rotated* virtual patterns should be taken into account in both phases. Instead of (21), the distance measure to be used in phase 1 is

$$d(\mathbf{x}[m], \mathbf{x}[n]) = \min_k \left[\|\mathbf{x}^0[m] - \mathbf{x}^k[n]\|^2 \right], \quad k = 0, \dots, M-1, \quad (22)$$

where $\mathbf{x}^k[i] = \left[e^{j \cdot 2k\pi/M} \mathbf{x}^T[i], t[i] \right]^T$ are rotated versions of the datapoint $\mathbf{x}^0[i]$ with respect to the temporal axis. For $k > 0$ the points $\mathbf{x}^k[i]$ represent the virtual patterns of $\mathbf{x}^0[i]$. Finally, in phase 2, all $M-1$ virtual patterns per datapoint should be added to the clustering problem.

An overview of the complete algorithm will be given in Section 3. It includes the decoding stage that maps each cluster to a constellation symbol.

12.4.2.3 Exploiting additional information: OSTBC MIMO schemes

Space-time block coding (STBC) (Alamouti, 1998) has emerged as one of the most promising techniques to exploit spatial diversity in multiple-input multiple-output (MIMO) systems. Among the space-time coding schemes, the orthogonal space-time block coding (OSTBC) (Alamouti, 1998, Tarokh, Jafarkhani et al., 1999) is one of the most attractive because it is able to provide full diversity gain

with very simple encoding and decoding. The best-known OSTBC is Alamouti coding (Alamouti, 1998). It uses the following coding matrix to transmit a block of 2 symbols d_1 and d_2 :

$$\mathbf{D} = \begin{bmatrix} d_1 & -d_2^* \\ d_2 & d_1^* \end{bmatrix} \quad (23)$$

The Alamouti code uses 2 time slots to transmit 2 symbols, which makes it a *rate-1 code*. It uses 2 transmit antennas, where the first one sends the symbols $[d_1, -d_2^*]$ and the second one sends the symbols $[d_2, d_1^*]$. The sequences sent by both antennas are orthogonal, as can be easily verified:

$$[d_1, -d_2^*]^H \cdot [d_2, d_1^*] = d_1^* d_2 - d_2 d_1^* = 0 \quad (24)$$

The structure the Alamouti coding adds to the data can be exploited in the clustering problem as follows. With Alamouti coding, each original symbol pair is sent in two consecutive time slots, where the second time slot contains a transformed version of these symbols. To exploit the redundancy in a clustering framework, new datapoints $\mathbf{x}^A[i]$ can be constructed as follows:

$$\mathbf{x}^A[i] = \begin{cases} [\mathbf{x}^T[i], \mathbf{x}^H[i+1]]^T, & \text{for } i = 0, 2, 4, \dots \\ [\mathbf{x}^T[i], -\mathbf{x}^H[i-1]]^T, & \text{for } i = 1, 3, 5, \dots \end{cases} \quad (25)$$

This transformation is based on the relationship of the time slots: it adds extra components to the original datapoints $\mathbf{x}[i]$ that, due to the code structure, belong to the same “time-slot pair” and therefore contain the same information.

As with spatial multiplexing, the resulting datapoints $\mathbf{x}^A[i]$ will form data clusters, and it is not difficult to verify that the number of these clusters will be the same with or without transformation (25). However, the transformation causes an increase of the embedding dimensionality, which increases the distance between points of different clusters more than the distance between points of the same cluster. This greatly improves the clustering results, as will be shown in the experiments conducted.

Algorithm 1. The spectral clustering-based algorithm to decode time-varying MIMO channels.

In case of Alamouti coding, apply the transformation (23) to the datapoints.

Generate the datapoints $\mathbf{x}^0[i]$ and the virtual patterns $\mathbf{x}^k[i]$, for $k = 0, \dots, M-1$.

Spectral clustering phase 1:

Use the gaussian kernel with distance measure (22) to obtain the kernel matrix W .

Apply the NJW algorithm to retrieve M^{N_t-1} clusters.

Spectral clustering phase 2:

for each obtained cluster

create a set with the datapoints $\mathbf{x}^0[i]$ from this cluster and all corresponding virtual patterns.

Obtain the kernel matrix of this set, using the gaussian kernel (16) with Euclidean distances.

Apply the NJW algorithm to retrieve M final clusters.

Discard the virtual patterns from each cluster.

end for

decode the data symbols by assigning symbol slots to clusters as described in Section 12.3

After embedding the datapoints according to (25), the two-phase spectral clustering process can be applied as previously indicated.

12.4.3 Symbol decoding and final algorithm

Once the symbol clusters have been successfully retrieved, the original time-varying problem has been reduced to a simpler decoding problem, which is the only supervised part of the proposed algorithm. Symbols need to be assigned to each cluster, and to this end a small number of pilot symbols $\mathbf{d}[i]$ are transmitted at the start of the symbol block, specifically N_t pilots must be transmitted during N_t slots. Note that these pilot symbol slots are not needed for the clustering process.

Defining the matrix of pilot symbols $\mathbf{D}_p = [\mathbf{d}[1], \mathbf{d}[2], \dots, \mathbf{d}[N_t]]$ and the matrix of corresponding received data $\mathbf{X}_p = [\mathbf{x}[1], \mathbf{x}[2], \dots, \mathbf{x}[N_t]]$, an approximation of the initial channel matrix $\mathbf{H}$ can be obtained as

$$\hat{\mathbf{H}} = \mathbf{X}_p \mathbf{D}_p^{-1}. \quad (26)$$

The algorithm concludes by assigning the symbol slot $\mathbf{d}$ to the cluster whose first datapoint in time is closest to the vector $\hat{\mathbf{H}}\mathbf{d}$.

The final algorithm to decode fast time-varying M -PSK MIMO systems is summarized in Algorithm 1. Note that associated to each original datapoint $\mathbf{x}[i]$ there is one datapoint $\mathbf{x}^0[i]$ to which the temporal index was added, and $M-1$ virtual patterns $\mathbf{x}^k[i]$, with $k = 1, \dots, M-1$.

12.5 SIMULATION RESULTS

A number of simulations were carried out to illustrate the performance of the proposed algorithms. In all experiments, the following parameters were assumed: the channels were independent Rayleigh flat-fading and the temporal variation of each channel between a transmit and receive antenna pair was based on the Clarke-Gans model (Rappaport, 2001). The symbols $\mathbf{d}[n]$ were grouped into frames consisting of $N = 256$ slots.

The presented spectral clustering-based algorithm was compared to the generalized decision feedback equalizer (GDFF) proposed in (Choi, Yu, et al., 2005): a state-of-the-art adaptive algorithm for decoding fast time-varying MIMO systems. The GDFF algorithm is based on the V-BLAST architecture. In contrast to the presented clustering-based decoding algorithm, the GDFF algorithm is supervised and adaptive. For each time instant, it successively detects and cancels symbols from the received data vector via decision feedback filtering. The filter tap weights and symbol detection order are updated using an RLS-based time- and order-update algorithm. Its complexity is $O(N_t^3)$ but it provides some savings compared to V-BLAST. During its training period, it needs to send a number of pilot symbols to initialize the algorithm.

12.5.1 BPSK systems with spatial multiplexing

In the first set of experiments, a BPSK signal was used with different MIMO system configurations. The number of transmit and receive antennas was $N_t = N_r = 2$ in the first setup, $N_t = N_r = 4$ in the second setup, and $N_t = 2$ and $N_r = 4$ in the last setup. In all cases the normalized Doppler frequencies $f_d T = 5 \cdot 10^{-4}$ and $f_d T = 10^{-3}$ were considered, where $f_d = f_c \cdot v/c$ with

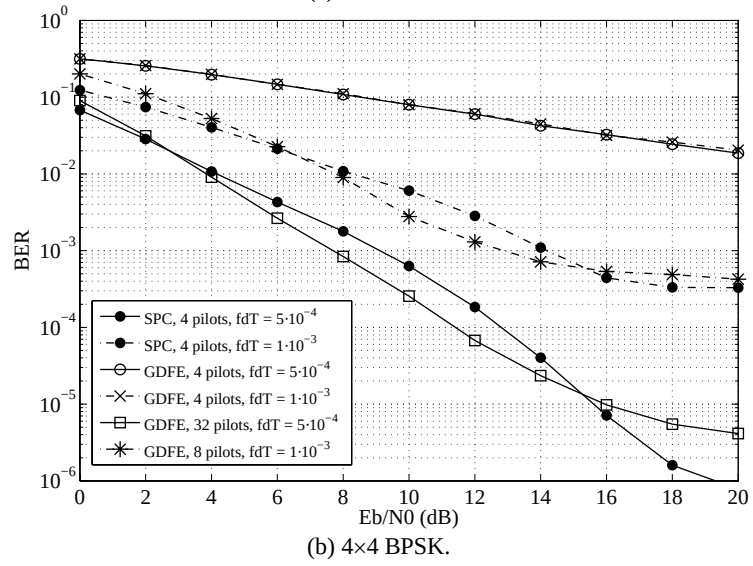
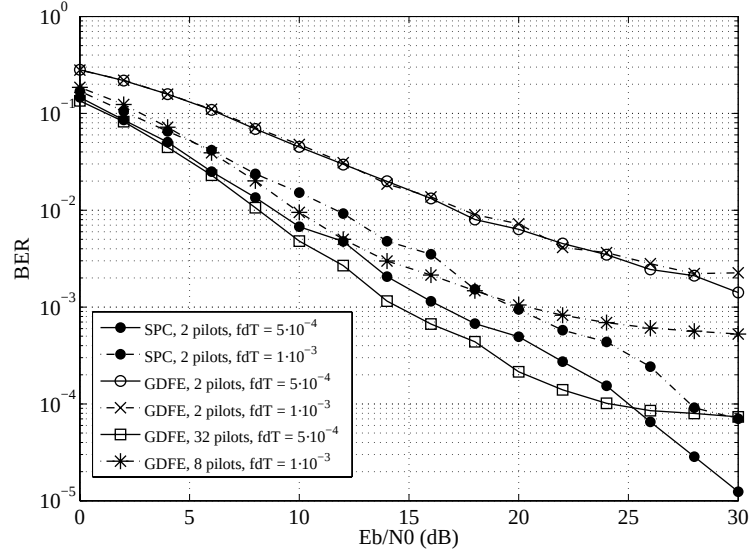


Figure 4: BER curves for different BPSK systems.

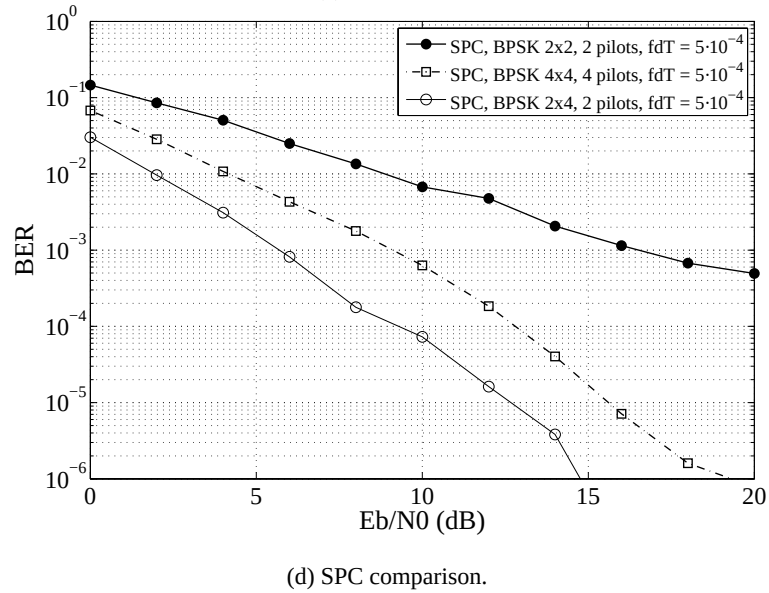
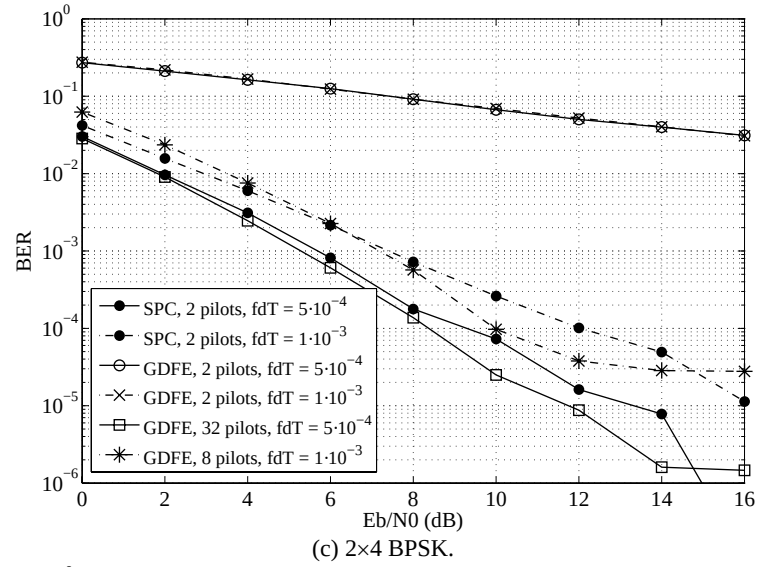


Figure 4: (Continued): BER curves for different BPSK systems.

receiver velocity v and c is the speed of light. For a GSM symbol period $T = 3.7 \cdot 10^{-6}$ seconds and a carrier frequency $f_c = 900$ MHz, these normalized

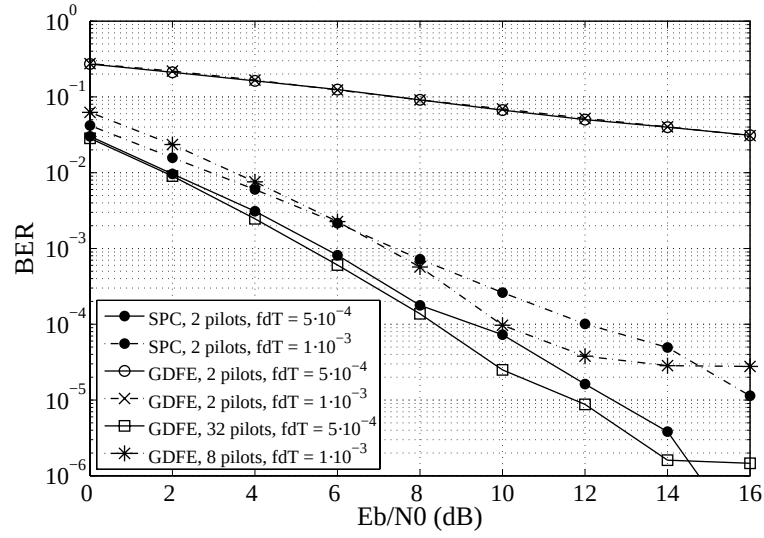
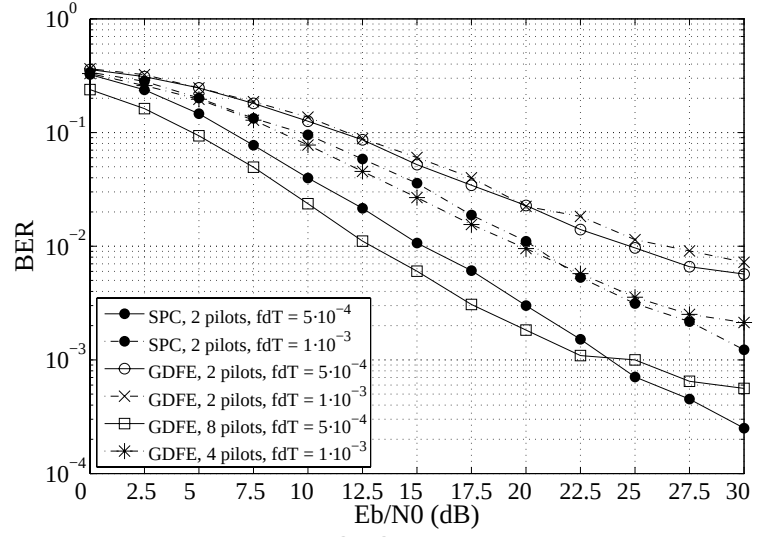


Figure 5: BER curves for different QPSK systems.

Doppler frequencies correspond to receivers moving at 162 km/h and 324 km/h, respectively. For a carrier frequency $f_c = 1800$ MHz, the normalized Doppler frequencies correspond to 81 km/h and 162 km/h, respectively.

The bit error rate (BER) curves of the two algorithms were compared. In the first place the proposed spectral clustering method (referred to as SPC) was tested, in which self-tuning spectral clustering was applied with $L = 5$. The number of pilot symbol slots was fixed as N_t for this method, and they were sent at the beginning of each block. In the second place, the GDFE algorithm from (Choi, Yu et al., 2005) was applied, with forgetting factor $\lambda = 0.95$.

The BER against E_b/N_0 for the 2×2 , 4×4 and 2×4 setups are shown in Figures 4(a), (b) and (c), respectively. Because the GDFE algorithm is essentially a supervised method it requires to transmit more pilot symbols. Therefore, apart from its BER curves for N_t pilots (both of which coincide, in all figures), a second set of BER curves was also displayed for a higher number of pilots, to achieve the same performance as the spectral clustering algorithm. For the three GDFE algorithm needs 32 pilot symbols to achieve similar performance as the presented method when $f_d T = 5 \cdot 10^{-4}$, and 8 pilots when $f_d T = 1 \cdot 10^{-3}$.

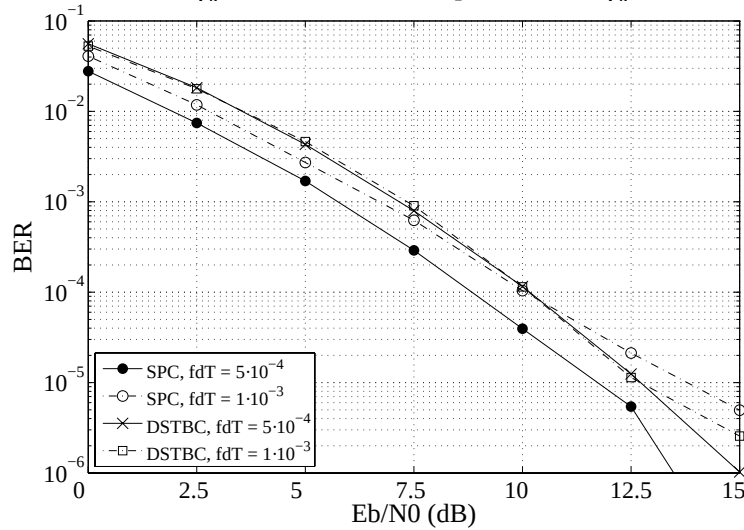
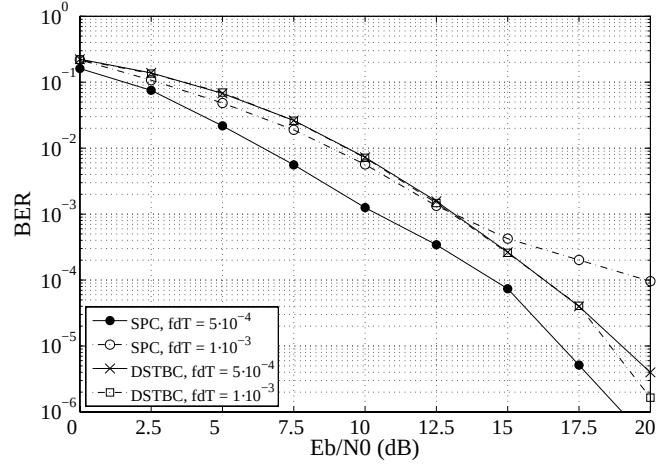


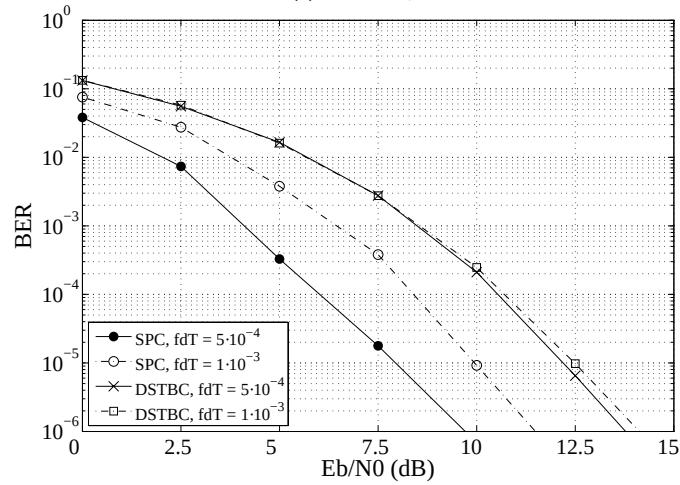
Figure 6: BER curves for a 2×2 BPSK MIMO system with Alamouti coding.

In Figure 4(d) the three obtained BER curves for the proposed algorithm at $f_d T = 5 \cdot 10^{-4}$ are compared. The most significant result of this comparison is the improvement in BER that is experienced when increasing the number of

receiver antennas from 2 to 4 (black dots vs. circles). This can be explained by observing that the clusters will be more separated when new dimensions are added.



(a) 2×2 QPSK.



(b) 2×4 QPSK.

Figure 7: BER curves for different QPSK systems with Alamouti coding.

$M^{N_t} = 4^4 = 32$ which results in clusters with an average of $256/32 = 8$ points. This would be insufficient to reliably detect clusters, even when exploiting the constellation geometry.

12.5.2 QPSK systems with spatial multiplexing

In the second set of experiments, a QPSK signal was used for different MIMO system configurations. We only include results for the cases when the number of transmit antennas was $N_t = 2$, more specifically a 2×2 and a 2×4 scenario. In case 4 transmit antennas were used, the number of clusters would increase to

The proposed spectral clustering was applied as in the previous experiment, with 2 pilot symbol slots. In Figure 5, its results are compared to the GDFE algorithm, for which the first 2 pilot symbol slots were used, and this number of pilots was then changed to obtain similar performance. We can observe that 8 pilot symbol slots are sufficient for the GDFE to obtain a similar performance, for both MIMO systems and both tested normalized Doppler frequencies $f_d T$.

In cases where only a few pilot symbols can be sent, the spectral clustering algorithm obtains superior performance for the tested MIMO systems. However, it requires the calculation of the eigenvectors of its kernel matrix, which generally requires $O(N^3)$ operations. In most cases this can be lowered to $O(N^2)$ (Golub, Van Loan, 1996) taking into account the fact that the kernel matrix is symmetric and can be approximated by a tridiagonal matrix. This analysis suggests that spectral clustering could be used as an initialization for the GDFE or any other supervised algorithm. Specifically, given only N_t pilot symbol slots it can estimate a short symbol vector sequence which can be used as a pilot sequence for a (computationally more efficient) supervised algorithm.

12.5.3 MIMO systems with Alamouti coding

The experiments from Sections 1 and 2 were repeated with Alamouti-coded data. Here, the results of the spectral clustering algorithm were compared to the differential space-time block code detection (DSTBC) (Gao, Lao et al., 2002), which can also be decoded in a blind fashion (i.e. no channel state information is required). Both algorithms were used with N_t pilot symbols.

The results for a 2×2 BPSK system are displayed in Figure 6. Results for 2×2 and 2×4 QPSK systems can be found in Figure 7. For the 2×2 systems, both algorithms obtain very similar results at the tested Doppler spreads. The differences between both methods can be observed more easily in Figure 8, where the results for the 2×2 BPSK scenario are shown as a function of the normalized Doppler spread $f_d T$ and for a noise level of 10 dB SNR. The

spectral clustering algorithm outperforms the DSTBC algorithm for moderately high Doppler spreads up to $f_d T = 1 \cdot 10^{-3}$.

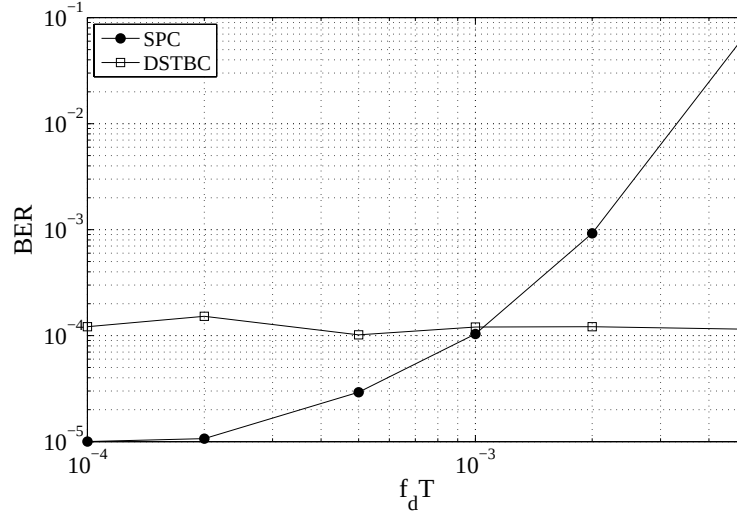


Figure 8: BER comparison between the spectral clustering and DSTBC algorithms, for a 2x2 BPSK system and with respect to the normalized Doppler frequency $f_d T$.

12.6 CONCLUDING REMARKS

We presented a novel clustering algorithm that is capable of decoding fast flat-fading time-varying M -PSK MIMO channels. This algorithm operates on the received data to which a temporal dimension is added, and it exploits constellation geometry to retrieve symbol threads from it. The only supervised part of the presented clustering method is the final decoding phase.

If only a few pilot symbol slots are transmitted, the presented algorithm achieves superior BER performance compared to the tested supervised algorithm. The algorithm was also extended to exploit the additional structure introduced by the Alamouti code. Although the algorithm can be used for different M -PSK systems, in practice it is limited to moderate numbers of transmit antennas and BPSK and QPSK constellations.

ACKNOWLEDGMENTS

This work was supported by MEC (Ministerio de Educación y Ciencia) under grant TEC2007-68020-C04-02 TCM, and the work of Steven Van Vaerenbergh was supported by FPU grant AP2005-5366.

References

- Alamouti, S. (1998). A simple transmit diversity technique for wireless communications. *Selected Areas in Communications*, IEEE Journal on 16 (8), 1451–1458.
- Alzate, C., Suykens, J. (2006). A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods. In: *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. pp. 138–144.
- Bach, F. R., Jordan, M. I. (2003). Learning spectral clustering. Tech. rep., EECS Department, University of California, Berkeley.
- Chen, S., Mulgrew, B., Grant, P. (1993). A clustering technique for digital communications channel equalization using radial basis function networks. *IEEE Transactions on Neural Networks* 4 (4), 570–590.
- Choi, J., Yu, H., Lee, Y. (2005). Adaptive MIMO decision feedback equalization for receivers with time-varying channels. *IEEE Transactions on Signal Processing* 53 (11), 4295–4303.
- Chung, F. (1997). Spectral Graph Theory. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society.
- Cid-Sueiro, J., Artés-Rodríguez, A., Figueiras-Vidal, A. R. (1994). Recurrent radial basis function networks for optimal symbol-by-symbol equalization. *Signal Process.* 40 (1), 53–63.
- Diamantaras, K. I. (2006). A clustering approach for the blind separation of multiple finite alphabet sequences from a single linear mixture. *Signal Processing* 86 (4), 877–891.
- Diamantaras, K. I., Papadimitriou, T., (2006). Blind deconvolution of multi-input single-output systems with binary sources. *IEEE Transactions on Signal Processing* 54 (10), 3270–3731.
- Fiedler, M. (1975). A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal* 25, 619–633.
- Foschini, G. J., Golden, G. D., Valenzuela, R. A., Wolniansky, P. W. (1999). Simplified processing for high spectral efficiency wireless communication employing multi-element arrays. *IEEE Journal on Selected Areas in Communications* 17 (9), 1841–1852.
- Gao, C., Lao, D., Haimovich, A., (2002). Bit error probability for space-time block code with coherent and differential detection. *Proc. 56th Vehicular Technology Conference (VTC*

- 2002-Fall) 1, 410–414.
- Golub, G. H., Van Loan, C. F., (1996). *Matrix Computations*. The John Hopkins University Press.
- Karami, E., Shiva, M., (2006). Decision-directed recursive least squares MIMO channels tracking. *EURASIP Journal on Wireless Communications and Networking* 2006, Article ID 43275, 10 pages.
- Kopsinis, Y., Theodoridis, S. (2003). A novel cluster based MLSE equalizer for M-PAM signaling schemes. *Signal Processing* 83 (9), 1905–1918.
- Lee, J., Beach, C., Tepedelenlioglu, N. (1999). A practical radial basis function equalizer. *IEEE Transactions on Neural Networks* 10 (2), 450–455.
- Mitchinson, B., Harrison, R., (2002). Digital communications channel equalization using the kernel adaline. *IEEE Trans. Communications* 50 (4), 571–576.
- Montalvão Filho, J., Dorizzi, B., Mota, J. M., (2002). Channel estimation by symmetrical clustering. *IEEE Trans. Signal Processing* 50 (6), 1459–1469.
- Ng, A. Y., Jordan, M. I., Weiss, Y., (2002). On spectral clustering: Analysis and an algorithm. In: Dietterich, T. G., Becker, S., Ghahramani, Z. (Eds.), *Advances in Neural Information Processing Systems 14*. MIT Press, Cambridge, MA, pp. 849–856.
- Pérez-Cruz, F., Navia-Vázquez, A., Alarcón-Diana, P., Artés-Rodríguez, A., (2001). SVC-based equalizer for burst TDMA transmissions. *Signal Processing* 81, 1681–1693.
- Rappaport, T. (2001). *Wireless Communications: Principles and Practice*. Prentice Hall PTR.
- Rontogiannis, A., Kekatos, V., Berberidis, K., (2006). A square-root adaptive V-BLAST algorithm for fast time-varying MIMO channels. *IEEE Signal Processing Letters* 13 (5), pp. 265–268.
- Santamaría, I., González, R., Pantaleón, C., Príncipe, J. C., (2003). Maximum margin equalizers trained with the Adatron algorithm. *Signal Processing* 83 (3), 593–602.
- Sebal, D., Bucklew, J., (2000). Support vector machine techniques for nonlinear equalization. *IEEE Trans. Signal Processing* 48 (11), 3217–3226.
- Shi, J., Malik, J., (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8), 888–905.
- Tarokh, V., Jafarkhani, H., Calderbank, A. R. (1999). Space-time block codes from orthogonal designs. *IEEE Transactions on Information Theory* 45 (5), 1456–1467.
- Van Vaerenbergh, S., Estébanez, E., Santamaría, I. (2007). A spectral clustering algorithm for decoding fast time-varying BPSK MIMO channels. In: *15th European Signal Processing Conference (EUSIPCO 2007)*. Poznan, Poland.
- Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. In: *Proc. Int. Conf. on Computer Vision (ICCV)*. Kerkyra, Greece, pp. 975–982.
- Zelnik-Manor, L., Perona, P. (2005). Self-tuning spectral clustering. In: Saul, L. K., Weiss, Y., Bottou, L. (Eds.), *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, pp. 1601–1608.

Chapter 13

Swarm Intelligence Based Link Stability Routing Optimisation Algorithm for Mobile Ad hoc Network

Wei Ren, Evor L. Hines, Mark S. Leeson, Yousef S. Kavian and Daciana Iliescu

13.1 INTRODUCTION

A mobile ad hoc network (MANET) consists of a collection of wireless nodes, which dynamically form a temporary network, without using any existing infrastructure of wired networks or centralized administration (Corson and Macker 1998). Current routing protocols for MANET can generally be categorized as either proactive routing protocols or reactive routing protocols. Proactive routing schemes like destination sequenced distance vector (DSDV) (Perkins and Bhagwat 1994) constantly maintain and update routing tables of each node in the network, even for those nodes that currently have nothing to transmit. This reduces the available capacity of the network for actual data communication. The reactive routing protocols, such as ad hoc on-demand distance vector (AODV) (Perkins, Royer et al. 1999) and dynamic source routing (DSR) (Johnson and Maltz 1996), have the advantage of not having to maintain extensive routing tables for all destinations in the network. In AODV, for example, routes are only discovered on an as-needed basis and only maintained as long as they are necessary. Due to the large end to end time delay, this may not be suitable for real-time data and multimedia communication applications.

In recent years, there has been great interest in so called swarm intelligence (SI). Many algorithms based on SI have been proposed for efficient routing in MANET, see for example Caro and Dorigo 1998; Tham Marwaha et al. 2002; Ziane and Mellouk 2005; Rajagopalan and Shen 2006; Wu, Song, et al. 2007. These protocols in general are hybrid routing protocols that try to take advantage of both proactive and reactive routing protocols to balance the delay and control

overhead. However, most of these algorithms have certain drawbacks. Moreover, none of the existing SI based algorithms have considered the link stability issues.

The routing protocol introduced here tries to improve the routing efficiency in MANET by introducing a link stability function to produce a new hybrid SI based routing protocol. The proposed routing algorithm is highly adaptive, efficient, and scalable. Compared with other SI based algorithms, this protocol uses, whenever possible, stable links for data transmission, thus minimizing any possible flooding in the network.

The chapter is structured as follows: Section 2 provides the background to this work. Section 3 focuses on the proposed swarm intelligence based link stability routing algorithm. Section 4 presents the simulation and experimental results. Finally section 5 presents conclusions and suggestions for further work.

13.2 BACKGROUND

13.2.1 *Traditional MANET routing protocols*

Since the advent of DARPA packet radio networks in the early 1970s (Jubin and Tornow 1987), numerous protocols have been developed for ad-hoc mobile networks. Such protocols must deal with the typical limitations of these networks, which include high power consumption, low bandwidth, and high error rate. As shown in Figure 1, these routing protocols may generally be categorized as: proactive routing protocols or reactive routing protocols.

13.2.1.1 *Proactive routing protocols*

In proactive (table-driven) protocols, nodes periodically search for routing information within a network. The control overhead for these protocols is foreseeable, because it is independent of the traffic profiles and has a fixed upper bound. This is a general advantage of proactive routing protocols.

One typical proactive routing protocol is called destination sequenced distance-vector routing (DSDV) (Perkins and Bhagwat 1994). It is based on the idea of the classical Bellman-Ford routing algorithm with certain improvements such as making it loop-free. The DSDV is the foundation of many other distance vector routing protocols such as AODV which is addressed later.

13.2.1.2 Reactive routing protocols

The reactive (on-demand) routing protocols represent the true nature of ad hoc networks, which is much more dynamic than infrastructured networks. Instead of periodically updating the routing information, the reactive routing protocols update routing information when a routing requirement is presented, consequently reducing the control overhead, especially in high mobility networks where the periodical update will lead to significant wasted overhead.

The AODV routing protocol described in Perkins, Royer, et al. 1999 is a typical example of reactive routing protocols. The protocol is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes on an on-demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. In this work, AODV is chosen to represent the classical MANET routing protocol, and compared with our proposed algorithm. A brief overview of AODV is presented here.

If a node using AODV desires to send information to a destination node, it first looks up its own routing table to see if a valid route exists. If a valid route does not exist, a source node broadcasts a route request (RREQ) message to its neighbours, which then forwards the request to their neighbours, and so on, until either the destination or an intermediate node with a route to the destination is located. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates. Along with its own sequence number and the broadcast ID, the source node also includes the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination with a destination sequence number equal to or greater than the one listed in the RREQ. If additional copies of the same RREQ are later received, these packets are simply discarded.

When the RREQ reaches the destination or an intermediate node (having fresh enough route to the destination), it responds by sending a route reply (RREP) packet to the source. Periodic 'HELLO' broadcasts are used in AODV by the nodes in the network to inform each mobile node of other nodes in its neighbourhood. These broadcasts are used to maintain local connectivity. If a node along the route moves, its upstream neighbour notices the move and propagates a link failure notification route error message (RERR) to each of its

active upstream neighbours to inform them of the removal of that part of the route.

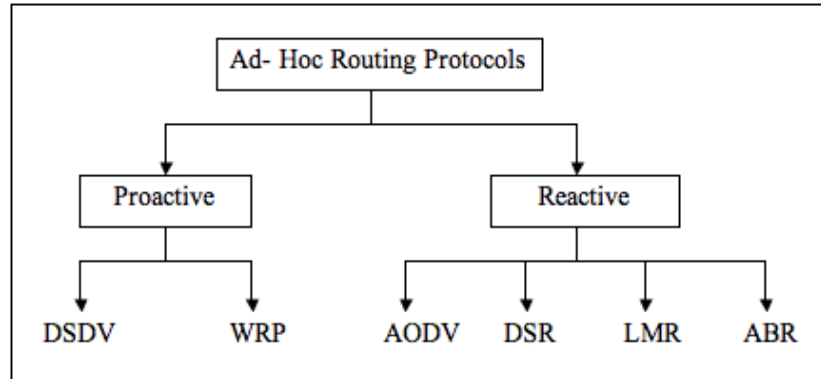


Figure 1: Categorization of ad-hoc routing protocols.

13.2.2 Swarm intelligence based routing algorithms

Typical biological swarms, such as ants or honeybees, often contain thousands of individuals. They perform extraordinarily complex tasks such as global optimization and resource allocation using only ‘local’ information. One famous example of biological-swarm behaviour is ant colony foraging (Becker Deneubourg, et al. 1992) (see Figure 2). Many ant species have trail-laying trail-following behaviour when foraging (searching for food): individual ants deposit a chemical substance called ‘pheromone’ as they move from a food source to their nest, and foragers follow such pheromone trails. Subsequently, more ants are attracted by these pheromone trails and in turn reinforce them even more. As a result of this autocatalytic effect, the optimal solution will emerge rapidly. The concentration of pheromone on a certain path is an indication of its usage. With time, the concentration of pheromone decreases due to diffusion effects. This Property is important because it integrates dynamics into the path search process. The ability of the social ant to self-organise is understood to depend on four principles: positive feedback, negative feedback, randomness and multiple interactions.

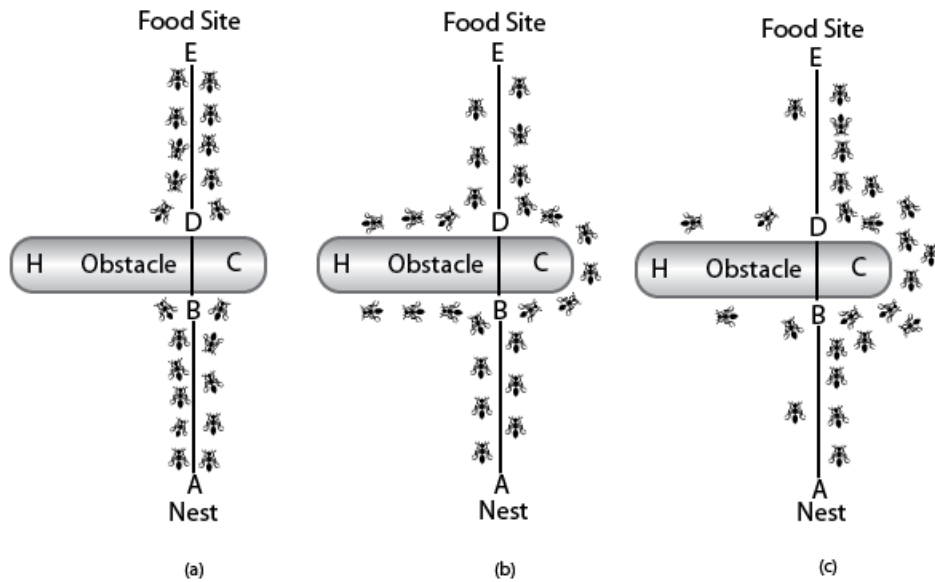


Figure 2: Schematic of ant colony foraging behaviour.

Observations of the foraging behaviour of real ants have inspired the development of a large number of ant-based algorithms, used to solve mainly combinatorial optimisation problems defined over discrete search spaces. Caro and Dorigo's AntNet (Caro and Dorigo 1998) is one of the well-known ant colony based routing algorithms. It was originally designed for routing in packet-switch networks. Its objective was to optimise the performance of the entire network. In AntNet, routing was achieved by launching forward ants at regular intervals from a source node N_s to a destination node N_d to discover a feasible low-cost paths and by backward ants that travels from N_d to N_s to update pheromone tables at each intermediate node. The goodness of a path is reinforced according to the trip times of forward ants, and this goodness value is used to formulate the communication route from the source node N_s to the destination node N_d . With similar ideas, many routing protocols (Tham Marwaha et al. 2002; Ziane and Mellouk 2005; Rajagopalan and Shen 2006; Wu, Song, et al. 2007) have been developed for MANETs.

13.2.3 Swarm intelligence versus traditional routing

Sim and Sun (2003) discussed the differences between swarm intelligence base routing and the traditional routing algorithms from the following three perspectives.

Routing information: In traditional routing algorithms, a node N_i depends on the routing information furnished by all its neighbouring nodes to construct a complete routing table. And in SI based routing; the paths from a source to a destination are explored independently and in parallel.

Routing overhead: Generally speaking, routing in traditional routing algorithms involves the transmission of routing tables of each node N_i to every one of its neighbours. For a large network N , the routing table of N_i , which consists of a list of cost vectors to all other nodes in N , is large. Since each N_i needs to transmit its routing table to everyone of its neighbouring nodes, the routing overhead can be very large. Routing in SI based routing algorithms is achieved by ‘transmitting ants’ rather than routing tables. Even though it is noted that the size of an ant may vary in different systems or implementations, in general, the size of ants is relatively small. This is because ants are generally very simple agents.

Network adaptivity: In dynamic networks transmitting large routing table in short or regular intervals may incur large routing overhead. However, the possible flooding of the routing table in longer intervals may result in slower responses to changes in network topology. Since ants are relatively small they can be ‘piggy backed’ in data packets. Hence more frequent transmission of ants to provide updates of routing information may be possible. Therefore, using SI based algorithms for routing in dynamic network seems to be an appropriate area for research.

Table 1 summarizes the differences between traditional routing and SI based algorithms.

Table 1: Traditional routing algorithms vs SI algorithms.

	Traditional Algorithms	SI Algorithms
Routing Preferences	Based on transmission time/delay	Based on pheromone concentration
Exchange of routing information	Routing information and data packet transmitted separately	Can be piggybacked in data packets
Adapting to topology change	Transmit entire routing table	Frequent Transmission of ants
Routing overhead	High	Low
Routing update	Update entire routing table	Update an entry in a pheromone table independently

13.2.4 Link stability issues

In MANET, the topology of the network changes frequently. Each time a link that is currently communicating breaks will cause the current package to be dropped and also may cause the source node to perform “route discovery” processes that will result in a “flood” in the whole network, which is very costly. Therefore, the re-routing of existing connections should be carried out as seldom as possible. In other words, routes should be established along “stable”, i.e. durable, paths. In the design of most ad hoc routing protocols, this issue has not been addressed. Most of the protocols focus on parameters like finding the shortest path to save bandwidth. Or finding the lowest delay, which is likely to be a path with generally low delay and hop count.

Only a few protocols aim to establish stable routes. The associativity based routing (ABR) (Toh 1997) is a protocol that prefers “stable” links over “transient” links. A link is considered stable if it exists for a time duration of at least $A_{thresh} = 2r_{tx} / v$, where r_{tx} is the transmission range and v denotes the relative speed of two devices. It is left open how to determine the relative speed v among the mobiles which in turn determines A_{thresh} . Signal stability adaptive routing (SSA) (Dube, Rais, et al. 1997) follows a similar approach. It distinguishes strongly connected from weakly connected links where a link is considered to be strongly connected, if it has been active for a certain predefined amount of time. Paths are established exclusively along stable links. In Wu, Xu, et al. (2007) and Jiang, He, et al. (2001), a prediction heuristics for a link’s residual life time is introduced. The algorithm relies on the availability of GPS

receivers to acquire distance and velocity information of neighbouring nodes. Apart from the disadvantages associated with these methods, e.g. unavailability in indoor environments or high demand for battery power. The problem with this approach is that the distance of a receiver is only a very vague estimate of link availability in realistic scenarios.

In this work, the link stability function we use is an algorithm based on statistical methods. We require each node to keep a table to measure the link stability of all the paths to its available neighbours. The longer the link remains the better the link stability index is kept for the path. The equations for the link stability function are presented in Section 13.2.2.

13.3 SWARM INTELLIGENCE BASED LINK STABILITY ROUTING ALGORITHM

13.3.1 Introduction

MANETs are known to be dynamic infrastructureless networks, most of the research concerning MANETs is focused on the fast changing network topology. However, it is worth noticing that in real life, stable links exist everywhere, even within these highly dynamic infrastructureless MANET networks. For example, if a MANET has been setup in an office environment, desktop PCs or charging laptops can be considered as relatively stable nodes, and two of them within the communication range can set up a stable link. Another example is a motorway MANET scenario. In this case, the relatively stable nodes will be the vehicles traveling in the same direction with the same speed.

As discussed earlier, a broken link in MANET is especially costly, so ideally the stable link should be used as much as possible to minimize the delay and bandwidth wastage caused by a broken link in MANETs.

For the above reasons, a new routing protocol for MANETs is developed. The SI based link stability routing algorithm (SILS) is a hybrid routing algorithm. It is based on the ad hoc networking with SI (ANSI) (Rajagopalan and Shen 2006) routing protocol, and combined with a link stability vector. Experiments show that the SILS protocol can recognize the relatively stable links in the network well and adapts quickly to the change of MANETs topology.

13.3.2 SILS data structures

13.3.2.1 Ant Structure:

An artificial ant is a simple computational agent that models the foraging behavior of real ants. The artificial ant can be divided into the forward ant and the backward ant, both have the same data structure. The following information is carried by an ant π :

- The ID of the ant, which is the (node ID, sequence number) pair,
- The number of nodes, m , which π visits, including the node π originated from,
- The nodes visited stack S_π , containing information about all the nodes $V = \{v_1, v_2, \dots, v_m\}$ along the route, which can be reached by backtracking the ant π 's movement.
- Pheromone amount at $v \in V$, p_v .

13.3.2.2 Ant Decision Table

An ant decision table at node i , A_i , is a data structure that stores pheromone trail information for routing from node i to a destination d via k possible next hop nodes $J = \{j_1, j_2, \dots, j_k\}$. These pheromone trail information are used to compute the routing table. Each ant decision table entry A_{ijd} for node i maintains a row for the destination-next hop pair (d, j) along with the $\tau_{ijd}(t)$, η_{ijd} , $\psi_{ijd}(t)$, and a_{ijd} values described below:

- $\tau_{ijd}(t)$ is the pheromone trail concentration left on a trail ij used as a first hop to destination d at current time t due to all the ants that have traversed the trail, taking into consideration pheromone evaporation. Thus, τ is thereby a measure of the goodness of trail ij . The pheromone τ_{ijd}^π deposited by a backward ant π traversing from destination node d to the source node i via nodes $j \in J$ is given by the equation:

$$\tau_{ijd}^\pi = 1 + \frac{p_i - p_d}{\text{depth}(d)} \quad (i, j, d \in V) \quad (1)$$

where p_i and p_d are the pheromone amounts of ant π at nodes i and d , respectively, the value $\text{depth}(d)$ is the number of hops that the ant π traversed from node i to node d , and $V = \{v_1, v_2, \dots, v_m\}$ denotes the set of m nodes visited by π .

In the ant colony algorithm (ACO), an important characteristic of the ant decision table is the pheromone evaporation. All the values in the ant decision table are evaporated, and so were negatively reinforced. The ‘evaporate’ equations are defined as:

$$evaporate(\tau_{ijd}(t), \delta) = \delta \cdot \frac{\tau_{ijd}(t)}{2^{1/4}} \quad (2)$$

and

$$\tau_{ijd}(t + \delta) = evaporate(\tau_{ijd}(t), \delta) + \tau_{ijd}^{\pi'} \quad (3)$$

where δ is the unit time passed since time t , and $\tau_{ijd}^{\pi'}$ is the pheromone amount laid by the following ants π' .

- (b) η_{ijd}^{π} is the heuristic value of going from i to d via the next hop j . In SILS routing, η_{ijd}^{π} laid by the backward ant π , represents a measure of the distance to the destination d from i . The equation to compute η_{ijd}^{π} is given as:

$$\eta_{ijd}^{\pi} = 1 + \frac{1}{depth(d)} \quad (i, j, d \in V) \quad (4)$$

The value $depth(d)$ is the number of hops that the ant π traversed from node i to node d , and $V = \{v_1, v_2, \dots, v_m\}$ denotes the set of m nodes visited by π .

- (c) $\psi_{ijd}(t)$ is the link stability index for link ij that formulates the route to the destination d at current time t . If the time interval between the two periodical broadcasted ants is Δ , then the equation for the link stability index updated by the following group of ants at time $t + \Delta$ is given as:

$$\psi_{ijd}(t + \Delta) = \sqrt{\psi_{ijd}(t) + 1} \quad (5)$$

if link ij is still available for communication at time $(t + \Delta)$, otherwise,

$$\psi_{ijd}(t + \Delta) = 0 \quad (6)$$

- (d) The $a_{ijd}(t)$ value for a destination d represents the goodness value of choosing j as the next hop to reach d , and is computed using the following formula:

$$a_{ijd}(t) = \frac{[\tau_{ijd}(t)]^\alpha [\eta_{ijd}]^\beta [\psi_{ijd}(t)]^\gamma}{\sum_{l \in J} [\tau_{ild}(t)]^\alpha [\eta_{ild}]^\beta [\psi_{ild}(t)]^\gamma} \quad (7)$$

where parameters α, β, γ define the relative importance of the pheromone concentration $\tau_{ijd}(t)$, the heuristic value of hop distance η_{ijd} , and the link stability index $\psi_{ijd}(t)$, respectively.

13.3.2.3 Routing Table

The routing table at node i is a table containing an entry for each destination d reachable from node i along with the best next hop, j , to d . The best next hop, j , to a destination d is the next hop that contains the largest a_{ijd} value in A_i . The value of j is thereby updated every time a backward ant traverse back to node i .

13.3.3 Description of SILS protocol

Our SILS routing protocol for MANETs is a flexible protocol in which the nodes in the network have proactive and reactive capabilities. A detailed outline of SILS routing protocol is given as follows:

If a route to a destination D is required, but not known at the source node S , S broadcasts a forward reactive ant to discover a route to D . Each reactive ant has an initial pheromone value ϕ_f , and is reduced by 1 every time the reactive ant has been forwarded. This initial pheromone value determines the depth of search to the destination node in the network.

- (1) If a forward ant visits an intermediate node X , X will record this ant's ID and re-broadcast the ant to its neighbors. In order to get rid of 'loops', an ant will not be re-broadcasted if it has been seen at the node.
- (2) When D receives the forward reactive ant from S , it generates a backward reactive ant to the source S . During the forwarding process,

the backward reactive ant updates (a) the pheromone table of all the nodes in the path from S to D , (b) the heuristic value, (c) the link stability vector in each node.

- (3) The node S will periodically broadcast proactive ants to update the pheromone table information and the link stability parameter for the path from S to D until the route to the destination D is no longer needed.
- (4) When a route fails at an intermediate node X , Node X buffers the packets which could not be routed and initiates a route discovery to find D , and sends a route error message back to the source node S . Additionally the pheromone via node X for the path S to D will be decreased, and the link stability vector at X will be reset.
- (5) If an error message is received by the source node S , S buffers the packets, updates the pheromone table, and resends the packet via the current best available route.

13.4 SIMULATION AND EXPERIMENTAL RESULTS

To study the characteristics and evaluate the performance of our SILS routing protocol, we have conducted simulation experiments using the simulator for network algorithms (Sinalgo) environment developed by the Distributed Computing Group. The performance of SILS is compared with ANSI and AODV for the same network and load characteristics.

13.4.1 Simulation and network mobility model

13.4.1.1 Protocol parameters setting

The current implementation of SILS used $\alpha = \beta = \gamma = 2$. For ANSI, we set $\alpha = \beta = 2$. In both the SILS and ANSI protocols, the initial pheromone amount ϕ_f of the ant is set to 4. The ants are periodically broadcasted at every 16 time units.

13.4.1.2 Network mobility models and environment setup

We perform the experiments under two different network mobility scenarios. Both of the simulation environments consist of 100 mobile nodes in a physical region of size 200×200 square unit. All the mobile nodes have a transmission radius of between 30 to 50 unit distances. The nodes in all the experiments are initially been distributed using the Grid distribution model as shown in Figure 3. This model ensures that in all of our repeated experiments, the source and the destination nodes have the same initial distance and also have the same node density around them.

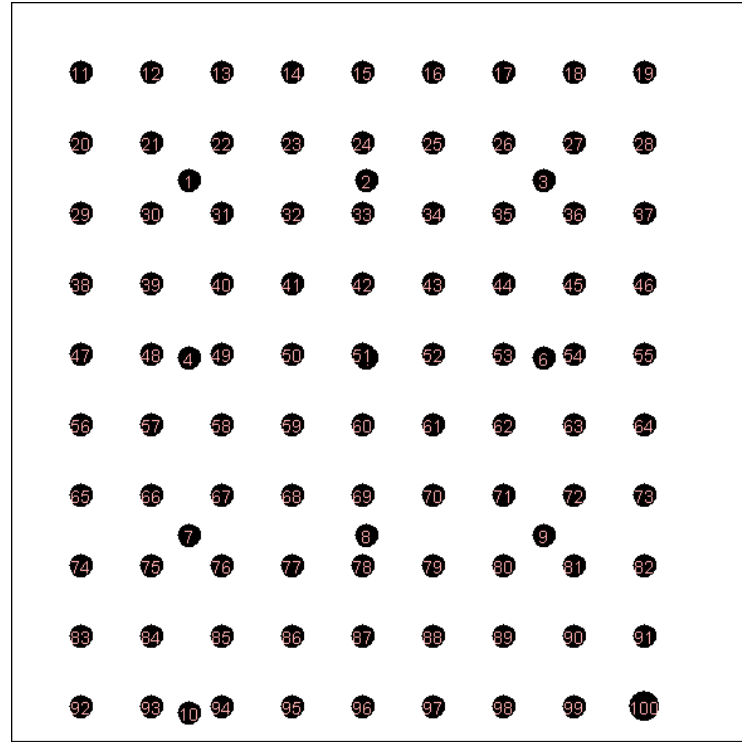


Figure 3: The initial distribution of the nodes.

In the first set of experiments, all the nodes in the network use the random waypoint mobility model. In this mobility model, a node moves on a straight line to a uniformly and randomly selected position in the deployment field. Once it arrives, it waits for a predefined amount of time, before it selects a new position

to walk to. The speed of each node is calculated using a Gaussian distribution with mean=1.0, variance=0.5, the waiting time is set using the Poisson distribution with lambda=50.

In the second set of experiments MANETs in real life office environment has been simulated. In this scenario, we set 10 nodes with no mobility to represent the desktop PCs or charging mobile equipments. The settings for the rest of the nodes are given the same settings as in the first set of experiments

13.4.2 Experimental results

Our first set of experiments was setup under the random waypoint mobility model in order to study what the performance of SILS is like in a common MANETs environment. For the purpose of comparison, we used baseline protocols: AODV and ANSI, as references.

For each protocol, we ask the source node to continuously send 10, 20, 35, 50 units of packages to the destination respectively. Each transmitting process is repeated 10 times, and the average performance results of each protocol are shown in Figures 4 and 5.

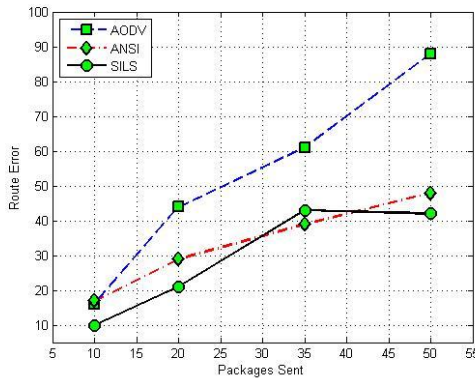


Figure 4: Delivery failure count under the random waypoint mobility model scenario.

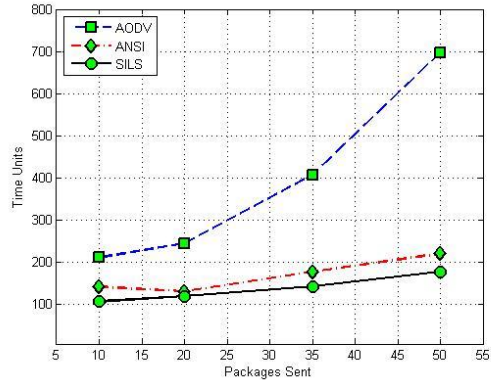


Figure 5: The cost in terms of time units under the random waypoint mobility model scenario.

From Figure 4, we can see that SILS and ANSI have much higher package delivery ratio than AODV. The package delivery failure in SILS/ANSI is reduced by 40%-50% compared to AODV. This shows that the SI based algorithms performs well to increase the MANETs package delivery ratio. It also can be

seen that SILS and ANSI have similar performances under the random way point network mobility scenario. This is because under the current mobility model settings, the topology of the network changes rapidly, thus, the fitness of links is mainly dependent on the pheromone trail concentration and the heuristic value.

In the second set of experiments, the network is initialized with the same node distribution model as we did in the first set of experiments. The only difference is we disabled the mobility of the first 10 generated nodes. By doing this we establish a few 'stable links' in the current network scenario.

Again, we ask the source node to continuously transfer data to destination node with 10, 20, 35, 50 units of packages respectively. Each process is repeated 10 times, and the average performance is shown in Figures 6 and 7.

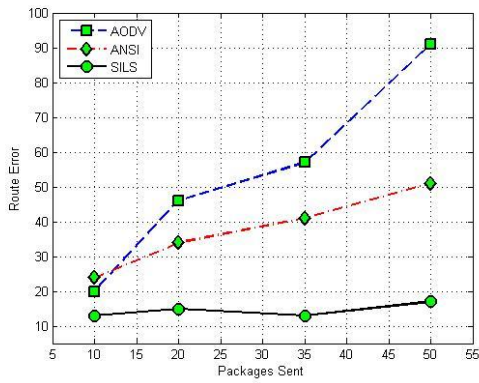


Figure 6: Delivery failure count under the mobility model scenario with 'stable links'.

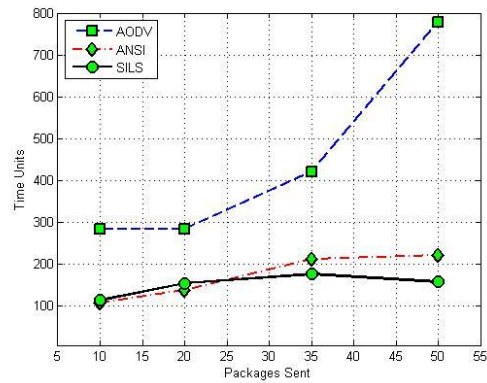


Figure 7: The cost in terms of time units under the mobility model scenario with 'stable links'.

It can be clearly seen from the figures, that our SILS routing protocol has a much better performance than the other two protocols. The link stability function can adapt the stable links effectively. By communication via these stable links, the package delivery failure in SILS when transferring 50 units of packages is reduced by nearly 80% and 67% compared to AODV and ANSI respectively. The results prove that by introducing the link stability function, the package delivery ratio in the MANETs can be effectively optimised. With the reduced 'flooding' within the network, the end to end delay is only 1/3- 1/5 of AODV results, and is reduced by 15% - 30% compared to ANSI.

13.5 CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

In this work we compare the traditional routing protocols and SI based protocols for MANETs, and propose one possible real life mobility scenario for MANETs. Most of the current research consider MANETs to be highly dynamic networks, and it is noted that stable links do exist in such networks. We believe the 'link stability' parameter is a primary key for route optimisation, as they can effectively reduce the package delivery failure.

We also present SILS, a SI based hybrid routing protocol for MANETs. Our simulation shows that in the context of popular network mobility models, such as the random waypoint mobility model, the performance of SILS is as good as ANSI, but has a much lower package delivery failure ratio compared with AODV. However, if stable links exist in the network, SILS is able to recognize them quickly, and the package delivery ratio is increased to 44% - 66%.

Currently, the link stability function for SILS does not perform well under random waypoint mobility model. A link stability function introduced by Gerharz, Waal, et al. (2003) can predict the residual link life time, and works well under random waypoint mobility model. This stability function will be implemented in the next version of our SILS routing protocol. Additionally, further work will include the optimisation of the SILS parameters. As the optimisation process is a multi-variable optimisation problem, it will be interesting to see if it could be realised using evolutionary algorithms such as Genetic Algorithms (GAs).

References

- Becker R., J. L. Deneubourg, S. Goss (1992). "Trails and U-turns in the selection of the shortest path by the ant *Iasius niger*." *Journal of Theoretical Biology*; 159:397-415.
- Caro G.D., and M. Dorigo (1998). "AntNet: Distributed stigmergetic control for communications network," *J.Artif. Intell. Res.*, vol. 9, pp.317-365.
- Corson, S. and J. Macker (1998). "Mobile Ad hoc Networking (MANET): Routing protocol Performance Issues and Evaluation Considerations", Internet draft, draft-ietf-manet-issues-01.txt.
- Distributed Computing Group, "Sinalgo Tutorial" available at:
"http://dgc.ethz.ch/projects/sinalgo/", last accessed on 14/05/2008.
- Dube R., C. D. Rais, K. Y. Wang, and S. K. Tripathi (1997). "Signal stability based adaptive routing (SSA) for ad-hoc mobile networks." *IEEE Personal Communication*.
- Gerharz, M, C. D. Waal, P. Martini, P. James (2003). "Strategies for finding stable paths in mobile wireless ad hoc networks." *Local Computer Networks, LCN '03. Proceedings. 28th Annual IEEE International Conference* pp: 130- 139.
- Jiang S., D. He, and J. Rao (2001). "A prediction-based link availability estimation for mobile ad hoc networks." In *Proc. IEEE INFOCOM*, pages 1745-1752. IEEE.
- Johnson, D. B. and D. A. Maltz (1996). "Dynamic Source Routing in ad hoc wireless networks," *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter5, pp. 153-181. Kluwer Academic Publishers.
- Jubin, J. and J. Tornow (1987). "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, Vol.75, No. 1, pp. 21-32.
- Perkins, C. E. and P. Bhagwat (1994). "Highly dynamic Destination Sequenced Distance-Vector routing (DSDV) for mobile computers," in *Proc. of the SIGCOMM '94 Conf. on Communications Architecture, Protocols and Applications*, pp.234-244.
- Perkins, C. E., E. M. Royer and S. R. Das (1999). "Ad Hoc On-Demand Distance Vector (AODV) Routing," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100.
- Rajagopalan S. and C. C. Shen (2006). "ANSI: A swarm intelligence-based unicast routing protocol for hybrid ad hoc networks," *Volume 52, Issues 8-9, "Journal of Systems Architecture"*, pp 485-504.
- Sim K. M., and W. H. Sun (2003). "Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions." *IEEE Transactions on systems, Man, and Cybernetics-Part A: Systems and Humans*, VOL. 33, NO. 5.
- Tham C. K., S. Marwaha and D. Srinivasan (2002). "Mobile agents based routing protocol for mobile ad hoc networks," in *Proceedings of IEEE Globecom*.

- Toh C. K. (1997). "Associativity based routing for ad hoc mobile networks." *Wireless Personal Communications Journal*, Special Issue on Mobile Networking and Computing Systems, 4(2):103-139.
- Wu Z., H.T. Song, X.M. Xu, S.F. Jiang (2007). "Ant-Based Stable Disjoint Path Routing Algorithm in MANETs," *Pervasive Computing and Applications*, 2007. ICPCA 2007. 2nd International Conference," pp297-302.
- Ziane S. and A. Mellouk (2005). "A Swarm Intelligent Scheme for Routing in Mobile Ad Hoc Networks," *ICW*, pp. 2-6, 2005, *Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05)*.

Section IV: Intelligent Systems and Antennas

Chapter 14

Support Vector Regression for the Modeling and Synthesis of Antenna Arrays

Rafael González-Ayestarán and Fernando Las-Heras

14.1 INTRODUCTION

One of the most important issues to address in radiocommunication system design is the signal-to-noise ratio (SNR) in the received signals. This parameter is strongly related to the performance of the radiating systems and their ability to concentrate energy in the receiver discarding other unwanted signals or noise. Such ability is increased through the proper design of the radiation pattern of the antennas.

Although different radiation properties can be obtained through an adequate design of an antenna, probably the most effective way to control the radiation pattern is the use of a combination of antennas (array) (Balanis, 1981) working as a single radiating system. The global behavior of the antenna array is determined by the superposition of the individual radiation patterns corresponding to each element. The global radiation pattern can be controlled by applying different ‘feeding values’ to each individual radiating element. The problem of calculating the feeding values that must be applied to each element of an array in order to obtain a global radiation pattern complying with a set of specifications is referred to as *synthesis*, while the problem of determining the radiation pattern corresponding to a given set of feeding values applied to the array is referred to as *analysis*.

Accurate analysis of array antennas can be performed through the use of efficient numerical tools able to account for the real properties of the array, such

as coupling effects or the presence of passive elements in a near environment. However, traditional synthesis methods do not take into account these real conditions, assuming a certain degree of error in the solution that could suffice in some applications but may not be acceptable in many others. That approximation is used because of the complexity involved in developing accurate models able to take into account all the real effects involved in antenna arrays and the difficulty involved in including these real effects in synthesis schemes.

An interesting alternative is to include all the real properties of the antenna in the models that take part in the synthesis schemes is the use of data-learning methods. Data-learning methods (also known as knowledge-discovery methods) are based on the search for the relationship existing between different sets of available data. Typically these data are input values applied to the system to be modeled and the corresponding output values. A data-learning method is able to extract from the data a model of the transformation, function or mapping inherently embedded in by the system. This type of technique may be considered to be between the branches of artificial intelligence, signal processing or optimization. The book *Neural Networks (NN)* (Haykin, 1994) is probably the best known representative of this family of techniques.

Support vector machines (SVM) (Gunn, 1998) are powerful state-of-the-art data-learning techniques based on the structural risk minimization (SRM) principle. They are able to solve binary classification problems and have been the subject of considerable study in recent years. Applications of SVMs vary from classification problems (Cortes, Vapnik, 1995) to channel equalization (Mitchinson, Harrison, 2002; Santamaría, Ayestarán et al., 2003). More recently, support vector theory has been extended to the domain of regression problems (Schölkopf, Smola, 2002), also based on the SRM principle, becoming a powerful base from which to solve a range of different problems. These include filter design (Santamaría, 2003), component modeling or direction-of-arrival estimation (Pastorino, Randazzo, 2005), which have been solved and the results have been excellent due to outcomes such as improved generalization capabilities and in input-output knowledge discovery. This chapter is concerned with the use of support vector regression (SVR) to develop accurate models of the antennas accounting for the coupling effects or any other feature that could affect the radiation pattern.

Support vector regression is based on the discovery of the relationship between inputs and outputs, which implies the need to use previously calculated or measured data for the purpose of training the system. Provided that an analysis

tool able to account for real antenna conditions is available, the training patterns can be obtained by applying different sets of ‘feeding values’ and calculating the corresponding radiations patterns. These training patterns can then be used to perform the regression and obtain an accurate model of the antenna. Once this model is developed, it can be included in different synthesis schemes. Of course, measured data, if available, could lead to much more accurate models.

14.2 SUPPORT VECTOR REGRESSION FOR THE MODELING OF ANTENNA ARRAYS

Let $\vec{v}$ be a column vector containing the M feeding values applied to the elements of an antenna array placed along the x -axis, $\vec{v} = [v_1, v_2, \dots, v_N]^T$, $v_i \in C$; $(\cdot)^T$ denotes the transpose. The value of the co-polar component of the radiation pattern generated by the antenna in the direction given by the $\{\theta, \varphi\}$ -coordinates at a given distance R_0 from its center, can be expressed as:

$$E(\theta, \varphi) = \vec{v}^T \vec{g}(\theta, \varphi) \quad (1)$$

where $\vec{g}(\theta, \varphi) = [g_1(\theta, \varphi), g_2(\theta, \varphi), \dots, g_M(\theta, \varphi)]^T$, and $g_i(\theta, \varphi) \in C$ is a term that, in some sense, indicates the influence of the i -th radiant element in the direction $\{\theta, \varphi\}$. In that way, the vector $\vec{g}(\theta, \varphi)$ describes the relation between the applied array voltages and the radiation pattern in the direction $\{\theta, \varphi\}$. Then, for each φ -plane, a matrix $\vec{G}(\varphi) = [\vec{g}(\theta_1, \varphi), \vec{g}(\theta_2, \varphi), \dots, \vec{g}(\theta_N, \varphi)]$ can be defined, allowing us to calculate the field radiated by the structure in the directions θ_i , $i = 1 \dots N$, being N the number of θ -planes, when the voltage set is applied, given as

$$\vec{e}^T(\varphi) = \vec{v}^T \vec{G}(\varphi) \quad (2)$$

where $\vec{e}(\varphi) = [E(\theta_1, \varphi), E(\theta_2, \varphi), \dots, E(\theta_N, \varphi)]^T$. Other φ -planes can be considered to be defining a set of matrices $\vec{G}(\varphi_i)$, $i = 1, \dots, L$, L being the number of considered φ -planes. Then, the matrix model can now be expressed as a 3D matrix $\underline{G}$ with dimensions $M \times N \times L$ defined as the combination of the L matrices with dimensions $M \times N$ corresponding to each φ -plane.

The accurate 3D characterization of the radiating structure consists of the determination of the 3D matrix $\underline{G}$, accounting for all the real properties of the antenna. SVR can be used to find the relationship between the array feeding values and the radiation pattern. Let us consider that P voltage sets and their

corresponding radiation patterns $\{\vec{v}_n, E_n(\theta, \varphi)\}$, $n=1, \dots, P$, are available for training purposes (using an adequate analysis tool, measured in an anechoic chamber or in its operating environment if it is possible). If we focus on a certain direction of the space $\{\theta_i, \varphi_j\}$, our goal is to find the vector $\vec{g}(\theta_i, \varphi_j)$ that fits on Eq.(1) for all the available data pairs with a predefined accuracy. Furthermore, we expect that any other data not included in the available set will also fit such condition with the same range of error (this is called *generalization* and it is one of the main subjects of the Structural Risk Minimization (SRM) principle). The SRM principle establishes that $\vec{g}(\theta_i, \varphi_j)$ can be obtained through the minimization of the following cost function (Gunn, 1998; Schölkopf, Smola, 2002):

$$J(\vec{g}(\theta_i, \varphi_j)) = \frac{1}{2} \|\vec{g}(\theta_i, \varphi_j)\|^2 + C \sum_{n=1}^P |E_n(\theta_i, \varphi_j) - \vec{g}(\theta_i, \varphi_j)^T \vec{v}_n|_{\varepsilon} \quad (3)$$

where

$$|E_n(\theta, \varphi) - \vec{g}(\theta, \varphi)^T \vec{v}_n|_{\varepsilon} = \max(0, |E_n(\theta, \varphi) - \vec{g}(\theta, \varphi)^T \vec{v}_n| - \varepsilon) \quad (4)$$

is the so-called Vapnik's ε -insensitive loss function, and $C > 0$ is a penalty value which establishes a tradeoff between the generalization ability of the model (according to the SRM theory, the first term in (3) guarantees generalization) and the cost of deviations larger than ε (the second term only penalizes such errors). The P training patterns are used for the regression, but it should be noticed that the available data may be noisy, or the assumed model (1) could not be correct, so that the proposed solution may not be feasible. This fact is handled by introducing a set of positive slack variables ξ_n and $\tilde{\xi}_n$ represents the value of deviations from the " ε -tube" defined for Vapnik's loss function around the model. Then, the minimization of (3) can be rewritten as the constrained optimization problem of minimizing

$$J(\vec{g}(\theta_i, \varphi_j), \xi, \tilde{\xi}) = \frac{1}{2} \|\vec{g}(\theta_i, \varphi_j)\|^2 + C \sum_{n=1}^P (\xi_n + \tilde{\xi}_n) \quad (5)$$

subject to

$$\vec{g}(\theta_i, \varphi_j)^T \vec{v}_n - E_n(\theta_i, \varphi_j) \leq \varepsilon + \xi_n \quad (6)$$

$$E_n(\theta_i, \varphi_j) - \vec{g}(\theta_i, \varphi_j)^T \vec{v}_n \leq \varepsilon + \tilde{\xi}_n \quad (7)$$

$$\xi_n + \tilde{\xi}_n \geq 0 \quad (8)$$

for all $n = 1, \dots, P$.

The Lagrange multiplier technique (Arfken, 1985) leads to the optimal regressor (i.e., the optimal values of $\vec{g}(\theta_i, \varphi_j)$), given by

$$\vec{g}(\theta_i, \varphi_j) = \sum_{n=1}^P (\tilde{\alpha}_n - \alpha_n) \vec{v}_n \quad (9)$$

where $\tilde{\alpha}_n$ and α_n are positive Lagrange multipliers calculated by maximizing the dual problem given by the following quadratic expression:

$$\begin{aligned} W(\alpha, \tilde{\alpha}) = & -\sum_{n=1}^P \varepsilon(\tilde{\alpha}_n + \alpha_n) + \sum_{n=1}^P E_n(\theta_i, \varphi_j)(\tilde{\alpha}_n + \alpha_n) \\ & - \frac{1}{2} \sum_{n,m=1}^P (\tilde{\alpha}_n + \alpha_n)(\tilde{\alpha}_m + \alpha_m) \vec{v}_n^T \vec{v}_m \end{aligned} \quad (10)$$

subject to $0 \leq \tilde{\alpha}_n, \alpha_n \leq C$, for all $n = 1, \dots, P$. This is a convex quadratic programming (QP) scenario and, therefore, has a globally optimal solution that can be efficiently found, for example using (Gunn, 1998). According to support vector theory, the input patterns (voltage sets) that appear in the expansion (9) are points where exactly one of the Lagrange multipliers is greater than zero, and the difference $(\tilde{\alpha}_n - \alpha_n)$ does not vanish. These input patterns that contribute to the solution (typically less than P) are called *support vectors*.

Repeating this regression for each of the considered $N \cdot L$ directions of the space, an estimation of the matrix $\vec{G}$ is obtained. If the training patterns have been obtained by including the real antenna characteristics (mutual coupling, presence of parasitic elements, etc.) the resulting model will also include them when representing the behavior of the antenna.

After the matrix $\vec{G}$ has been calculated, the field radiated by the modeled antenna, when a specific voltage set is applied to it, can be estimated using the expression

$$\vec{e}^T = \vec{v}^T \vec{G} \quad (11)$$

where $\vec{e} = [\vec{e}(\varphi_1), \vec{e}(\varphi_2), \dots, \vec{e}(\varphi_L)]$ is a matrix with dimensions $N \times L$ resulting from the product between a vector and a 3D-matrix, and contains the samples of the radiated field in all the considered directions of the space.

As a first approach to the synthesis problem, if a radiation pattern is somehow specified in terms of its amplitude and its phase, and assuming that it is feasible for the array being modeled, it is possible to calculate the corresponding voltage set using the expression

$$\vec{v}^T = \vec{e}^T \vec{G}^{-1} \quad (12)$$

where $\vec{G}^{-1}$ is named the pseudoinverse 3D-matrix of $\vec{G}$ and is defined as the 3D-matrix combination of the pseudoinverse matrices $\vec{G}^{-1}(\varphi_i)$, $i = 1 \dots L$.

14.3 MULTI-REGRESSION AND EQUIVALENT 2D MODELING OF 3D PROBLEMS

The procedure described in section 2 allows the construction of a 3D-matrix model of the behavior of an antenna array able to relate the voltages (or any feeding values) applied to the elements of the array and the corresponding radiated field distribution. However, the problem associated to the use of Eq.(9) is the need to perform $N \cdot L$ independent regressions, that is, as many as the number of directions considered in the space. While such numbers could be acceptable in many 2D problems, typically in 3D problems the number of directions of the space makes it necessary to consider an alternative method which is able to reduce the computational cost associated with the implementation of multiple regressions.

A first approach to the reduction of the number of regressions could be given by the use of support vector multi-regression (SVMR) (Sánchez, Prado, et al., 2004), a non-conventional approximation to SVR in which the regression leads to an optimal matrix instead of an optimal vector. According to SVMR, the problem of obtaining the model that relates inputs and outputs in an antenna array can be formulated as the minimization of the following cost function

$$J(\vec{G}(\varphi_j)) = \frac{1}{2} \sum_{i=1}^N \|\vec{g}_i(\varphi_j)\|^2 + C \sum_{n=1}^P L_\varepsilon(u_i) \quad (13)$$

where

$$L_\varepsilon(u) = \begin{cases} 0, & u < \varepsilon \\ u^2 - 2u\varepsilon + \varepsilon^2, & u \geq \varepsilon \end{cases} \quad (14)$$

and $u_n = \|\vec{e}_n^T(\varphi_j) - \vec{v}_n^T \vec{G}(\varphi_j)\|$,
 $\vec{e}_n(\varphi_j) = [E_n(\theta_1, \varphi_j), E_n(\theta_2, \varphi_j), \dots, E_n(\theta_N, \varphi_j)]^T$, being $\vec{g}_i(\varphi_j)$ the i -th column of the matrix model $\vec{G}(\varphi_j)$, i.e. $\vec{g}_i(\varphi_j) = \vec{g}(\theta_i, \varphi_j)$.

It is important to notice that the use of the cost function expressed in (13) is an extension of Vapnik's ε -insensitive loss function to multiple dimensions, but based on the L_2 norm instead of the L_1 norm. A direct extension of the L_1 -based function would result in the need to account for each dimension (θ -plane) independently, leading to an independent regression for each column without any improvement over the basic SVR. The use of a function based on the L_2 -norm allows consideration of all the directions of the space in a unique constraint yielding a single support vector regression for all the directions (Sánchez, Prado et al., 2004). However, this cost function penalizes the deviation of the training data from the linear model according to a quadratic value for the data that lay out of the ε -tube defined by the standard Vapnik's formulation, smoothing the penalization for certain data. This fact could result in a solution that could differ from the optimal linear model from the SRM principle point of view. The most important consequence of this fact is that more training data could be necessary to approximate the solution to the optimal linear model with the same degree of accuracy. In some sense, the use of this multi-regression procedure represents a tradeoff between the reduction of the number of regressions (that means time and computational cost) and the need for more data or the loss of accuracy.

Assuming that we are able to find an optimal point of tradeoff between cost and accuracy, this first approach allows reduction of the number of regressions from $N \cdot L$ to L , as each θ -plane 2D model can be regressed at once. Furthermore, this idea opens the way to a second approach which is able to obtain a full 3D model in only one regression.

Let us focus again on the 3D-matrix model of an antenna array, $\vec{G}$. According to Eq.(11), the samples of the radiated field distribution in a 3D domain can be expressed as a matrix that can be calculated by multiplying the voltages applied to the array by the 3D-matrix model. It is straightforward to see that such a relationship can also be expressed using a 2D-matrix relationship. Let us define an extended matrix $\tilde{G} = [\vec{G}(\varphi_1), \vec{G}(\varphi_2), \dots, \vec{G}(\varphi_L)]$ as the combination of all the φ -plane matrix models or the decomposition of the 3D-matrix model

$\tilde{G}$ into a 2D-matrix. Let us consider also an extended field of samples vector $\tilde{\vec{e}} = [\vec{e}^T(\varphi_1), \vec{e}^T(\varphi_2), \dots, \vec{e}^T(\varphi_L)]^T$, also let us combine all the φ -plane field samples vectors, now in a column vector instead of a matrix as in Eq.(11). Then, it is obvious that

$$\tilde{\vec{e}}^T = \vec{v}^T \tilde{G} \quad (15)$$

This equation represents an equivalent 2D-matrix formulation for the proposed 3D problem. The subject of the array modeling can now be considered in respect of the matrix $\tilde{G}$, and, according to the SVMR formulation, it can be obtained by performing only one multi-regression using the following cost function:

$$J(\tilde{G}) = \frac{1}{2} \sum_{i=1}^{N \cdot L} \|\tilde{g}_i\|^2 + C \sum_{n=1}^P L_\varepsilon(u_i) \quad (16)$$

where $L_\varepsilon(u)$ is the same function previously defined, but now $u_n = \|\tilde{\vec{e}}_n - \vec{v}_n^T \tilde{G}\|$, $\tilde{\vec{e}}_n$ is the extended field samples vector corresponding to the n -th training pattern, and $\tilde{g}_i$ is the i -th column of the matrix model $\tilde{G}$.

Using this formulation, the problem of obtaining an accurate 3D model of the behavior of an antenna array can be reduced to performing one regression, reducing notably the computational cost and the corresponding time until a suitable model is found. However, if reducing the problem to L regressions led to less accurate models (or to the need for more training data), in this case the problem is even more significant, and much more training data are necessary to obtain a model with the same accuracy as using $N \cdot L$ regressions. In non-rigorous terms of information, we can consider that we are trying to obtain the same model but we are using a simpler process that has been found to lead to less accurate results. The reduced complexity should be compensated for somehow. In this case, it can be compensated for by using more data to increase the presence of information in the process.

As we did in section 2, let us consider a basic 3D synthesis problem in which a radiation pattern is specified in terms of its amplitude and its phase in different (3D) positions of the space. Assuming that such radiation pattern could be synthesized with the modeled array, it is possible to calculate the corresponding voltage set using the expression

$$\vec{v}^T = \tilde{\vec{e}}^T \tilde{G}^{-1} \quad (17)$$

where $\tilde{G}^{-1}$ is the pseudoinverse matrix of $\tilde{G}$.

14.4 ANTENNA ARRAY SYNTHESIS USING SVR-BASED MODELS

In sections 2 and 3 SVR has been introduced to develop different models of an antenna array depending on the dimensionality of the problem: 2D, 3D or an equivalent 2D representation. In any case, at this point an accurate model of the behavior of an array is available and specified as a matrix or a 3D-matrix. Eq.(12) and (17) provide a means to obtain the voltages that must be applied to the ports of the array in order to obtain a certain radiation pattern specified in terms of its amplitude and phase. However, this is not the usual way to specify a target radiation pattern.

Typically, information about the phase distribution of the radiation pattern is not available or is irrelevant for synthesis. Under such circumstances, a unique relationship between the feeding values and the radiation pattern specified only in terms of its amplitude is not guaranteed, and Eq.(12) and (17) cannot be considered suitable for obtaining the voltages corresponding to a specified radiation pattern.

In most practical applications, specifications for synthesis problems are not given as a deterministic radiation pattern but as a set of parameters that can be translated into a template or a mask defining the maximum and minimum value of the radiation pattern amplitude at each aspect angle. In these cases, the details about the shape of the field distribution are not as relevant as certain parameters such as the directivity, side lobes level, main lobe direction, null positions, ripple, etc. Under such circumstances, the degrees of freedom in the problem increase notably. The proposed matrix models are not able to handle so many degrees of freedom, as there is not a unique function to be modeled or approximated.

Apparently, these limitations do not allow us to solve the synthesis problems directly if they are not specified in terms of the amplitude and phase of desired radiation patterns, but the simplicity of the matrix models allows us to include them in more sophisticated schemes which are able to solve synthesis problems with specifications given in different ways.

14.4.1 Synthesis from field amplitude data

When a radiation pattern is specified in terms of its desired amplitude but its phase is not specified or known, the relationship between voltages and radiated

field given in Eq.(1) cannot be considered valid. In fact, the uniqueness in the relationship between voltages and field amplitude is not strictly guaranteed.

Due to the fact that the information is incomplete, additional constraints or information must be included in the synthesis process. Such additional information can be related to the real properties of the radiating structure. Depending on these real properties, the antenna will be able to radiate according to different (infinite) field distributions, but only a certain subspace of radiated field distributions will be suitable for a real antenna array. Including this information in the synthesis scheme could allow retrieval of a coherent phase distribution according to the properties of the specified radiating structure. Fortunately, if the antenna matrix model has been properly calculated, it contains all the information about the real radiating properties of the antenna, including the subspace of possible radiated field distributions.

Using the matrix model to relate field samples and voltages and to include information about the possible field distributions means using the model at least twice. Intuitively, a recurrent use of the matrix model to reconstruct a phase distribution and a set of voltages implies the implementation of an iterative scheme.

An iterative scheme is proposed to reconstruct a coherent phase distribution according to the specified field amplitude and the radiating structure. The proposed process consists of the use of the model of the radiating structure to reduce the distance between the present field phase distribution and another phase distribution feasible in the radiating structure along with the specified field amplitude. Let $|\vec{e}| = (|E(d_1)|, |E(d_2)|, \dots, |E(d_N)|)^T$ be a vector containing the desired radiation pattern specified in amplitude along N directions of the space d_i , $i = 1, \dots, N$. The algorithm to calculate the voltages begins by considering a null initial phase distribution together with the specified radiation pattern amplitude, generating an extended radiation pattern. Through Eq.(12), a first approximation to the necessary voltages is obtained. This approximation is used in q.(2) to obtain the corresponding radiation pattern, whose amplitude is compared with the specified one. If the error does not comply with a predefined stop criterion, the phase of the reconstructed radiation pattern is used together with the specified amplitude, generating another extended radiation pattern used in a new iteration to obtain a new approximation to the voltages. This process is graphically represented in Figure 1.

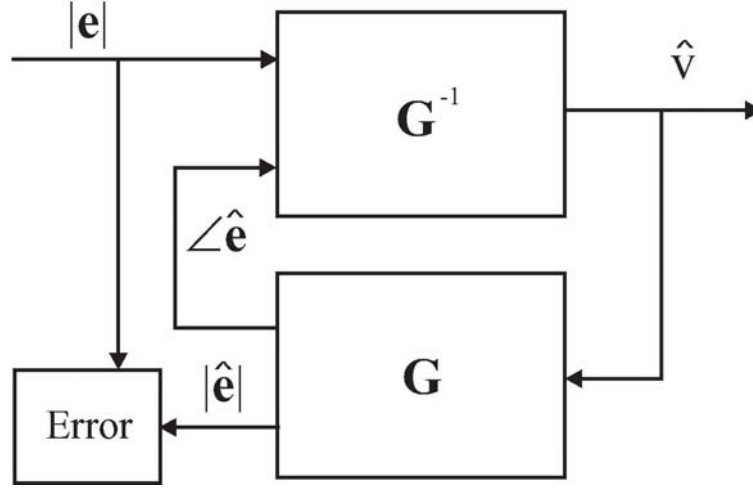


Figure 1: Iterative scheme for synthesis from field amplitude.

It is important to notice that the use of the pseudoinverse matrix of the model $\bar{G}$, provided that this model is accurate, guarantees a minimum mean square error solution. In this case this fact means that the voltages calculated in each iteration of the method are the best possible voltages for the radiating structure being studied and the present extended field distribution (specified amplitude and present phase), and they correspond to a radiated field distribution with less than or equal to the distance to the solution to that of the present extended field distribution.

This iterative method is a very good example of how the matrix models can be easily included in different schemes without any increase of the complexity. Due to the fact that the model includes all the real properties of the antenna array, this simple synthesis scheme will also take them into account leading to an accurate solution.

14.4.2 Synthesis from template-based specifications

In most practical applications, specifications for synthesis problems are not given as a target radiation pattern, but as a set of parameters that can be translated into a template defining the maximum and minimum value of the radiation pattern amplitude at each aspect angle, $B_{max}(\theta, \varphi)$ and $B_{min}(\theta, \varphi)$ respectively. This problem is equivalent to considering a target field amplitude $M(\theta, \varphi)$ defined as

$$M(\theta, \varphi) = \frac{B_{\max}(\theta, \varphi) + B_{\min}(\theta, \varphi)}{2} \quad (18)$$

and allows a maximum deviation of $\varepsilon(\theta, \varphi)$, also defined as

$$\varepsilon(\theta, \varphi) = B_{\max}(\theta, \varphi) - M(\theta, \varphi) = M(\theta, \varphi) - B_{\min}(\theta, \varphi) \quad (19)$$

i.e., $\varepsilon(\theta, \varphi)$ represents the distance between the mean allowed value of the field amplitude and any of the bounds.

The penalization of values laying outside of the template can then be done through the use of Vapnik's ε -insensitive loss function (4), that adopts a form that makes it possible to adopt the SVR problem to the resolution of this kind of problem:

$$|M(\theta, \varphi) - \vec{g}(\theta, \varphi)^T \vec{v}|_{\varepsilon(\theta, \varphi)} = \max\{0, |M(\theta, \varphi) - \vec{g}(\theta, \varphi)^T \vec{v}| - \varepsilon(\theta, \varphi)\} \quad (20)$$

where $\vec{g}(\theta, \varphi)$ is the column of the matrix model $\vec{G}$ corresponding to the direction of the space defined by $\{\theta, \varphi\}$.

Notice that in this case the matrix model is known and the voltage set is unknown and is, therefore, the subject of our problem. Although the SRM principle is not strictly necessary in this case, the statement of the problem using Vapnik's function establishes the following cost function to be minimized:

$$J(\vec{v}) = \frac{1}{2} \|\vec{v}\|^2 + C \sum_{i=1}^N \sum_{j=1}^L |M(\theta_i, \varphi_j) - \vec{v}^T \vec{g}(\theta_i, \varphi_j)|_{\varepsilon(\theta_i, \varphi_j)} \quad (21)$$

where N is the considered number of θ -directions of the space, and L is the number of φ -directions of the space. The second term of the cost function will try to provide a radiation pattern equal to $M(\theta, \varphi)$ with a maximum error of $\varepsilon(\theta, \varphi)$, guaranteeing that the field obtained will comply with the specified template. On the other hand, the first term is not strictly necessary in this problem, as we are trying to obtain a final solution, not a model with generalization capabilities. However, it is still interesting to include it because it penalizes solutions with large norm leading to a minimum energy/power set of vectors, also an important issue in communication systems design.

It is also important to notice that the "fictitious field" $M(\theta, \varphi)$ is only specified in terms of amplitude. As developed in (Santamaría, Vía et al., 2004), field amplitude-only data can be expressed through the L_2 -norm as

$$\|M(\theta, \varphi)\|^2 = \|\vec{v}^T \vec{g}(\theta, \varphi)\|^2 \quad (22)$$

Taking into account that $\vec{v}$, $\vec{g}(\theta, \varphi)$ and $M(\theta, \varphi)$ are, in general, complex values, E(22) can be written as

$$\|\vec{v}^T \vec{g}(\theta, \varphi)\|^2 = \text{Re}\{\vec{v}^T \vec{g}(\theta, \varphi)\}^2 + \text{Im}\{\vec{v}^T \vec{g}(\theta, \varphi)\}^2 \quad (23)$$

This expression can also be rewritten as

$$\begin{aligned} \|\vec{v}^T \vec{g}(\theta, \varphi)\|^2 = & \text{Re}\{M(\theta, \varphi)\} \begin{bmatrix} \text{Re}\{\vec{v}^T\} \text{Im}\{\vec{v}^T\} \end{bmatrix} \begin{bmatrix} \text{Re}\{\vec{g}(\theta, \varphi)\} (24) \\ -\text{Im}\{\vec{g}(\theta, \varphi)\} (25) \end{bmatrix} \\ & + \text{Im}\{M(\theta, \varphi)\} \begin{bmatrix} \text{Re}\{\vec{v}^T\} \text{Im}\{\vec{v}^T\} \end{bmatrix} \begin{bmatrix} \text{Im}\{\vec{g}(\theta, \varphi)\} (26) \\ \text{Re}\{\vec{g}(\theta, \varphi)\} (27) \end{bmatrix} \end{aligned} \quad (28)$$

or, in a more compact formulation, as

$$\|\vec{v}^T \vec{g}(\theta, \varphi)\|^2 = \tilde{\vec{v}}^T \tilde{\vec{g}}(\theta, \varphi) \quad (29)$$

where the following vectors have been defined:

$$\tilde{\vec{v}} = \begin{bmatrix} \text{Re}\{\vec{v}\} \\ \text{Im}\{\vec{v}\} \end{bmatrix} \quad (30)$$

and

$$\begin{aligned} \tilde{\vec{g}}(\theta, \varphi) = & \text{Re}\{M(\theta, \varphi)\} \begin{bmatrix} \text{Re}\{\vec{g}(\theta, \varphi)\} (31) \\ -\text{Im}\{\vec{g}(\theta, \varphi)\} (32) \end{bmatrix} + \text{Im}\{M(\theta, \varphi)\} \begin{bmatrix} \text{Im}\{\vec{g}(\theta, \varphi)\} (33) \\ \text{Re}\{\vec{g}(\theta, \varphi)\} (34) \end{bmatrix} \\ = & \begin{bmatrix} \text{Re}\{M(\theta, \varphi) \vec{g}^*(\theta, \varphi)\} \\ \text{Im}\{M(\theta, \varphi) \vec{g}^*(\theta, \varphi)\} \end{bmatrix} \end{aligned} \quad (35)$$

where the superscript $(.)^*$ denotes complex conjugate. Since $M(\theta, \varphi)$ is a function of template values, for this particular problem $\text{Im}\{M(\theta, \varphi)\} = 0$. Now our cost function can be expressed, in order to deal with the fact that phase specifications are not available, as:

$$J(\tilde{\vec{v}}) = \frac{1}{2} \|\tilde{\vec{v}}\|^2 + C \sum_{i=1}^N \sum_{j=1}^L \left| \tilde{M}(\theta_i, \varphi_j) - \tilde{\vec{v}}^T \tilde{\vec{g}}(\theta_i, \varphi_j) \right|_{\tilde{\varepsilon}(\theta_i, \varphi_j)} \quad (36)$$

where $\tilde{\varepsilon}(\theta, \varphi) = \varepsilon^2(\theta, \varphi)$ and $\tilde{M}(\theta, \varphi) = \|M(\theta, \varphi)\|^2$. As was done in Eq.(5), two positive slack variables, ξ and $\tilde{\xi}$ are introduced in order to deal with the fact that a solution completely inside the template could not be feasible in the radiating structure considered, and some deviations could occur. Then, (36) becomes

$$J(\tilde{\mathbf{v}}, \xi, \tilde{\xi}) = \frac{1}{2} \|\tilde{\mathbf{v}}\|^2 + C \sum_{i=1}^N \sum_{j=1}^L (\xi_{ij} + \tilde{\xi}_{ij}) \quad (37)$$

subject to

$$\tilde{\mathbf{v}}^T \tilde{\mathbf{g}}(\theta_i, \varphi_j) - \tilde{M}(\theta_i, \varphi_j) \leq \tilde{\varepsilon}(\theta_i, \varphi_j) + \xi_{ij} \quad (38)$$

$$\tilde{M}(\theta_i, \varphi_j) - \tilde{\mathbf{v}}^T \tilde{\mathbf{g}}(\theta_i, \varphi_j) \leq \tilde{\varepsilon}(\theta_i, \varphi_j) + \tilde{\xi}_{ij} \quad (39)$$

$$\xi_{ij}, \tilde{\xi}_{ij} \geq 0 \quad (40)$$

The optimal set of feeding values can then be derived as

$$\tilde{\mathbf{v}} = \sum_{i=1}^N \sum_{j=1}^L (\tilde{\alpha}_{ij} - \alpha_{ij}) M(\theta_i, \varphi_j) \tilde{\mathbf{g}}^*(\theta_i, \varphi_j) \quad (41)$$

where $\tilde{\alpha}_{ij}$ and α_{ij} are the positive Lagrange multipliers obtained in this case by solving the dual optimization problem of maximizing the expression

$$\begin{aligned} W(\tilde{\alpha}, \alpha) = & - \sum_{i=1}^N \sum_{j=1}^L \tilde{\varepsilon}(\theta_i, \varphi_j) (\tilde{\alpha}_{ij} + \alpha_{ij}) + \sum_{i=1}^N \sum_{j=1}^L \tilde{M}(\theta_i, \varphi_j) (\tilde{\alpha}_{ij} + \alpha_{ij}) \\ & - \frac{1}{2} \sum_{i,k=1}^N \sum_{j,r=1}^L (\tilde{\alpha}_{ij} + \alpha_{ij}) (\tilde{\alpha}_{kr} + \alpha_{kr}) \tilde{\mathbf{g}}^T(\theta_i, \varphi_j) \tilde{\mathbf{g}}(\theta_k, \varphi_r) \end{aligned} \quad (42)$$

subject to $0 \leq \tilde{\alpha}_{ij}, \alpha_{ij} \leq C$.

If we consider again the case of a radiation pattern specified in terms of its amplitude, we can define a “tube” of tolerated deviation ε around it. Then, if we focus in a direction of the space given by $\{\theta, \varphi\}$, we have just defined a template where $M(\theta, \varphi)$ is equal to the desired amplitude $|E(\theta, \varphi)|$, and the maximum and minimum bounds are $B_{max}(\theta, \varphi) = |E(\theta, \varphi)| + \varepsilon$ and

$B_{min}(\theta, \varphi) = |E(\theta, \varphi)| - \varepsilon$. In this way, the template-case formulation is also valid for the amplitude-only case, accepting a maximum deviation of ε at each aspect angle. Furthermore, ε could be set to 0 if it is necessary.

According to this last idea, the template-case formulation can be considered a unified framework to deal with any kind of specifications for the synthesis problem by choosing the target $M(\theta, \varphi)$ appropriately and the value of the maximum deviation.

14.5 EXAMPLES AND PRACTICAL ISSUES

In this section we will show how the SVR framework can be used to solve different synthesis problems. For the sake of simplicity, we present only simulated experiments where the training data have been obtained using accurate simulation tools and considering simulated radiating structures over which we solve synthesis problems. However, experimental data obtained from measurements held in an anechoic chamber, or in a non-ideal environment, could be used if they were available.

In the examples presented we have considered a 16 half-wavelength collinear dipole array, with elements placed along the x -axis and with centers uniformly separated by 0.7λ , λ being the wavelength. In the first place, we will obtain the model of the antenna using SVR, that implies obtaining a set of training patterns. A set of 50 vectors of random voltages have been generated and applied to the array, obtaining the co-polar component of the radiation pattern at 91 directions of the space angularly equispaced between $\theta = -90^\circ$ and $\theta = 90^\circ$ and at 91 directions also angularly equispaced between $\varphi = 0^\circ$ and $\varphi = 180^\circ$, using a full-wave analysis tool (Numerical Electromagnetic Code (NEC), Part II, 1977) based on the Method of Moments (MoM) (Harrington, R. F., 1968) and wire modeling. The parameters for the multi-regression have been chosen as $C = 1$ and $\varepsilon = 0$, hence resulting in the 3D-matrix model $\tilde{\underline{G}}$. The maximum deviation ε has been set to 0 because working with simulations we guarantee that our data are not noisy. Then, the available data should fit perfectly the linear model and, as we are looking for an accurate model, no deviation must be allowed. If our data were obtained through measurement, ε should be set to a non-zero value related with the estimated variance of the measurement noise.

The 3D-matrix model (or the equivalent 2D model) is used in the following examples to solve different synthesis problems. In the first example, a nominal voltage distribution with hamming window-type amplitude and linear phase is applied to the antenna, obtaining the corresponding 2D radiation pattern for the plane $\varphi = 0$, plotted in Figure 2. The amplitude of this radiation pattern, considered as the target pattern, is proposed as input to the amplitude-only synthesis iterative method proposed in section 4.1 considering only the 2D-matrix model corresponding to the plane considered. The reconstructed voltages are obtained after 20 iterations and represented in Table 1, while the radiation pattern that they generate (synthesized pattern) is also plotted in Figure 2. It is worth mentioning that the coupling effects between the elements of the array have been taken into account through the use of the calculated matrix model $\bar{G}(\theta, \varphi = 0)$ because an analysis tool able to account for them has been used to obtain the training pattern. The maximum differences between the nominal and synthesized radiation patterns are lower than 0.07dB at -40dB, and lower than 0.9dB at -50dB. A solution obtained using a synthesis method based on minimization of a cost function was not enough to account for the influence of coupling effects (Las-Heras, Galocha et al., 1997) is used to calculate the corresponding radiation pattern (considering coupling effects) also represented in Figure 2.

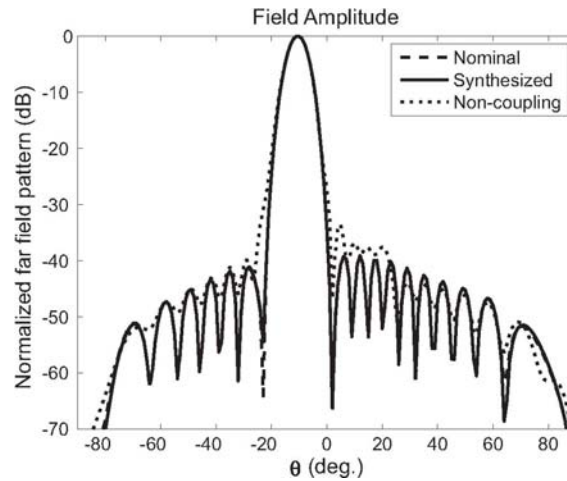


Figure 2: Nominal and synthesized radiation pattern.

Table 1: Calculated feeding values.

Element	Amplitude (V)	Phase (deg.)
1	0.0847	123.1
2	0.1258	171.0
3	0.2459	-139.6
4	0.4183	-94.4
5	0.6072	-48.1
6	0.7923	-2.2
7	0.9261	43.9
8	10.000	90.2
9	0.9920	135.9
10	0.9047	-178.6
11	0.7534	-133.2
12	0.5735	-87.2
13	0.3841	-41.9
14	0.2229	4.4
15	0.1170	46.1
16	0.0749	90.1

In these results we can appreciate how the accuracy of the model makes it possible to obtain a solution that fits the specification without significant differences, while a method based on models that do not account for the real properties of the antenna obtains a solution that differs from the specified pattern when the voltages obtained are applied to a coupling-element array.

In the second example, the same model as before is used to solve another synthesis problem using the same antenna. The target radiation pattern (also in the plane $\varphi = 0$) is specified by a set of templates determining the required maximum and minimum radiation pattern values at each aspect angle, as represented in Figure 3. Using the SVR-based method proposed in section 4.2 with $C=1$, a set of voltages is obtained. These voltages, represented in Table 2, generate a radiation pattern, also plotted in Figure 3, which perfectly complies with the specifications. A solution calculated with the non-coupling coupling method proposed in (Las-Heras, Galocha et al., 1997) is analyzed using the MoM-based tool resulting in the radiation pattern also plotted in Figure 3.

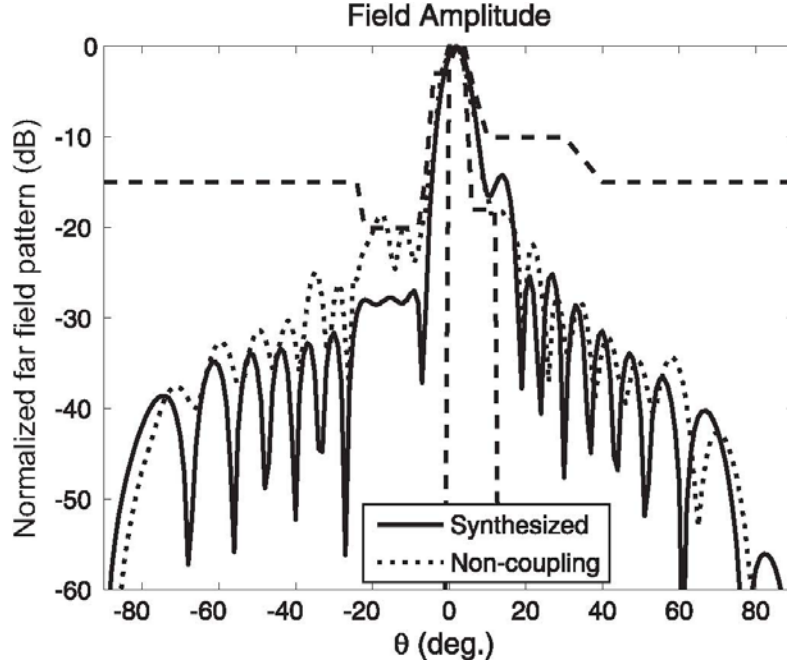


Figure 3: Template and synthesized radiation pattern.

In the following example the previously used collinear dipoles have been replaced by 16 half-wavelength parallel dipoles, also parallel to the y -axis, with their centers separated by 0.7λ and distributed along the x -axis, in the same positions as the previously studied collinear dipoles. A 3D model of a radiating structure has also been developed using 75 sets of random voltages that are applied to the array, obtaining the co-polar component of the radiation pattern in all the considered directions of the space using a full-wave analysis tool based on the Method of Moments. The radiation patterns have been obtained considering 91 directions of the space angularly equispaced between $\theta = -90^\circ$ and $\theta = 90^\circ$, and 181 φ -constant planes corresponding to equispaced values between $\varphi = 0^\circ$ and $\varphi = 180^\circ$. Once the 3D matrix model $\bar{\underline{G}}$ has been obtained, it is evaluated in both analysis and synthesis cases. A set of voltages following again a hamming window-type with linear phase distribution is applied to the structure (Table 3). Figure 4 represents the corresponding radiation pattern (co-polar

Table 2: Calculated feeding values.

Element	Amplitude (V)	Phase (deg.)
1	0.1628	-152.8
2	0.2935	146.6
3	0.3357	139.4
4	0.3820	140.1
5	0.5319	139.2
6	0.7512	138.9
7	0.9247	134.3
8	0.9855	120.5
9	10.000	101.4
10	0.9475	87.7
11	0.7559	82.4
12	0.5274	80.8
13	0.3909	79.0
14	0.3338	81.3
15	0.2575	79.2
16	0.2076	27.1

component) calculated both using the full-wave analysis tool based on the MoM and wire modeling (nominal radiation pattern), and using the proposed SVR model to evaluate its analysis accuracy. The maximum differences between both patterns are lower than 0.6dB at -40dB and lower than 2dB at -50dB.

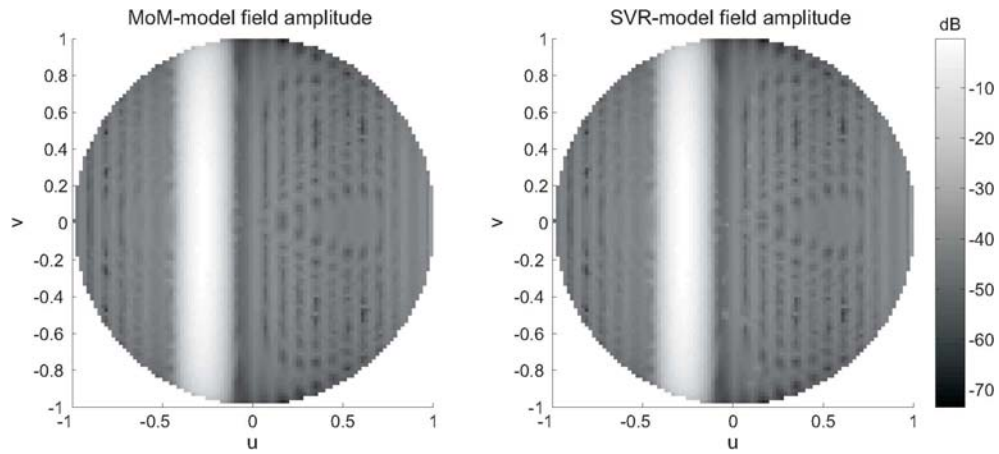
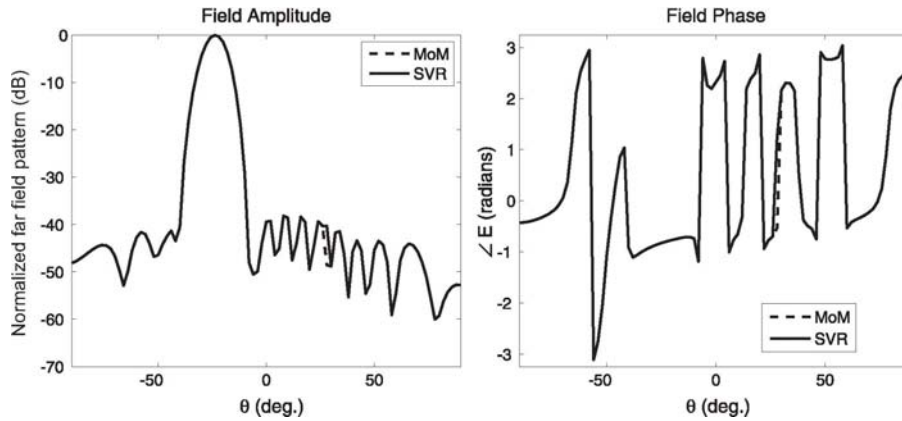


Figure 4: Radiation u-v pattern (co-polar component) using MoM and SVR models.

Table 3: Calculated feeding values.

Element	Nominal	Synthesized
1	0.080 $\angle 74.4^\circ$	0.082 $\angle 75.4^\circ$
2	0.121 $\angle 148.9^\circ$	0.123 $\angle 149.7^\circ$
3	0.234 $\angle -136.5^\circ$	0.234 $\angle -136.4^\circ$
4	0.401 $\angle -62.0^\circ$	0.403 $\angle -62.1^\circ$
5	0.594 $\angle 12.4^\circ$	0.591 $\angle 12.5^\circ$
6	0.777 $\angle 86.9^\circ$	0.774 $\angle 86.7^\circ$
7	0.921 $\angle 161.3^\circ$	0.918 $\angle 161.2^\circ$
8	1.000 $\angle -124.1^\circ$	0.995 $\angle -124.2^\circ$
9	1.000 $\angle -49.6^\circ$	1.000 $\angle -49.6^\circ$
10	0.921 $\angle 24.8^\circ$	0.921 $\angle 24.9^\circ$
11	0.777 $\angle 99.3^\circ$	0.779 $\angle 99.3^\circ$
12	0.594 $\angle 173.8^\circ$	0.591 $\angle 174.1^\circ$
13	0.401 $\angle -111.7^\circ$	0.404 $\angle -111.7^\circ$
14	0.234 $\angle -37.2^\circ$	0.237 $\angle -36.3^\circ$
15	0.121 $\angle 37.2^\circ$	0.120 $\angle 37.0^\circ$
16	0.080 $\angle 111.7^\circ$	0.081 $\angle 111.2^\circ$

Figure 5: Radiation pattern (co-polar component) at $\varphi = 45^\circ$ using MoM and SVR models.

In Figure 5, the radiation pattern in the plane $\varphi = 45^\circ$ is represented. No significant differences can be seen between the two distributions, the maximum differences being lower than 0.007dB at -40dB.

The 3D radiation pattern calculated, making use of the MoM model is specified for the synthesis method presented in section 4.1. The set of voltages calculated after 12 iterations is represented in Table 3. The value of the mean square error between the nominal and the synthesized voltage distributions is $5 \cdot 10^{-6}$.

All the previous examples considered were working in far field conditions. In fact, in all the equations used there is a missing coordinate: the distance between the antenna and the point where the radiated field is evaluated. We have assumed that far field conditions make the radiation pattern independent of the distance. However, this assumption could not be true if far field conditions are not satisfied. Including a distance coordinate in the formulation is almost straightforward and it would mean increasing in one dimension the problems. If in a 3D case we have to account for $N \cdot L$ directions of the space, in this 4D case $N \cdot L \cdot K$ positions of the space must be considered, K being the number of distances where the radiated field is evaluated. In most practical applications there is no need to consider more than one distance, but in many cases such distance do not satisfy far field conditions, being necessary to predefine the domain where the radiated field is evaluated.

In the next example, a near field (NF) model of an antenna array is necessary to solve a NF-synthesis problem at a distance of 1.5λ . The 16 half-wavelength collinear dipole array modeled in the first two examples is considered again, but in this case we need a new model of the antenna able to represent its behavior at such distance. A set of 150 voltage vectors is randomly generated, and they are used to calculate the E_θ component of the corresponding NF distribution at the distance of 1.5λ , and at 91 directions in space angularly equispaced between $\theta = -90^\circ$ and $\theta = 90^\circ$ using our full-wave analysis tool based on the MoM. In this case, the multi-regression with parameters $C=1$ and $\varepsilon=0$ leads to the elements of the matrix $\tilde{G}$, in this case a 2D-matrix with dimensions 16×91 . A nominal near field distribution (Figure 6) is specified. The amplitude of this distribution provides the amplitude-only synthesis iterative method. The synthesized near field distribution after 20 iterations is also plotted in Figure 6. As in the far field case, no differences can be seen between the two distributions.

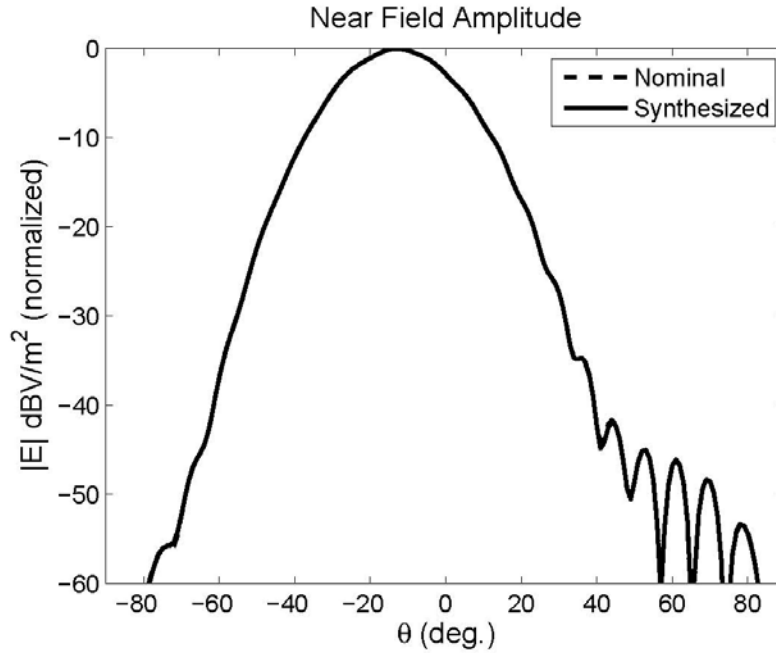


Figure 6: Nominal and synthesized NF distributions.

14.6 FINAL COMMENTS

Although all the formulation related to the radiated field throughout the chapter does not mention any reference to field components, it has been assumed implicitly that all the components satisfy the relationship expressed in Eq.(1). This fact allows us to obtain a model of the behavior of an antenna in terms of the relationship between feeding values and the field component of interest using the same formulation. The only difference can be found in the training data to be used, as the radiated field values must correspond to the field component for which the model is being developed.

Another point of interest is given by the fact that the presented formulation does not make any reference to the actual positions of the elements of the array. Although all the examples are proposed using uniform arrays, there is no need to consider a constant distance between the radiating elements. Furthermore, there

is no need to consider equal elements. The models are obtained from input and output data, so no assumption is made about the geometry of the radiating structure.

Finally, it is also interesting to note that, independently of the use of single regressions or multi-regression, each direction of the space is treated separately and simply represents one column of the matrix (or 3D-matrix) models. This fact means that it is not necessary to consider equispaced directions of the space, and non-uniform sampling of the field distribution could be used.

The use of Support Vector Regression represents a flexible and powerful tool for the modeling and synthesis of array antennas able to account for the real properties of the radiating structure and its environment. Probably the next challenge to make use of its capabilities is improving our own ability to obtain proper training data through measurements of our real systems.

References

- Arfken, G. (1985). Lagrange Multipliers. In: *Mathematical Methods for Physicists*, Academic Press, Orlando.
- Balanis, C. A. (1981). *Antenna Theory*. John Wiley and Sons, New York (1981)
- Cortes, C., Vapnik, V. (1995). Support Vector Networks, *Machine Learning*, **20**(3), 321–354.
- Gunn, S. R. (1998). *MATLAB Support Vector Machine Toolbox*, Image Speech and Intelligent Systems Research Group, University of Southampton
- Harrington, R. F. (1968). *Field Computation by Moment Methods*. Macmillan Co., New York
- Haykin, S. (1994). *Neural Networks, A Comprehensive Foundation*, Macmillan Co., New York
- Las-Heras, F., Galocha, B., Besada, J. L. (1997). Equivalent Source Modeling and Reconstruction for Antenna Measurement and Synthesis, *IEEE Antennas and Propagation Society International Symposium*, Montreal .
- Mitchinson, B., Harrison, R.F. , (2002). Digital Communications Channel Equalization Using the Kernel Adaline, *IEEE Transactions on Communications*, **50**(4), 571–576.
- Numerical Electromagnetic Code (NEC), Part II (1977):NEC Program Description & Part III: NEC User's Guide, Lawrence Livermore Laboratory.

- Pastorino, M., Randazzo, A. (2005). A smart antenna system for direction of arrival estimation based on a support vector regression, *IEEE Transactions on Antennas and Propagation*, **53**(7), 2161–2168.
- Sánchez, M., Prado, M., Arenas, J., Pérez-Cruz, J. (2004). SVM Multiregression for Nonlinear Channel Estimation in Multiple-Input Multiple-Output Systems, *IEEE Transactions on Signal Processing*, **52**(8), 2298–2307.
- Santamaría, I. (2003). Design of Linear-Phase FIR Filters Using Support Vector Regression Approach, *Electronics Letters*, **39**(19), 1422–1423.
- Santamaría, I., Ayestarán, R. G., Pantaleón, C., Principe, J. C. (2003). Maximum Margin Equalizers Trained with the Adatron Algorithm, *Signal Processing*, **83**, 593–602.
- Santamaría, I., Vía, J., Merino, J. (2004). SVM-based blind beamforming of constant modulus signals, *International Joint Conference on Neural Networks (IJCNN, Budapest)*.
- Schölkopf, B., Smola, A. J. (2002). *Learning with Kernels*, MIT Press, Cambridge.
- Vapnik, V. (1995): *The Nature of Statistical Learning Theory*, Springer Verlag, New York.

Chapter 15

Support Vector Machines for the Estimation of the Directions of Arrival of Electromagnetic Waves

Andrea Randazzo and Matteo Pastorino

15.1 INTRODUCTION

New wireless communication systems strongly require the use of smart antennas, i.e. antennas able to modify their behaviors (in particular, their radiation properties) depending on certain external conditions (Winters 1998; Liberti and Rappaport 1999; Paulraj et al. 2000; Dandekar et al. 2002; Blostein and Leib 2003; Choi et al. 2003; Sarkar et al. 2003). Often, the receiving antenna modifies its behavior depending on the basis of the location of the (possibly moving) transmitting system. Moreover, the determination of the incident angles of waves impinging on receiving antennas is of fundamental importance not only for the development of smart antennas for wireless communications, but also in many applications related to radar, sonar, and other detection apparatuses for security, imaging, etc (Chandran 2006; Martínez-Ramón et al. 2007). The efficiency of the smart antenna is strongly related to the effectiveness of the system in precisely estimating the directions of arrival (DOAs) of the incident electromagnetic signals, which are captured by the antenna in its receiving mode.

Consequently, the problem of estimating the DOAs of electromagnetic waves constitutes in general a prerequisite for the development of smart antennas and it has been widely studied in recent years. Several efficient approaches have been devised and tested. Among them, the multiple signal classification (MUSIC) algorithm (Schmidt 1986), the maximum likelihood (ML) method, and the estimation of signal parameters via rotational invariance technique (ESPRIT) (Roy and Kailath 1989) are commonly used methods. Moreover, the use of

neural network has been also successfully exploited (El Zooghby et al. 1997, 2000). Approaches relying on combined spatial and temporal processing have also been proposed (Kyungjung et al. 2001).

Recently, a new class of methods for solving the DOA estimation problems based on the use of support vector machines (SVMs) (David Sanchez 2003) has been successfully introduced. The main advantage of this approach is related to the excellent generalization capabilities (i.e., the ability of the system to determine incident angles not included in a fixed training set) that SVMs have shown in several applications, including antenna synthesis (Martínez-Ramón and Christodoulou 2006), pattern recognition (Vapnik and Lerner 1963), classification (Guo and Li 2003), time prediction (Wu et al. 2003), and regression problems (Vapnik et al. 1997).

The present authors proposed in (Pastorino and Randazzo 2005, 2006) a new algorithm, in which the estimated DOAs are obtained by using a nonlinear support vector regression (SVR) approach (Smola and Schölkopf 2004). In the proposed approach, the function that relates the angles of incidence of the electromagnetic waves to the outputs of the receiving antennas is approximated by using a set of SVMs, which are trained starting from several samples of the input/output values of this mapping. After the definition of the internal parameters of the SVMs, the DOA estimation is performed in a fast and accurate way during the test phase.

In the present chapter, the mathematical formulation of the proposed estimation algorithm is discussed. Furthermore, some relevant results recently obtained by using the SVM-based approach are reviewed. Comparisons with other existing estimation techniques and experimental validations are also reported and discussed. Moreover, although the DOA estimation methods are usually evaluated by considering waves with propagation vectors lying in the horizontal plane, the SVM-based estimator is also assessed in the case of electromagnetic waves with arbitrary propagation directions. Finally, the capabilities of the approach when dealing with reflected electromagnetic waves in near field conditions are evaluated, too.

The chapter is organized as follows. In Section 2, the mathematical basis of the DOA estimation approach are reported, whereas in Section 3 the SVM-based estimation method is briefly outlined. Finally, numerical and experimental results are reported and discussed in Section 4.

15.2 PROBLEM STATEMENT

Let us consider an uniform linear array composed by N receiving elements located at positions uniformly spaced along the x -axis, i.e.,

$$\mathbf{r}_n = x_n \hat{\mathbf{x}} = (n-1)d\hat{\mathbf{x}}, \quad n=1, \dots, N \quad (1)$$

being d the inter-element spacing (Figure 1). M narrowband electromagnetic waves with central frequency ω impinge on the array from directions (ϕ_m, θ_m) , $m = 1, \dots, M$.

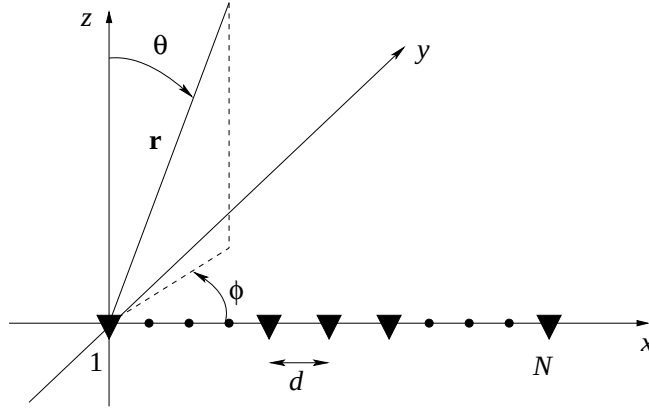


Figure 1: Configuration of the smart antenna

The electromagnetic field is a linear combination of the contributions of the M waves, i.e.,

$$\mathbf{E}(\mathbf{r}) = \sum_{m=1}^M \mathbf{E}_m(\mathbf{r}) \quad (2)$$

where $\mathbf{E}_m(\mathbf{r})$, $m = 1, \dots, M$, denotes the electromagnetic field due to the m -th incident wave. If the electromagnetic sources are located in the far-field region of the smart antenna, the received electromagnetic waves can be modeled as plane waves, i.e.,

$$\mathbf{E}_m(\mathbf{r}) = \mathbf{e}_m e^{-j\mathbf{k}_m \cdot \mathbf{r}} \quad (3)$$

being $\mathbf{e}_m$, $m = 1, \dots, M$, the complex amplitude of the m -th plane wave and $\mathbf{k}_m = -k \sin \theta_m \cos \phi_m \hat{\mathbf{x}} - k \sin \theta_m \sin \phi_m \hat{\mathbf{y}} - k \cos \theta_m \hat{\mathbf{z}}$, where $k = \omega(\epsilon_0 \mu_0)^{1/2}$ is the

free-space wavenumber. The far field assumption will be always considered in the training phase of the estimation algorithm, whereas, in the test phase, near-field conditions can be assumed.

The open-circuit output voltage of the n -th antenna is given by

$$v_n = \mathbf{E}(\mathbf{r}_n) \cdot \mathbf{l}_e^n + n_n = \sum_{m=1}^M \mathbf{E}_m(\mathbf{r}_n) \cdot \mathbf{l}_e^n + n_n \quad (4)$$

where $\mathbf{l}_e^n$ is the effective length of the n -th element and n_n , $n = 1, \dots, N$, is a Gaussian noise (assumed with zero mean value and variance σ^2). If the receiving elements are equal, it results

$$\mathbf{l}_e^1 = \mathbf{l}_e^2 = \dots = \mathbf{l}_e^N = \mathbf{l} \quad (5)$$

As an example, if the receiving elements are dipoles of length $h = \lambda/2$ and directed along the z axis, it results (Collin 1985)

$$\mathbf{l} = \frac{\lambda}{\pi} \left(\frac{\cos(\pi/2 \cos \theta)}{\sin \theta} \right) \hat{\boldsymbol{\theta}} \quad (6)$$

being λ the central wavelength corresponding to ω and $\hat{\boldsymbol{\theta}}$ the unit vector denoting the θ direction.

Under far field conditions, since $\mathbf{e}_m = e_m^\theta \hat{\boldsymbol{\theta}} + e_m^\phi \hat{\boldsymbol{\phi}}$, where $\hat{\boldsymbol{\phi}}$ is the unit vector of the ϕ coordinate, it results

$$v_n = \sum_{m=1}^M \frac{\lambda}{\pi} e_m^\theta \left(\frac{\cos(\pi/2 \cos \theta_m)}{\sin \theta_m} \right) e^{-j(n-1)kd \sin \theta_m \cos \phi_m} + n_n \quad (7)$$

In the scientific literature it is commonly assumed that the electric field is z -polarized and the propagation is limited to the x - y plane (i.e., $\theta_m = \pi/2$). In that case, Eq.(7) can be simplified and the open-circuit voltage at the output of the n -th antenna can be written as

$$v_n = \sum_{m=1}^M s_m e^{-j(n-1)kd \cos \phi_m} + n_n \quad (8)$$

being $s_m = \lambda e_m^\theta / \pi$. The above assumption is always made during the training phase of the considered estimation approach, whereas, in some examples, it will be removed in the test phase.

By using a matrix notation, Eq.(8) is rewritten as

$$\mathbf{v} = \mathbf{A}\mathbf{s} + \mathbf{n} \quad (9)$$

where

$$\begin{aligned}
 \mathbf{v} &= [v_1, v_2, \dots, v_N]^t \\
 \mathbf{n} &= [n_1, n_2, \dots, n_N]^t \\
 \mathbf{s} &= [s_1, s_2, \dots, s_M]^t \\
 \mathbf{A} &= \begin{bmatrix} 1 & \dots & 1 \\ e^{-jkd \cos \phi_1} & & e^{-jkd \cos \phi_M} \\ \dots & \dots & \dots \\ e^{-j(N-1)kd \cos \phi_1} & \dots & e^{-j(N-1)kd \cos \phi_M} \end{bmatrix}
 \end{aligned} \tag{10}$$

and the superscript t denotes the transpose of the array. Starting from Eq.(9), the spatial correlation matrix is computed (El Zooghby et al. 1997, 2000), i.e.

$$\mathbf{R} = E\{\mathbf{v}\mathbf{v}^h\} = \mathbf{A}\mathbf{P}\mathbf{A}^h + \sigma^2\mathbf{I} \tag{11}$$

where $\mathbf{P}$ is a $M \times M$ matrix given by $\mathbf{P} = E\{\mathbf{s}\mathbf{s}^*\}$, $\mathbf{I}$ is a $N \times N$ unit matrix, and the superscript h indicates the complex conjugate transpose. Since the correlation matrix $\mathbf{R}$ is hermitian, only the upper triangular part (composed by $N(N+1)/2$ complex elements) is considered. These matrix elements are rearranged in an array $\mathbf{b}$, given by

$$\mathbf{b} = [r_{11}, r_{12}, \dots, r_{1N}, r_{22}, \dots, r_{2N}, \dots, r_{(N-1)(N-1)}, r_{(N-1)N}, r_{NN}]^t \tag{12}$$

where $r_{mn} = [\mathbf{R}]_{mn}$, $m, n = 1, \dots, N$. Moreover, the array $\mathbf{b}$ is also normalized and the new array $\mathbf{z} = \mathbf{b}/\|\mathbf{b}\|$ will be used to estimate the DOAs.

Finally, the problem of estimating the DOAs of the M incident waves can be formulated as the identification of the unknown inverse function $\mathbf{F}$ which relates the angles ϕ_m , $m = 1, \dots, M$, to the array $\mathbf{z}$. The m -th component of $\mathbf{F}$, denoted as F_m , can be estimated by using a regression approach, in which an approximation $\hat{F}_m$ to F_m is constructed starting from the knowledge of several input/output pairs $(\mathbf{z}, \phi_m)$.

15.3 NONLINEAR SVM ESTIMATOR

The inverse functions F_m , $m = 1, \dots, M$, can be successfully approximated by using a SVR. Let us first consider the case in which $M = 1$, for which the regression problem can be stated as follows. Let ϕ_1 be the actual incidence direction of the impinging wave. The input/output pairs used to train the SVM are constructed by considering L fixed angles $\hat{\phi}_1^l$, $l = 0, \dots, L-1$ and by calculating (using the equations described in the previous Section) the corresponding

normalized array output vectors $\hat{\mathbf{z}}^l$, $l = 0, \dots, L-1$. Thus, the training set Ω_{train}^l is given by

$$\Omega_{train}^l = \left\{ (\hat{\mathbf{z}}^0, \hat{\phi}_1^0), \dots, (\hat{\mathbf{z}}^{L-1}, \hat{\phi}_1^{L-1}) \right\} \quad (13)$$

The values contained in the training set Ω_{train}^l are used to evaluate the function $\tilde{F}_1$ that approximate the unknown actual function F_1 . In particular, the approximating function is given by

$$\tilde{F}_1(\mathbf{z}) = \langle \mathbf{w}, \varphi(\mathbf{z}) \rangle + b \quad (14)$$

In Eq.(14), φ is a nonlinear function used to transform the input data $\mathbf{z}$ from their original space, namely Σ , to a higher dimensional space H (usually called *feature space*) and $\langle \cdot, \cdot \rangle$ denotes the inner product of H . Moreover, $\mathbf{w}$ and b are parameters whose optimal values are obtained by minimizing the regression risk associated with this SVR, which is given by

$$R_{reg} = C \sum_{l=0}^{L-1} c(\hat{\mathbf{z}}^l, \hat{\phi}_1^l) + \frac{1}{2} \|\mathbf{w}\|^2 \quad (15)$$

where C is a constant and $c(\hat{\mathbf{z}}^l, \hat{\phi}_1^l)$ is the ε -insensitive loss function, i.e.,

$$c(\hat{\mathbf{z}}^l, \hat{\phi}_1^l) = \begin{cases} 0 & \text{if } |\hat{\phi}_1^l - \tilde{F}_1(\hat{\mathbf{z}}^l)| \leq \varepsilon \\ |\hat{\phi}_1^l - \tilde{F}_1(\hat{\mathbf{z}}^l)| - \varepsilon & \text{otherwise} \end{cases} \quad (16)$$

In Eq.(16), ε is the allowable error during the training phase. According to (Vapnik 1998), the minimization of the regression risk is accomplished by using a dual formulation. In particular, a standard dualization method utilizing Lagrange multipliers is employed. The following quadratic optimization problem, to be solved with respect to the Lagrange multipliers α_i and α'_i , is obtained.

$$\begin{aligned} \max \left\{ -\varepsilon \sum_{l=0}^{L-1} (\alpha'_l + \alpha_l) + \sum_{l=0}^{L-1} \hat{\phi}_m^l (\alpha'_l - \alpha_l) - \right. \\ \left. - \frac{1}{2} \sum_{l,i=0}^{L-1} (\alpha'_l + \alpha_l)(\alpha'_i - \alpha_i) \kappa(\hat{\mathbf{z}}^l, \hat{\mathbf{z}}^i) \right\} \end{aligned} \quad (17)$$

subject to

$$\begin{aligned} \sum_{l=0}^{L-1} (\alpha'_l - \alpha_l) &= 0 \\ \alpha'_l, \alpha_l &\in [0, C] \end{aligned} \quad (18)$$

The parameter C has been found to control the generalization properties of the SVM, i.e., the capabilities of the approach in estimating DOAs not included in the training set (Smola and Schölkopf 2004). In particular, high values of C produce small estimation errors for the DOAs contained in the training set, but they result in low generalization capabilities.

In Eq.(17), κ is the kernel function, working on the original space Σ and defined as

$$\kappa(\hat{\mathbf{z}}^l, \mathbf{z}) = \langle \varphi(\hat{\mathbf{z}}^l), \varphi(\mathbf{z}) \rangle \quad (19)$$

Commonly used kernels are polynomial and Gaussian kernels (Smola and Schölkopf 2004). For the DOA estimation problem, it has been found that the Gaussian kernel, defined as

$$\kappa(\hat{\mathbf{z}}^l, \mathbf{z}) = e^{-\gamma \|\mathbf{z} - \hat{\mathbf{z}}^l\|^2} \quad (20)$$

where γ is a constant, produces good estimation results (Pastorino and Randazzo 2005).

Using the dual optimization problem, $\mathbf{w}$ can be written in terms of the input data $\mathbf{z}$ as

$$\mathbf{w} = \sum_{l=0}^{L-1} (\alpha_l - \alpha'_l) \varphi(\hat{\mathbf{z}}^l) \quad (21)$$

By substituting (21) in (14), $\tilde{F}_1$ is rewritten as

$$\begin{aligned} \tilde{F}_1(\mathbf{z}) &= \sum_{l=0}^{L-1} (\alpha_l - \alpha'_l) \langle \varphi(\hat{\mathbf{z}}^l), \varphi(\mathbf{z}) \rangle + b \\ &= \sum_{l=0}^{L-1} (\alpha_l - \alpha'_l) \kappa(\hat{\mathbf{z}}^l, \mathbf{z}) + b \end{aligned} \quad (22)$$

The quadratic optimization problem in Eq.(17)-(18) can be efficiently solved by using, for example, the Sequential Minimal Optimization (SMO) algorithm (Keerthi et al. 2001). It is worth noting that only a subset of the Lagrange multipliers will have a nonzero value (those satisfying the condition $|\hat{\phi}_1^i - \tilde{F}_1(\mathbf{z}^i)| \geq \varepsilon$), thus a sparse solution is obtained. The samples associated with the nonzero Lagrange multipliers are called support vectors (Vapnik 1998). Consequently, only the support vectors contribute to the sum in equation (22).

In the case in which $M > 1$, M SVMs are used, and, after the definition of a proper training set, each one is trained by using the same procedure described above. Let ϕ_m , $m = 1, \dots, M$, be the actual incident DOAs. Moreover, let $\Delta \phi^{m(m+1)} = \phi_{m+1} - \phi_m$,

$m = 1, \dots, M-1$, be the actual angular separations between two adjacent incident angles (clearly, there are $M-1$ angular separations). In order to correctly estimate the DOAs of the M impinging waves it is necessary to consider, in the training phase, incident waves with several angular separations. In particular, Q fixed angular separations, denoted as $\Delta\hat{\phi}_q^{m(m+1)}$, $q = 1, \dots, Q$, are used in the learning phase. The training set for the m -th SVM is then constructed as follows. For each angular separation, L_q training samples of the M incident angles, denoted as $\hat{\phi}_m^{l_q}$, $l_q = 0, \dots, L_q-1$, $q = 1, \dots, Q$, $m = 1, \dots, M$, are chosen. By using these training angles, a set of $L = \sum_{q=1}^Q L_q$ values of $\hat{\mathbf{z}}^l$, $l = 0, \dots, L-1$, is calculated by using the equations previously reported. The training set for the m -th SVM, Ω_{train}^m , $m = 1, \dots, M$, is then given by

$$\Omega_{train}^m = \left\{ (\mathbf{z}^0, \hat{\phi}_m^0), \dots, (\mathbf{z}^{L-1}, \hat{\phi}_m^{L-1}) \right\} \quad (23)$$

By using the same procedure described for the case $M = 1$, an approximation of the function $\tilde{F}_m$ is obtained by using SVR theory, and it is used to compute the estimate $\tilde{\phi}_m$ of the m -th angle ϕ_m .

15.4 OVERVIEW OF NUMERICAL AND EXPERIMENTAL RESULTS

The proposed approach has been numerically and experimentally evaluated in several papers. A summary of the results obtained is provided in this Section.

The first reported results concerned the field of wireless communications, in which the received waves are assumed to be generated by remote sources under far-field radiation conditions. Moreover, in most cases, the incident waves are assumed to propagate in a horizontal plane, and the receiving array lies on this plane. In this case, each incident wave is only determined by the azimuth angle ϕ .

The numerical analysis has been performed both by considering one or more incident waves, with different noisy environments, usually represented by additive Gaussian noise. Moreover, the effects of the choice of different key parameters for the SVM implementation have also been studied.

Following the notation used in (Pastorino and Randazzo 2005), the symbols $\hat{\eta}$ and $\bar{\eta}$ denote the generic quantity η (an incident angle or an angular separation) in the training and test phases, respectively, whereas $\tilde{\eta}$ denotes the estimated value of η . Accordingly, the estimation errors are defined as

$$\tau_m^l = \left| \tilde{\phi}_m^l - \bar{\phi}_m^l \right| \quad (24)$$

where m and l denote the m -th wave and the l -th sample, respectively. For each wave, the mean error is given by

$$\tau_m = \frac{1}{L} \sum_{l=0}^{L-1} \tau_m^l \quad (25)$$

where L is the number of samples. Finally, the mean value of the absolute errors (computed for all the waves of the testing set) is defined as

$$\tau = \frac{1}{M} \sum_{m=1}^M \tau_m = \frac{1}{ML} \sum_{m=1}^M \sum_{l=0}^{L-1} \tau_m^l \quad (26)$$

where M is the number of incident waves.

Table 1: Parameters of simulation number 1.

Number of waves (M)	$M = 2$
Number of angular separations (Q)	$Q = 4$
Angular separations in the training set ($\Delta \hat{\phi}_q^{m(m+1)}$, $q = 1, \dots, Q$, $m = 1, \dots, M-1$)	$\Delta \hat{\phi}_1^{12} = 5^\circ$, $\Delta \hat{\phi}_2^{12} = 10^\circ$, $\Delta \hat{\phi}_3^{12} = 15^\circ$, $\Delta \hat{\phi}_4^{12} = 20^\circ$
Number of training samples for each angular separation (L_q , $q = 1, \dots, Q$)	$L_1 = 176$, $L_2 = 171$, $L_3 = 166$, $L_4 = 161$
Total number of training samples (L)	$L = 674$
DOAs in the training phase ($\hat{\phi}_m^{l_q}$, $m = 1, \dots, M$, $l_q = 0, \dots, L_q - 1$, $q = 1, \dots, Q$)	$\hat{\phi}_1^{l_q} = l_q 1^\circ$ $\hat{\phi}_2^{l_q} = l_q 1^\circ + \Delta \hat{\phi}_q^{12}$
Angular separations in the test set ($\Delta \bar{\phi}^{m(m+1)}$, $m = 1, \dots, M-1$)	$\Delta \bar{\phi}^{12} = 17.5^\circ$
Number of test samples ($\bar{L}$)	$\bar{L} = 162$
DOAs in the test phase ($\bar{\phi}_m^l$, $m = 1, \dots, M$, $l = 0, \dots, \bar{L} - 1$)	$\bar{\phi}_1^l = l 1^\circ$ $\bar{\phi}_2^l = l 1^\circ + \Delta \bar{\phi}^{12}$
Signal-to-noise ratio (SNR)	SNR = 10 dB
SVM parameters (C , γ , ε)	$C = 0.82$, $\gamma = 0.5$, $\varepsilon = 0.098$ (Cherkassky and Ma 2004), $\varepsilon = 0.33$ (Kwok and Tsang 2003)

In the case of two incident waves ($M = 2$) in a noisy environment, results have been obtained in (Pastorino and Randazzo 2005). As an example, for the training set reported in Table 1 (Simulation number 1), very accurate estimation have been obtained for signal-to-noise ratio (SNR) ≥ 10 dB, for which τ resulted in less than 0.5° considering both the Kwok and Cherkassky's conditions (Kwok

and Tsang 2003, Cherkassky and Ma 2004). Moreover, the standard deviation has been found to be less than 0.5° for $65^\circ < \phi_1 < 110^\circ$, less than 1° for $10^\circ < \phi_1 < 150^\circ$, whereas it approaches 10° just for few samples near to $\phi_1 = 0^\circ$ and $\phi_1 = 180^\circ - \Delta\bar{\phi}^{12}$. Typically some errors have always been found for the extreme samples, due to an intrinsic ambiguity in the incident directions.

Table 2: Parameters of simulation number 2.

Number of waves (M)	$M = 2$
Number of angular separations (Q)	$Q = 5$
Angular separations in the training set ($\Delta\hat{\phi}_q^{m(m+1)}, q = 1, \dots, Q, m = 1, \dots, M-1$)	$\Delta\hat{\phi}_1^{12} = 5^\circ, \Delta\hat{\phi}_2^{12} = 10^\circ, \Delta\hat{\phi}_3^{12} = 15^\circ,$ $\Delta\hat{\phi}_4^{12} = 20^\circ, \Delta\hat{\phi}_5^{12} = 25^\circ$
Number of training samples for each angular separation ($L_q, q = 1, \dots, Q$)	$L_1 = 351, L_2 = 341, L_3 = 331, L_4 = 321,$ $L_5 = 311$
Total number of training samples (L)	$L = 1655$
DOAs in the training phase ($\hat{\phi}_m^{l_q},$ $m = 1, \dots, M, l_q = 0, \dots, L_q - 1, q = 1, \dots, Q$)	$\hat{\phi}_1^{l_q} = l_q 0.5^\circ$ $\hat{\phi}_2^{l_q} = l_q 0.5^\circ + \Delta\hat{\phi}_q^{12}$
Number of test samples ($\bar{L}$)	$\bar{L} = 3$
DOAs in the test phase ($\bar{\phi}_m^l,$ $m = 1, \dots, M, l = 0, \dots, \bar{L} - 1$)	$\bar{\phi}_1^1 = 90^\circ, \bar{\phi}_2^1 = 100^\circ,$ $\bar{\phi}_1^2 = 90^\circ, \bar{\phi}_2^2 = 110^\circ,$ $\bar{\phi}_1^3 = 100^\circ, \bar{\phi}_2^3 = 110^\circ$
SVM parameters (C, γ, ε)	$C = 10, \gamma = 0.01, \varepsilon = 0.03$

The results above are only concerned with simulated data. However, the capabilities of the approach have also been experimentally evaluated by using measured data obtained at the Applied Microwave Nondestructive Testing Laboratory of the University of Missouri-Rolla (Randazzo et al. 2007). With reference to the configuration reported in Table 2 (Simulation number 2), it has been experimentally proven that the absolute errors in the DOA estimation is less than 5° for a number of receiving elements greater than $N = 19$, whereas for $N \geq 5$ the errors are always less than 10° .

These experiments have been performed by synthesizing the antenna array by using a X-band open-ended rectangular waveguide, whereas another open-ended rectangular waveguide was used as a transmitter. Both the waveguides are connected to a HP8510C vector network analyzer.

Three or more incident waves have been also analyzed. The effects of the noise of the receiving probes have been numerically analyzed with reference to

three incident waves ($M = 3$). The assumed configuration is reported in Table 3 (Simulation number 3). The mean errors τ_m in the estimation of the incoming waves are reported in Table 4 for different values of SNR. As can be seen, the estimation is quite good even in a strongly noisy environment.

Another key question is represented by the quality of the estimation for incident waves that are not included in the training set. In the case of six incident waves ($M = 6$), the results concerning the estimation of incident angles included in the training set are directly compared with those obtained by using values not included in this set. The configuration used is reported in Table 5 (Simulation number 4). Figure 2 reports the DOA estimation errors in the two cases.

Table 3: Parameters of simulation number 3.

Number of waves (M)	$M = 3$
Number of angular separations (Q)	$Q = 4$
Angular separations in the training set ($\Delta\hat{\phi}_q^{m(m+1)}$, $q = 1, \dots, Q$, $m = 1, \dots, M-1$)	$\Delta\hat{\phi}_1^{12} = 2.5^\circ$, $\Delta\hat{\phi}_1^{23} = 5^\circ$, $\Delta\hat{\phi}_2^{12} = 5^\circ$, $\Delta\hat{\phi}_2^{23} = 10^\circ$, $\Delta\hat{\phi}_3^{12} = 10^\circ$, $\Delta\hat{\phi}_3^{23} = 20^\circ$, $\Delta\hat{\phi}_4^{12} = 15^\circ$, $\Delta\hat{\phi}_4^{23} = 30^\circ$
Number of training samples for each angular separation (L_q , $q = 1, \dots, Q$)	$L_1 = 173$, $L_2 = 166$, $L_3 = 151$, $L_4 = 136$
Total number of training samples (L)	$L = 626$
DOAs in the training phase ($\hat{\phi}_m^{l_q}$, $m = 1, \dots, M$, $l_q = 0, \dots, L_q - 1$, $q = 1, \dots, Q$)	$\hat{\phi}_1^{l_q} = l_q 0.5^\circ$ $\hat{\phi}_2^{l_q} = l_q 0.5^\circ + \Delta\hat{\phi}_q^{12}$ $\hat{\phi}_3^{l_q} = l_q 0.5^\circ + \Delta\hat{\phi}_q^{12} + \Delta\hat{\phi}_q^{23}$
Angular separations in the test set ($\Delta\bar{\phi}^{m(m+1)}$, $m = 1, \dots, M-1$)	$\Delta\bar{\phi}^{12} = 12.5^\circ$, $\Delta\bar{\phi}^{23} = 12.5^\circ$
Number of test samples ($\bar{L}$)	$\bar{L} = 142$
DOAs in the test phase ($\bar{\phi}_m^l$, $m = 1, \dots, M$, $l = 0, \dots, \bar{L} - 1$)	$\bar{\phi}_1^l = l 1^\circ$ $\bar{\phi}_2^l = l 1^\circ + \Delta\bar{\phi}^{12}$ $\bar{\phi}_3^l = l 1^\circ + \Delta\bar{\phi}^{12} + \Delta\bar{\phi}^{23}$
Signal-to-noise ratio (SNR)	SNR = ∞ , 20, 10, 5 dB
SVM parameters (C , γ , ε)	$C = 0.82$, $\gamma = 0.5$, $\varepsilon = 0.1$

Table 4: Errors on the estimation of the DOAs of $M = 3$ incoming waves for different values of SNR.

	τ_1	τ_2	τ_3	τ
SNR = ∞	0.32°	0.31°	0.38°	0.34°
SNR = 20 dB	0.33°	0.32°	0.41°	0.35°
SNR = 10 dB	0.64°	0.60°	0.75°	0.66°
SNR = 5 dB	1.83°	1.73°	2.25°	1.93°

Table 5: Parameters of simulation number 4.

Number of waves (M)	$M = 6$
Number of angular separations (Q)	$Q = 4$
Angular separations in the training set ($\Delta\hat{\phi}_q^{m(m+1)}$, $q = 1, \dots, Q$, $m = 1, \dots, M-1$)	$\Delta\hat{\phi}_q^{m(m+1)} = \Delta\hat{\phi}_q$, $m = 1, \dots, M-1$ $\Delta\hat{\phi}_1 = 5^\circ$, $\Delta\hat{\phi}_2 = 7.5^\circ$, $\Delta\hat{\phi}_3 = 10^\circ$, $\Delta\hat{\phi}_4 = 15^\circ$
Number of training samples for each angular separation (L_q , $q = 1, \dots, Q$)	$L_1 = 156$, $L_2 = 143$, $L_3 = 131$, $L_4 = 106$
Total number of training samples (L)	$L = 536$
DOAs in the training phase ($\hat{\phi}_m^{l_q}$, $m = 1, \dots, M$, $l_q = 0, \dots, L_q - 1$, $q = 1, \dots, Q$)	$\hat{\phi}_m^{l_q} = l_q 1^\circ + (m-1) \Delta\hat{\phi}_q$ $\Delta\bar{\phi}^{m(m+1)} = \Delta\bar{\phi}$, $m = 1, \dots, M-1$
Angular separations in the test set ($\Delta\bar{\phi}^{m(m+1)}$, $m = 1, \dots, M-1$)	$\Delta\bar{\phi} = 10^\circ$ (samples belonging to the training set) $\Delta\bar{\phi} = 8.2^\circ$ (samples not belonging to the training set) $\bar{L} = 131$ (samples belonging to the training set)
Number of test samples ($\bar{L}$)	$\bar{L} = 139$ (samples not belonging to the training set)
DOAs in the test phase ($\bar{\phi}_m^l$, $m = 1, \dots, M$, $l = 0, \dots, \bar{L} - 1$)	$\bar{\phi}_m^l = l 1^\circ + (m-1) \Delta\bar{\phi}$
Signal-to-noise ratio (SNR)	SNR = $+\infty$
SVM parameters (C , γ , ε)	$C = 0.82$, $\gamma = 0.5$, $\varepsilon = 0.0$

Concerning the computational aspects of the approach, the use of SVMs results in extremely short evaluations in the test phase, whereas the training phase can be performed once for all and off line. As reported in (Pastorino and

Randazzo 2005), for the case of $M = 2$ electromagnetic waves impinging on an array composed by $N = 6$ elements, the CPU time needed to train each SVM in the learning phase is 12.25 s, whereas, in the test phase, any SVM takes 0.025 s to estimate each DOA value (on a PC equipped with a 2 GHz processor and 512 MB of RAM).

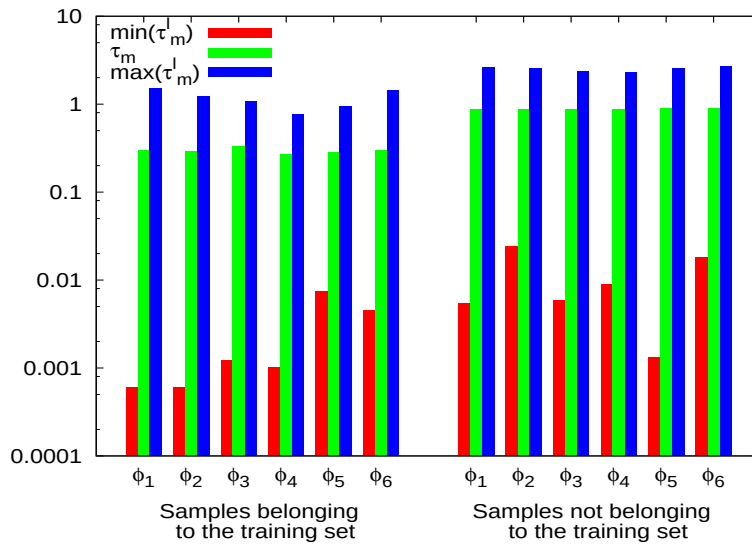


Figure 2: Comparison of the estimation errors (mean, minimum, and maximum errors) for six impinging waves with DOAs contained and not contained in the training set.

Table 6: Parameters of simulation number 5.

Number of waves (M)	$M = 1$
Total number of training samples (L)	$L = 181$
DOAs in the training phase ($\hat{\phi}_1^l, l = 0, \dots, L-1$)	$\hat{\phi}_1^{l_q} = l1^\circ$
Number of test samples ($\bar{L}_\phi, \bar{L}_\theta$)	$\bar{L}_\phi = 234, \bar{L}_\theta = 18$
Total number of test samples ($\bar{L}$)	$\bar{L} = 4212$
DOAs in the test phase ($\bar{\phi}_1^{l_\phi}, l_\phi = 0, \dots, \bar{L}_\phi - 1,$ $\bar{\theta}_1^{l_\theta}, l_\theta = 0, \dots, \bar{L}_\theta - 1$)	$\bar{\phi}_1^{l_\phi} = l_\phi 0.77^\circ$ $\bar{\theta}_1^{l_\theta} = 5^\circ + l_\theta 5^\circ$

The capabilities of the approach have been also evaluated by quantifying the errors due to incident waves whose propagation vectors have also vertical components. Consequently, the incident direction is identified by both θ and ϕ . The chosen parameters are provided in Table 6 (Simulation number 5). The estimated values of the DOAs for some relevant angles are reported in Table 7.

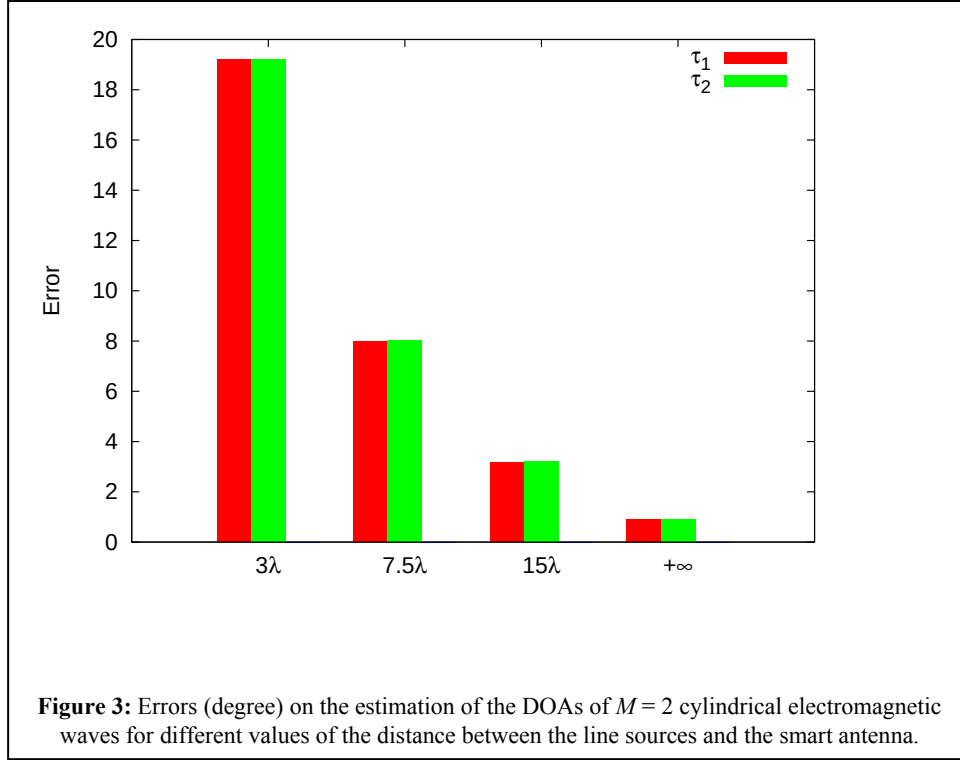
Table 7: Estimated values of the DOAs of $M = 1$ impinging wave having a vertical component in the propagation vector.

	$\theta_1 = 90^\circ$	$\theta_1 = 75^\circ$	$\theta_1 = 55^\circ$	$\theta_1 = 35^\circ$	$\theta_1 = 15^\circ$
$\phi_1 = 90^\circ$	90.04°	90.04°	90.03°	90.02°	90.01°
$\phi_1 = 115^\circ$	113.59°	112.70°	108.91°	102.77°	95.07°
$\phi_1 = 140^\circ$	139.01°	136.69°	127.83°	114.96°	100.29°
$\phi_1 = 165^\circ$	170.41°	159.06°	141.06°	122.43°	103.35°

Table 8: Parameters of simulation number 6.

Number of waves (M)	$M = 2$
Number of angular separations (Q)	$Q = 4$
Angular separations in the training set ($\Delta\hat{\phi}_q^{m(m+1)}$, $q = 1, \dots, Q$, $m = 1, \dots, M-1$)	$\Delta\hat{\phi}_1^{12} = 5^\circ$, $\Delta\hat{\phi}_2^{12} = 10^\circ$, $\Delta\hat{\phi}_3^{12} = 15^\circ$, $\Delta\hat{\phi}_4^{12} = 20^\circ$
Number of training samples for each angular separation (L_q , $q = 1, \dots, Q$)	$L_1 = 176$, $L_2 = 171$, $L_3 = 166$, $L_4 = 161$
Total number of training samples (L)	$L = 674$
DOAs in the training phase ($\hat{\phi}_m^{l_q}$, $m = 1, \dots, M$, $l_q = 0, \dots, L_q$, $q = 1, \dots, Q$)	$\hat{\phi}_1^{l_q} = l_q 1^\circ$ $\hat{\phi}_2^{l_q} = l_q 1^\circ + \Delta\hat{\phi}_q^{12}$
Number of test samples ($\bar{L}$)	$\bar{L} = 181$
DOAs in the test phase ($\bar{\phi}_m^l$, $m = 1, \dots, M$, $l = 0, \dots, \bar{L} - 1$)	$\bar{\phi}_1^l = 11^\circ$ $\bar{\phi}_2^l = 11^\circ + \Delta\bar{\phi}^{12}$
Signal-to-noise ratio (SNR)	SNR = $+\infty$
SVM parameters (C , γ , ε)	$C = 0.82$, $\gamma = 0.5$, $\varepsilon = 0.0$

In order to evaluate the capabilities of the approach in detecting electromagnetic waves generated by sources located in the near-field region of the antenna, the SVMs are still trained by considering plane waves, whereas in the test phase, the incident waves are produced by line currents moderately far from the receiving antenna.



The test samples $(\bar{\mathbf{z}}^l, \bar{\phi}_m^l)$, $l = 0, \dots, L-1$, are obtained by analytically computing the electric field at the elements of the array due to sources located at $\mathbf{r}_m^{lc}$, $m = 1, \dots, M$. In particular, the various line-current sources assume different angular positions along a circumference centered in the middle of the array, in order to keep fixed the distance $|\mathbf{r}_m^{lc}|$ during the construction of the training set. The other data for the configuration are shown in Table 8 (Simulation number 6).

The degradation in the estimation capabilities of the approach due to the “not-plane” nature of the incident waves are evaluated in term of the reconstruction errors previously defined. In particular, Figure 3 reports the errors in the estimation of the incident angles of the two cylindrical waves for different values of $|\mathbf{r}_m^{lc}|$. As can be seen, when $|\mathbf{r}_m^{lc}| > 15\lambda$ the estimation is quite good. However, even in the case in which the electromagnetic sources are very close to the smart antenna, the proposed approach is able to estimate the DOAs, although the errors are higher.

Beside being a computationally efficient DOA estimation approach, the SVM-based method seems to be able to provide very accurate estimations. In fact, some results yielded by this approach have been compared with those provided by the MUSIC method, which is a high resolution method. A very good agreement has been obtained both by using synthetic and experimental data (Pastorino and Randazzo 2005; Abou-Khousa et al. 2006).

15.5 CONCLUSIONS

In this Chapter, the application of Support Vector Machines for the estimation of the directions of arrival of electromagnetic waves has been discussed with reference to smart antennas used both in wireless communications and for localization and investigative purposes. The formulation of the approach has been briefly reported. Moreover, the capabilities of the method, together with details concerning the computational requirements, have been discussed with reference to several numerical and experimental results. It seems that the application of Support Vector Machines represents a very efficient choice and further significant developments are expected as results of the current and further research in these areas.

REFERENCES

- Abou-Khousa MA, Pastorino M, Randazzo A, Zoughi R (2006) An SVM-based approach for AOA estimation: numerical and experimental comparison with the MUSIC algorithm. In: Proc. 2006 Mediterranean Microwave Symposium (MMS'2006).
- Blostein SD and Leib H (2003) Multiple antenna systems: Role and impact in future wireless access. *IEEE Comm Mag* 41:94-101.
- Chandran S (2006) *Advances in Direction-of-Arrival Estimation*. Artech House.
- Cherkassky V, Ma Y (2004) Practical Selection of SVM Parameters and Noise Estimation for SVM Regression. *Neural Networks* 17:113-126.
- Choi MS, Grosskopf G, Rohde D, Kuhlow B, Przyrembel G, Ehlers H (2003) Experiments on

- DOA-estimation and beamforming for 60 GHz smart antennas. In: Proc. Vehicular Technology Conference, 2003 (VTC 2003-Spring), pp. 1041-1045.
- Collin RE (1985) Antennas and Radiowave Propagation. McGraw-Hill.
- Dandekar KR, Ling H, Xu G (2002) Experimental study of mutual coupling compensation in smart antenna applications. *IEEE Trans Wireless Comm* 1:480-487.
- David Sánchez VA (2003) Advanced support vector machines and kernel methods. *Neurocomputing*, Special issue on Support Vector Machines, 55:5-20.
- El Zooghby AH, Christodoulou CG, Georgiopoulos M (1997) Performance of radial-basis function network for direction arrival estimation with antenna arrays. *IEEE Trans Antennas Propagat* 45:1611-1617.
- El Zooghby AH, Christodoulou CG, Georgiopoulos M (2000) A neural network-based smart antenna for multiple source tracking. *IEEE Trans Antennas Propagat* 48:768-776.
- Guo G, Li SZ (2003) Content-Based Audio Classification and Retrieval by Support Vector Machines. *IEEE Trans Neural Networks* 14:209-215.
- Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK (2001) Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation* 13:637-649.
- Kyungjun K, Sarkar TK, Wang H, Salazar-Palma M (1997) Direction of arrival estimation based on temporal and spatial processing using a direct data domain (D3) approach. *IEEE Trans Antennas Propagat* 45:533-541.
- Kwok JT, Tsang IW (2003) Linear dependency between ϵ and the input noise in ϵ -support vector regression. *IEEE Trans Neural Networks* 14:544-553.
- Liberti J, Rappaport TS (1999) Smart antennas for wireless communications: IS95 and third generation CDMA applications. Prentice-Hall.
- Martínez-Ramón M, Christodoulou CG (2006) Support vector machines for antenna array processing and electromagnetics. Morgan and Claypool Publishers.
- Martínez-Ramón M, Rojo-Álvarez JL, Camps-Valls G, Christodoulou CG (2007) Kernel Antenna Array Processing. *IEEE Trans Antennas Propagat* 55:642-650.
- Martínez-Ramón M, Xu N, Christodoulou CG (2007) Antenna array processing for radar applications with support vector machines. In: *Ultra-Wideband Short-Pulse Electromagnetics* 8. Springer, New York, pp. 143-151.
- Pastorino M, Randazzo A (2005) A Smart Antenna System for Direction of Arrival Estimation based on a Support Vector Regression. *IEEE Trans Antennas Propagat* 53:2161-2168.
- Pastorino M, Randazzo A (2006) The SVM-based Smart Antenna for Estimation of the Directions of Arrival of Electromagnetic Waves. *IEEE Trans Instrum Meas* 55:1918-1925.
- Paulraj AJ, Gesbert D, Papadias C (2000) Smart antennas for mobile communications. In: *Encyclopedia for Electrical Engineering*. John Wiley Publishing Co., 2000, pp 1-15.
- Randazzo A, Abou-Khousa MA, Pastorino M, Zoughi R (2007) Direction of Arrival Estimation based on Support Vector Regression: Experimental Validation and Comparison with MUSIC. *IEEE Antennas Wireless Propagat Lett* 6:379-382.
- Roy R, Kailath T (1989) ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques. *IEEE Trans Acoust Speech Signal Processing* 37:984-995.
- Sarkar TK, Salazar-Palma M, Wicks M, Bonneau RJ (2003) Smart antennas. Wiley, New York.
- Schmidt RO (1986) Multiple emitter location and signal parameter estimation. *IEEE Trans Antennas Propagat* 34:276-280.
- Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. *Statistics and Computing* 14:199-222.

- Vapnik V (1998) Statistical Learning Theory. Wiley, New York.
- Vapnik V, Lerner A (1963) Pattern recognition using generalized portrait method. Automation and Remote Control 24:774-780.
- Vapnik V, Golowich S, Smola AJ (1997) Support vector method for function approximation, regression estimation, and signal processing. Neural Information Processing Systems, Cambridge, MA: MIT Press.
- Winters JH (1998) Smart antennas for wireless systems. IEEE Personal Comm 5:23-27.
- Wu CH, Wei CC, Su DC, Chang MH, Ho JM (2003) Travel time prediction with support vector regression. In: Proc. IEEE Intelligent Transportation Systems Conference, pp. 1438-1442.

Chapter 16

Kernel Methods for Smart Antennas

M. Martínez-Ramón, A. Navia-Vázquez, R. Jordán and C. G. Christodoulou

16.1 INTRODUCTION

In many array processing applications in communications it is essential to reach a reasonable performance level, particularly in those environments which change rapidly. For example, in mobile communications, users continuously change in number and allocation. Also, channels may vary due to moving objects. In order to extend the capacity of the channels, beamforming is used to allow users to reuse frequencies. In such situations, adaptive arrays are used for tracking issues.

Linear LMS or RLS are used because of their properties of fast convergence and low computational cost. The limits on their performance are related to the maximum available number K of array elements and the number N of data available for training the array parameters. K is lower bounded by the number of interferences that have to be cancelled. Also, K is upper bounded by many practical reasons, but the one that concerns us here is its associated number of parameters. If it is too large, then, a large N will be needed for a reasonable convergence; with N small, the system may overfit (Haykin, 1996).

These bounds can be extended with kernel algorithms. They consist of a nonlinear transformation of the data to a higher (possibly infinite) dimensional Hilbert space H provided with a dot product which may be expressed as a nonlinear function of the input data (kernel) (Aizerman, Braverman et al., 1964). Then, linear algorithms can be applied to H , which is nonlinear from the point of view of the input data. Their main drawback is that overfitting may occur due to the fact that the complexity of the system will increase the number of data needed for training. Nevertheless, control of complexity can be applied by the use of

Support vector machines (SVMs) (Vapnik, 1995), thus giving nonlinear algorithms which need the same number of data as their linear counterparts -see, e. g., (Martínez-Ramón, Rojo-Álvarez et al., 2007).

The main drawback of standard SVMs is that they are not adaptive. Nevertheless, there exist a set of techniques that use an iterative re-weighted least squares for online training of the machine, plus techniques to online adapt the subspace in which this procedure is applied, thus making possible an adaptive version of these kernel machines (Navia-Vazquez, Pérez-Cruz et al., 2001).

We present here an adaptive SVM, formulated in its complex valued version for antenna array processing. Examples, presented here, cover signal detection and tracking in multiuser nonstationary environments.

16.2 NONLINEAR ARRAY PROCESSING

16.2.1 Linear optimum beamformers

The output of a beamformer is a linear combination of the outputs of the elements which can be expressed as the dot product $\hat{y}[n] = \mathbf{w}^H \mathbf{x}[n]$. If a dataset $\mathbf{x}[n]$ is available for the training of the parameters, the procedure is called beamforming with temporal reference. Minimizing the square error expectation between the desired and obtained outputs gives us the well known MMSE solution

$$\mathbf{w} = \mathbf{R}^{-1} \mathbf{p} \approx (\mathbf{X}\mathbf{X})^{-1} \mathbf{X}\mathbf{y} \quad (1)$$

$\mathbf{R}$ being the autocorrelation matrix of the data matrix $\mathbf{X}$, and $\mathbf{p}$ being the cross correlation vector between the input data and the desired data vector $\mathbf{y}$.

Instead, one may apply a spatial reference, meaning that we want the beamformer to steer to a given direction. Assuming the use of the covariance method of data windowing, it is required to minimize the output energy

$$\begin{aligned} E &= E[|\hat{y}[n]|^2] \\ &= E[\mathbf{w}^H \mathbf{x}[n] (\mathbf{w}^H \mathbf{x}[n])^H] \\ &= \mathbf{w}^H \mathbf{R} \mathbf{w} \end{aligned} \quad (2)$$

while keeping constant the amplitude of the output corresponding to the desired angle of arrival, this is

$$\mathbf{w}^H \mathbf{s}_d[n] = r \quad (3)$$

Applying Lagrange multipliers to this constrained optimization problem leads to the solution

$$\mathbf{w} = \frac{r \mathbf{R}^{-1} \mathbf{s}_d[n]}{\mathbf{s}_d^H[n] \mathbf{R}^{-1} \mathbf{s}_d[n]} \quad (4)$$

which is the well known minimum variance distortionless method (MVDM).

The above algorithms are linear since they can be expressed as the dot product between a parameter vector and a data vector. Such a class of algorithms can be easily nonlinearized using the Kernel Trick.

16.2.2 The Kernel trick

The class of linear algorithms $y[n] = \mathbf{w}^T \mathbf{x}[n] + b$ whose parameters $\mathbf{w}$ can be expressed as a linear combination of dot products between data, i.e. $\mathbf{w} = \sum \alpha_i \mathbf{x}_i$, can be nonlinearized through the Kernel Trick (Vapnik, 1995). The basic idea is to have a nonlinear transformation $\varphi(\cdot)$ of the data $\mathbf{x}[n]$ into a higher (possibly infinite) Hilbert space for which the associated dot product is expresable as a function $K(\cdot, \cdot)$ of the input data, i. e.:

$$\langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j) \quad (5)$$

Such a function is called a Mercer Kernel. A Hilbert Space provided with a kernel is called a Reproducing Kernel Hilbert Space. Mercer's Theorem states the conditions for a Kernel to be a dot product in a Hilbert Space.

A nonlinear version of the estimator using kernels can be obtained by simply using a valid kernel as dot product instead of the linear dot product. Assuming a nonlinear transformation $\varphi(\mathbf{x})$ of the data, the algorithm has a (primal) expression as

$$y[n] = \mathbf{w}^T \varphi(\mathbf{x}[n]) + b \quad (6)$$

If the weight vector has an expression as a linear combination of a dataset

$$\mathbf{w} = \sum_{i=1}^N \alpha_i \varphi(\mathbf{x}[i]) \quad (7)$$

then, using the Kernel Trick, the estimator has a (dual) expression as

$$y[n] = \sum \alpha_i K(\mathbf{x}_i, \mathbf{x}[n]) + b \quad (8)$$

Valid kernels are, among many others, polynomial $\left((x_i^T x_j + 1)^p\right)$ and Gaussian RBF $\left(\exp\left(\frac{1}{2\sigma^2} \|x_i - x_j\|^2\right)\right)$.

16.3 A NONLINEAR SUPPORT VECTOR ARRAY PROCESSOR

16.3.1 Nonlinear SVM Minimum Variance Distortionless Response Method

The formulation for the MVDM can be combined with the SVM formulation (Capon, 1969) to obtain a nonlinear beamformer with control of the interference. We present here a nonlinear counterpart of MVDM algorithm using SVM. We apply here the complex SVM formulation presented in Martínez-Ramón, Xu et al., (2005), and the robust cost function derivation of the SVM presented in Rojo-Álvarez, Camps-Valls et al. (2005). Assume that a nonlinear transformation $\mathbf{x}: R^N \rightarrow \varphi(x): H$ from the input space R^N to a higher dimensional Hilbert space H whose dot product can be expressed in kernel form, i.e. $\varphi(\mathbf{x}_1)^H \varphi(\mathbf{x}_2) = K(\mathbf{x}_1, \mathbf{x}_2)$. A nonlinear SVM beamformer with sidelobe control can be expressed as:

$$L_p = \frac{1}{2} \mathbf{w}^H \mathbf{R} \mathbf{w} + \sum_i \ell_R(\xi_i + \xi_i) + \sum_i \ell_R(\zeta_i + \zeta_i) \quad (9)$$

where the autocorrelation matrix $\mathbf{R}$ has the expression

$$\mathbf{R} = \frac{1}{N} \Phi \Phi^H \quad (10)$$

Φ being a matrix whose column i is equal to $\varphi(\mathbf{x}_i)$ and where ℓ_R is the cost function:

$$\ell_R(e) = \begin{cases} 0 & |e| < \varepsilon \\ \frac{1}{2\gamma} (|e| - \varepsilon)^2 - \varepsilon & \varepsilon \leq |e| \leq \varepsilon + e_c \\ C(|e| - \varepsilon) - \frac{1}{2}\gamma C^2 & e_c \leq |e| \end{cases} \quad (11)$$

with $e_c = \varepsilon + \gamma C$, which is applied to the MVDM constraints, but adapted to the Support Vector formulation

$$\begin{aligned}
\operatorname{Re}(r[n] - \mathbf{w}^H(\mathbf{s}_d[n])) &\leq \varepsilon - \xi[n] \\
\operatorname{Im}(r[n] - \mathbf{w}^H(\mathbf{s}_d[n])) &\leq \varepsilon - \zeta[n] \\
\operatorname{Re}(-r[n] + \mathbf{w}^H(\mathbf{s}_d[n])) &\leq \varepsilon - \xi'[n] \\
\operatorname{Im}(-r[n] + \mathbf{w}^H(\mathbf{s}_d[n])) &\leq \varepsilon - \zeta'[n]
\end{aligned} \tag{12}$$

Applying Lagrange analysis to this primal functional straightforwardly gives the result

$$\mathbf{w} = \mathbf{R}^{-1} \Phi_d \boldsymbol{\psi} \tag{13}$$

where $\Phi_d = [\varphi(s_d[1]), \dots, \varphi(s_d[n])]^T$. Applying the previous result, and expressing the inverse of the autocorrelation matrix as $\mathbf{R}^{-1} = \mathbf{U} \mathbf{D}^{-1} \mathbf{U}^H$, one can write the dual as

$$\frac{1}{2} \boldsymbol{\psi}^H [\Phi_d^H \mathbf{U} \mathbf{D}^{-1} \mathbf{U}^H \Phi_d + \gamma \mathbf{I}] \boldsymbol{\psi} - \operatorname{Re}(\boldsymbol{\psi}^H \mathbf{r}) + \varepsilon 1(\alpha + \beta + \alpha' + \beta') \tag{14}$$

The optimization of the dual (14) gives us the lagrange multipliers $\boldsymbol{\psi}$ from which one can compute the optimal weight vector (13).

This algorithm is not solvable due to the fact that we may not have access to the data in the feature space. But we still can indirectly solve the problem by applying Kernel Principal Component Analysis (KPCA) techniques. Let the autocorrelation matrix in the feature space be defined as in (10). The eigenvalues $\mathbf{D}$ and eigenvectors $\mathbf{U}$ of these matrices satisfy $\mathbf{D} \mathbf{U} = \mathbf{R} \mathbf{U}$. Also, the eigenvectors can be expressed as a linear combination of the dataset as $\mathbf{U} = \Phi \mathbf{V}$. Combining these two expressions and premultiplying by Φ^H we get

$$\mathbf{D} \Phi^H \Phi \mathbf{V} = \Phi^H \frac{1}{N} \Phi \Phi^H \Phi \mathbf{V} \tag{15}$$

Here, one can find the matrix $\mathbf{K} = \Phi^H \Phi$ whose element i, j is the dot product $\varphi(\mathbf{x}_i)^H \varphi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$, that we can compute.

Putting expression (15) in (14) gives the result

$$\frac{1}{2} \boldsymbol{\psi}^H [\Phi_d^H \Phi \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^H \Phi_d + \gamma \mathbf{I}] \boldsymbol{\psi} - \operatorname{Re}(\boldsymbol{\psi}^H \mathbf{r}) + \varepsilon 1(\alpha + \beta + \alpha' + \beta') \tag{16}$$

The matrix $\mathbf{V} \mathbf{D}^{-1} \mathbf{V}^H = \mathbf{K}^{-1}$ appears, so we can rewrite

$$\frac{1}{2}\psi^H \left[\mathbf{K}_d^H \mathbf{K}^{-1} \mathbf{K}_d + \gamma \mathbf{I} \right] \psi - \text{Re}(\psi^H \mathbf{r}) + \varepsilon 1(\alpha + \beta + \alpha' + \beta') \quad (17)$$

This expression contains now two matrices that we know. The first is the Gram matrix of kernel products whose elements are known. The second is the matrix $\mathbf{K}_d = \Phi^H \Phi_d$ whose elements are $K(\mathbf{x}[n], \mathbf{s}_d[m])$. This dual functional can be optimized using quadratic a programming procedure.

Putting Eq.(15) into (13) gives the expression of the weights as a function of the dual parameters

$$\mathbf{w} = \mathbf{R}^{-1} \Phi_d \psi = \Phi \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^H \Phi^H \Phi_d \psi = \Phi \mathbf{K}^{-1} \mathbf{K}_d \psi \quad (18)$$

and then the support vector machine output for a snapshot $\mathbf{x}[m]$ can be expressed as

$$y[m] = \mathbf{w}^H \varphi(\mathbf{x}[m]) + b = \psi^H \mathbf{K}_d \mathbf{K}^{-1} \Phi^H \varphi(\mathbf{x}[m]) + b = \psi^H \mathbf{K}_d \mathbf{K}^{-1} \mathbf{k}[m] + b \quad (19)$$

where $\mathbf{k}[m] = [K(\mathbf{x}[1], \mathbf{x}[m]), \dots, K(\mathbf{x}[N], \mathbf{x}[m])]^T$

For the autocorrelation matrix to be computed in the Hilbert Space, the data must be centered, which implies

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{A}\mathbf{K} - \mathbf{K}\mathbf{A} + \mathbf{A}\mathbf{K}\mathbf{A} \quad (20)$$

where $\mathbf{A}$ is an $N \times N$ matrix whose elements are equal to $\frac{1}{M}$, and

$$\tilde{\mathbf{k}}[m] = \mathbf{k}[m] - \mathbf{a}\mathbf{K} - \mathbf{k}[m]\mathbf{A} + \mathbf{a}\mathbf{K}\mathbf{A} \quad (21)$$

where $\mathbf{a}$ is a row vector whose elements are equal to $\frac{1}{N}$.

16.3.2 Solution of Primal optimization in Hilbert Subspaces

Dual problems are hard to solve online, because matrix $\mathbf{K}$ containing all dot products grows with the number of data. In order to avoid this inconvenience, it is useful to reformulate the given optimization problem as a projection on a subspace that adequately represents the given data. The subspace will be spanned by a selected subset $\mathbf{c}_i$ of the incoming data through a selection procedure, allowing a finite dimension primal vector $\mathbf{w}$ to optimize which can be expressed as a linear combination of the vectors in the subspace

$$\mathbf{w} = \sum_i \beta_i \varphi(\mathbf{c}_i) \quad (22)$$

where $\mathbf{c}_i$ is a set of selected vectors in the input space. In order to keep base $\mathbf{c}_i$ representative, growing and pruning procedures must be applied to the set. We apply here the approach of Navia-Vazquez, Pérez-Cruz et al. (2001).

16.3.3 Growing procedure

Consider a set of R vectors $\mathbf{c}_i$ in $\mathbb{R}^n$ (or $\mathbb{C}^n$) whose mapping $\varphi(\mathbf{c}_i)$ into H spans a subspace of dimension less than or equal to R . We consider that this base is representative enough if for any incoming vector $\mathbf{x}[n]$, its projection over the subspace has a representation error less than a given threshold ν^{grow} , that is:

$$\min_{d_i} \left(\sum_i d_i \varphi(\mathbf{c}_i) - \varphi(\mathbf{x}[n]) \right)^2 < \nu^{grow} \quad (23)$$

If the representation error exceeds the threshold, we will need to add the incoming vector to the base, i.e. $\mathbf{c}_{R+1} = \mathbf{x}[n]$. In order to solve the minimization (23), we develop the squared norm:

$$\begin{aligned} \left(\sum_i d_i \varphi(\mathbf{c}_i) - \varphi(\mathbf{x}[n]) \right)^2 = \\ K(\mathbf{x}[n], \mathbf{x}[n]) + \mathbf{d}^T \mathbf{K}_c \mathbf{d} - 2\mathbf{d}^T \mathbf{k}_n \end{aligned} \quad (24)$$

where $\mathbf{K}_c$ is the matrix of dot products $K(\mathbf{c}_i, \mathbf{c}_j)$ between elements of the base, $\mathbf{d}$ is the vector of projections to be found and $\mathbf{k}_n$ is the vector of dot products $K(\mathbf{x}[n], \mathbf{c}_i)$. The solution of this optimization problem is straightforwardly found by computing the square error gradient:

$$\mathbf{d} = (\mathbf{K}_c)^{-1} \mathbf{k}_n \quad (25)$$

This procedure requires a matrix inversion per data sample, which represents an unbearable computational burden, as matrix $\mathbf{K}_c$ grows. Even if we apply a pruning procedure, this matrix may have a considerable dimension. Nevertheless, one may take advantage of the fact that this matrix is definite positive.

Let $\mathbf{K}_n$ be a kernel matrix computed from the n elements of the existing base. Let us assume, also, that $\mathbf{K}_n^{-1}$ is known. If a new sample $\mathbf{c}_{n+1}$ is added to the base, then this matrix will be incremented as

$$\mathbf{K}_{n+1} = \begin{pmatrix} \mathbf{K}_n & \mathbf{k}_{n+1} \\ \mathbf{k}_{n+1}^T & k(n+1) \end{pmatrix} \quad (26)$$

where $k(n+1) = k(\mathbf{c}_{n+1}, \mathbf{c}_{n+1})$. The question now is, given that $\mathbf{K}_n^{-1}$ is known, how to compute the inverse of $\mathbf{K}_{n+1}$?

Let us first write this inverse as

$$\begin{pmatrix} \mathbf{K}_n & \mathbf{k}_{n+1} \\ \mathbf{k}_{n+1}^T & k(n+1) \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{P} & \mathbf{q} \\ \mathbf{q}^T & r \end{pmatrix} \quad (27)$$

and, thus

$$\begin{pmatrix} \mathbf{K}_n & \mathbf{k}_{n+1} \\ \mathbf{k}_{n+1}^T & k(n+1) \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{P} & \mathbf{q} \\ \mathbf{q}^T & r \end{pmatrix} = \mathbf{I}_{n+1 \times n+1} \quad (28)$$

Identifying terms in Eq.(28), we have

$$\mathbf{K}_n \mathbf{P} + \mathbf{k}_{n+1} \mathbf{q}^T = \mathbf{I}_{n \times n} \quad (29)$$

$$\mathbf{k}_{n+1}^T \mathbf{q} + k(n+1)r = 1 \quad (30)$$

$$\mathbf{K}_n \mathbf{q} + \mathbf{k}_{n+1} r = 0 \quad (31)$$

from which

$$r = \frac{1}{k(n+1) - \mathbf{k}_{n+1}^T \mathbf{K}_n^{-1} \mathbf{k}_{n+1}} \quad (32)$$

$$\mathbf{P} = \mathbf{K}_n^{-1} (\mathbf{I} + \mathbf{k}_{n+1} \mathbf{k}_{n+1}^T \mathbf{K}_n^{-1} r) \quad (33)$$

$$\mathbf{q} = -\mathbf{K}_n^{-1} \mathbf{k}_{n+1} r \quad (34)$$

If we first compute $\mathbf{z} = \mathbf{K}_n^{-1} \mathbf{k}_{n+1}$, then

$$\mathbf{K}_{n+1}^{-1} = \begin{pmatrix} \mathbf{K}_n^{-1} (\mathbf{I} r^{-1} + \mathbf{k}_{n+1} \mathbf{z}^T) & -\mathbf{z} \\ -\mathbf{z}^T & 1 \end{pmatrix} r \quad (35)$$

which avoids a direct matrix computation, saving a huge quantity of time when the matrix grows. As an example, we measure the computation time of direct matrix inversion compared to this approach (see Figure 1, left).

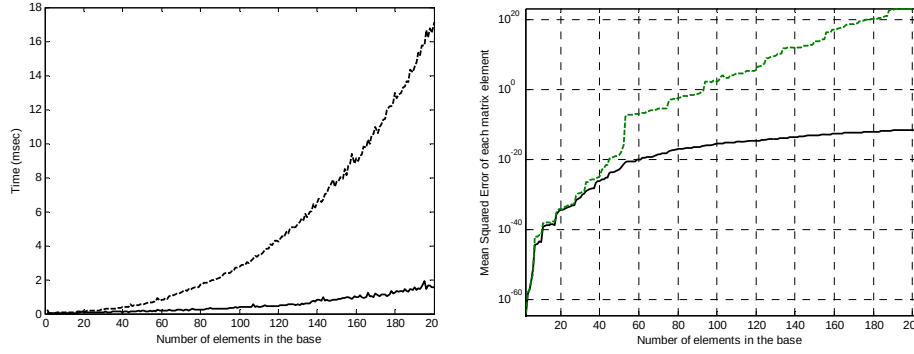


Figure 1: (Left) Computation time for the inverse matrix computation. Dash: direct computation; solid: iteratively using Eq.(34). (Right) Computation error. Solid: with regularization; dash: without regularization.

This approach may be ill conditioned because of the denominator of Eq.(33), but it is easily regularizable by adding a small term. For those cases in which $k(n+1) = k(\mathbf{c}_{n+1}, \mathbf{c}_{n+1}) = 1$ (as for example when using gaussian RBF kernels), the easiest way to regularize is to multiply $\mathbf{k}_{n+1}$ by $1 - \gamma$, $\gamma \ll 1$. This is equivalent to transform the kernel matrix by a convex combination as $\tilde{\mathbf{K}}_n = (1 - \gamma)\tilde{\mathbf{K}}_n + \gamma\mathbf{I}_{n \times n}$. Figure 1 (right) shows the importance of this regularization. This image shows the computation error when iteratively computing the matrix inverse of a kernel matrix constructed with a gaussian RBF kernel.

16.3.4 Pruning procedure

We compute the accumulated optimal projection coefficients and identify which base elements have less relevance. Let $\mathbf{d}_i^*$ be the optimal coefficients for representing pattern i , the accumulated weighted projection up to time n is simply $\mathbf{d}_n^{cum} = \sum_{i=1}^n \lambda^{(n-i)} |\mathbf{d}_i^*|$ and normalizing it such that its maximum value is 1.

Therefore, we only need to evaluate it if any of the elements of $\mathbf{d}_n^{cum}$ is below a given pruning threshold ν^{prune} and eliminate it from the base. This being the case,

we will also have to modify some vectors and matrices to be used in the adaptive algorithm by eliminating the corresponding values.

16.4 ADAPTIVE SUPPORT VECTOR MACHINES

Consider a set of snapshots $\mathbf{x}_j$ from an array and the corresponding set of (Quadrature Amplitude Modulation, QAM, or Phase Shift Keying, PSK) desired symbols y_j available for training purposes. A SVM classifier approach with complex arithmetics (Martínez-Ramón, Rojo-Álvarez et al., 2007) consists of minimizing the functional

$$L_p = \frac{1}{2} \mathbf{w}^2 - C \sum_i \xi_i - C \sum_i \zeta_i \quad (36)$$

subject to the constraints

$$\begin{aligned} \text{Re}[y_i^* o_i - A_i^2] &\leq \xi_i \\ \text{Im}[y_i^* o_i - A_i^2] &\leq \zeta_i \end{aligned} \quad (37)$$

where ξ_i and ζ_i are positive slack variables introduced to deal with non-separable problems, $A_i = |y_i|^2$, and C is the penalization for patterns incorrectly classified or inside the margins $|\text{Re}(y_i^* o_i)| < A_i, |\text{Im}(y_i^* o_i)| < 0$. This is a complex extension of the original SVM primal. Defining $\mathbf{K}_c$ as the matrix of kernel dot products $\varphi(\mathbf{c}_k)^T \varphi(\mathbf{c}_l) = k(\mathbf{c}_k, \mathbf{c}_l)$, and using (22), the norm of the weights can be written as $\|\mathbf{w}\|^2 = \beta^H \mathbf{K}_c \beta$. Using a Lagrange optimization and applying the Karush Kuhn Tucker conditions as in Burges, (1998), primal (36) takes the form

$$L = \frac{1}{2} \beta^H \mathbf{K}_c \beta - \sum_i \alpha_i \text{Re}[y_i^* o_i - A_i^2] - \sum_i \alpha'_i \text{Im}[y_i^* o_i - A_i^2] \quad (38)$$

where α_i, α'_i are Lagrange multipliers for the real and imaginary constraints defined in (26). Defining $e_i = y_i - o_i$, equation (27) can be written as

$$\begin{aligned} L &= \frac{1}{2} \beta^H \mathbf{K}_c \beta + \sum_i \alpha_i \text{Re}[y_i^* e_i] + \sum_i \alpha'_i \text{Im}[y_i^* e_i] \\ &= \frac{1}{2} \beta^H \mathbf{K}_c \beta + \sum_i \text{Re}[\psi_i y_i^* e_i] \end{aligned} \quad (39)$$

where $\psi = \alpha_i + j\alpha_i'$. To obtain an on-line adaptive SVM, we can transform this problem into an iterative re-weighted least squares problem minimization problem (Navia-Vazquez, Pérez-Cruz et al., 2001)

$$L = \frac{1}{2} \beta^H \mathbf{K}_c \beta + \sum_i (a_i + b_i) |e_i|^2 \quad (40)$$

where

$$a_i = \begin{cases} 0 & \text{Re}(y_i^* e_i) < 0 \\ M & 0 < \text{Re}(y_i^* e_i) < C/M \\ \frac{C \text{Re}(y_i^* e_i)}{|e_i|^2 A_i} & \text{Re}(y_i^* e_i) \geq C/M \end{cases} \quad (41)$$

$$b_i = \begin{cases} 0 & \text{Im}(y_i^* e_i) < 0 \\ M & 0 < \text{Im}(y_i^* e_i) < C/M \\ \frac{C \text{Im}(y_i^* e_i)}{|e_i|^2 A_i} & \text{Im}(y_i^* e_i) \geq C/M \end{cases} \quad (42)$$

We upper bound a_i, b_i by M when they are small in order to avoid singularities. This problem also appears in the standard SVM dual formulation when the Hessian matrix is rank deficient.

For any fixed set of parameters a_i, b_i minimization of (40) is a weighted least squares problem, where $a_i + b_i$ are the weights and β are the parameters to optimize. If we compute the gradient of (40) with respect to β , we obtain the following equation for their optimal values

$$[\mathbf{K}^T \mathbf{D} \mathbf{K} + \mathbf{K}_c] \beta = [\mathbf{K}^T \mathbf{D} (\mathbf{y} + \mathbf{b})] \quad (43)$$

where $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{c}_j)$, $\mathbf{y} = [y_1, \dots, y_N]^T$, and $\mathbf{D} = \text{diag}\{a_i + b_i\}$. The so called Iterative Re-weighted Least Squares (IRWLS) procedure consists of iteratively solving (43) and then, using this solution, finding a_i, b_i using (41) and (42). The above procedure involves an inverse matrix computation in each step, which is computationally costly. Nevertheless, using the matrix inversion lemma, the inverse can be avoided, and the resulting procedure is similar to the RLS.

First let us re-write the term to be inverted $\mathbf{R}_{zz}[n] = (\mathbf{K}_n^T \mathbf{D}[n] \mathbf{K}_n + \mathbf{K}_c)$ in a recursive form, using rank-one matrix updates:

$$\mathbf{R}_{zz}[n] = \mathbf{K}_n^T \mathbf{D}[n] \mathbf{K}_n + \mathbf{K}_c = \lambda \mathbf{R}_{zz}[n-1] + \mathbf{u}_n \mathbf{u}_n^T \quad (44)$$

For equation (44) to hold $b[n]$, $\mathbf{w}[n]$ and $\beta[n]$ are assumed to vary slowly, such that we can assume that $e_i[n] \approx e_i[n-1]$ to obtain the approximation:

$$\begin{aligned} a_i[n] &= \lambda a_i[n-1] \\ b_i[n] &= \lambda b_i[n-1] \end{aligned} \quad (45)$$

Also, update vector $\mathbf{u}_n$ is chosen such that $\mathbf{u}_n \mathbf{u}_n^T = \mathbf{z}_n a_n[n] \mathbf{z}_n^T + (1-\lambda) \mathbf{K}_c$. This is achieved by choosing $\mathbf{u}_n = \mathbf{z}_n a_n^{1/2}[n] + \mathbf{r}_n$. Vector $\mathbf{r}_n$ is constructed using ‘noise injection method’, which allows dynamic regularization adaptive filters and neural networks. The covariance matrix of the noise must verify $E\{\mathbf{r}_n \mathbf{r}_n^T\} = (1-\lambda) \mathbf{K}_c$. To build such a noise vector, we define $\mathbf{r}_n = \{\mathbf{n}_n^T; \mathbf{s}_n^T\}^T$, $\mathbf{n}_n$ is built with P samples drawn from a $N(0, \sqrt{1-\lambda})$ distribution, and $\mathbf{s}_n = \mathbf{F}^T \mathbf{n}'_n$ where $(1-\lambda) \mathbf{K}_c = \mathbf{F}^T \mathbf{F}$, and $\mathbf{n}'_n$ is built with R independent and identically distributed (i.i.d.) samples from a $N(0,1)$ Normal distribution.

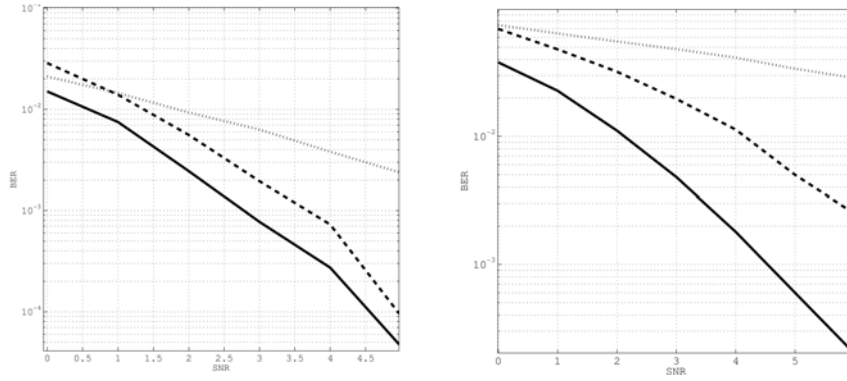


Figure 2: Comparisons of the linear MVDM (dot), the standard SVM (dash) and the MVDM-SVM.

We define $\mathbf{R}_{zz}[n] = \mathbf{Q}^{-1}[n]$, $\pi^T[n] = \mathbf{u}_n^T \mathbf{Q}[n-1]$, $\mathbf{q}[n] = \lambda + \pi^T[n] \mathbf{u}_n$, and $\mathbf{g}[n] = \pi[n] / \mathbf{q}[n]$ and apply the Lemma to $\mathbf{R}_{zz}[n] = \lambda \mathbf{R}_{zz}[n-1] + \mathbf{u}_n \mathbf{u}_n^T$, we obtain a recursion for $\mathbf{Q}[n] = \mathbf{R}_{zz}^{-1}[n]$:

$$\mathbf{Q}[n] = \frac{1}{\lambda} (\mathbf{Q}[n-1] - \mathbf{g}[n] \pi^T[n]) \quad (46)$$

It is also straightforward to obtain a recursion for $\mathbf{r}_{zy}[n] = \mathbf{K}_n^T \mathbf{D}[n] \mathbf{y}_n$:

$$\mathbf{r}_{zy}[n] = \lambda \mathbf{r}_{zy}[n-1] + \mathbf{k}_n (a_n[n] + b_n[n]) y_n \quad (47)$$

and at every instant n the weight solution $\beta[n]$ is obtained as:

$$\begin{aligned}
\beta[n] &= \mathbf{Q}[n] \mathbf{r}_{zy}[n] \\
&\approx \beta[n-1] + \mathbf{g}[n] \left((a_n[n] + b_n[n]) y_n - \mathbf{u}_n^T \beta[n-1] \right)
\end{aligned} \tag{48}$$

16.5 EXPERIMENTS

16.5.1 SVM MVDM

We compare our SVM-MVDM to the linear MVDM and to the standard Support Vector Machine. The kernel that has been used in all simulations is the gaussian RBF $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2})$. The variance parameter σ is set to 0.1, $\gamma = 0.1$ and $C = 1000$. The scenario consists of a multiuser environment with four users, from which one is the desired user and the rest are considered interferences. Each one transmit a statistically independent Quadrature Phase Shift Keying (QPSK) signal from different directions of arrival (DOA) with the the same power. Signal corrupted by additive white Gaussian noise (AWGN). The desired signal is a train of bursts with a preamble of 100 training bits and 1000 signal bits. Figure 2 shows the bit error rate (BER) of the desired signal as a function of the signal to noise ratio (SNR). We can see here that the SVM-MVDM has an improvement of about 1 dB over the standard complex SVM, and more about 3 dB over the linear MVDM. In BER, this translates into an improvement of more than a magnitude order. In the left panel the desired signal comes from a DOA of 10° , where the interferences had DOAs of -10° , -20° and 20° . In the right one, the DOAs were 0° for the desired signal, where the intereferent ones came from -10° , -5° and 15° . Here the improvement of the SVM-MVDM over the standard SVM is of about 2 dB. Note that the linear algorithm gives a BER that is over 10^{-2} , which in practice means that it is not able to solve the problem.

The scenario for the experiments consists of three signals coming from DOA of 0° , 25° , and -30° . The modulations are three independent QPSK, with synchronous timing. All users have the same power, the first of them (0°) representing the desired signal. The array consists of 5 elements equally spaced at $\lambda/2$ and the signal is corrupted by white Gaussian noise. The transmitted frames contained 20 training symbols and 400 unknown (test) symbols.

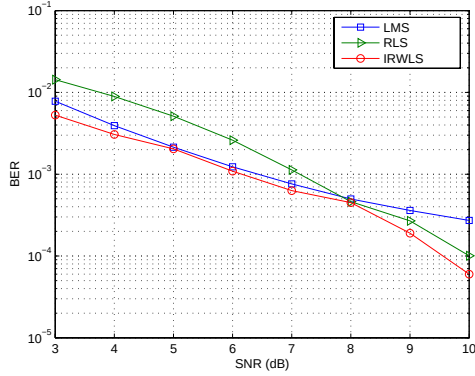


Figure 3: BER of linear adaptive algorithms, first scenario.

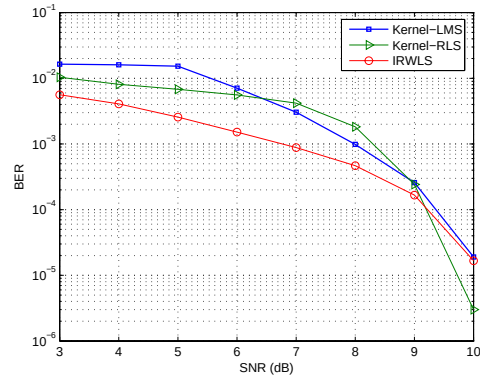


Figure 4: BER of kernel adaptive algorithms, first scenario.

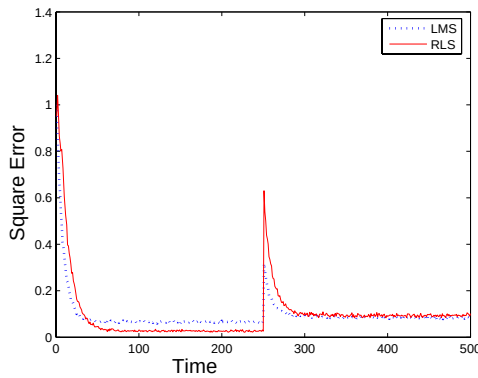


Figure 5: Mean Square Error for linear Methods, second scenario.

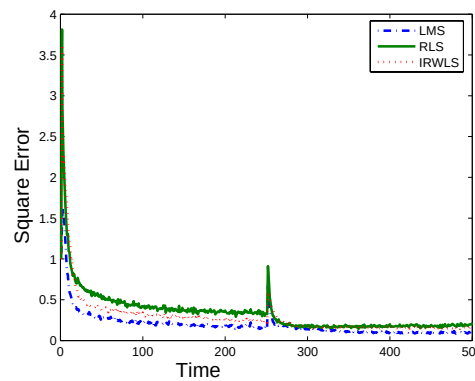


Figure 6: Mean Square Error for Kernel Methods, second scenario.

16.5.2 Adaptive SVM beamforming

We compare the IWRLS-SVM against the linear LMS, linear RLS and kernel versions of LMS and RLS.

We run two experiments. In the first one, the scenario is stationary, and we compare the BER of all systems for a range of different SNR. Figures 3 and 4 show the measured BER for all systems. All nonlinear systems show a significant

improvement for SNRs higher than 9 dB. The second experiment consists of a scenario in which after 250 received samples, the interferences disappear and two other appear from DOAs. Figures 5 and 6 show the MSE across time. It can be seen that the adaptation speed of all nonlinear systems is equal to or better than those of the linear ones. The used kernel is the Gaussian RBF.

16.6 CONCLUSIONS

We have presented an adaptive nonlinear framework applied to array processing. It uses a nonlinear transformation to a Hilbert space. We present a nonlinear version of the MVDM using SVM and Kernel PCA. Then, we present a way to produce adaptive versions of kernel methods. The data is projected into a subspace constructed by an online growing and pruning procedure in order to preserve the dimensions that are significant while discarding the others. In this subspace it is possible to formulate a complex version of an adaptive SVM through an IRWLS procedure. The performance of the linear LMS and RLS algorithms is compared in a static and a nonstationary scenario against kernel versions of LMS and RLS and the adaptive IRWLS-SVM.

ACKNOWLEDGEMENTS

This work has been supported by grants S-0505/TIC/0223 (Comunidad de Madrid, Spain) and TEC2005-00992 (CICYT, Spain).

REFERENCES

- Aizerman, M. A. , Braverman, E. M., & Rozoner, L.I. (1964). Theoretical foundations of the potential function method in pattern recognition learning,. *Aut. and remote Control*, vol. 25, pp. 821-837.
- Burges, C.J.C. (1998) "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 1–32.
- Capon, J. (1969). High resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8), 1408-1418.
- Haykin, S. (1996). *Adaptive filter theory*, 3rd ed., Prentice Hall, N.J., USA.
- Huber, P. J. (1972). Robust statistics: a review (43). *Ann. Statistics*.
- Martínez-Ramón, M., Rojo-Álvarez, J. L., Camps-Valls, G., Christodoulou, C. G., (2007). "Kernel Antenna Array Processing", *IEEE Trans. on Antennas and Propagation*, Vol. 55, N° 3, pp. 642 – 650.
- Martínez-Ramón, M., Rojo-Álvarez, J. L., Camps-Valls, G., Navia-Vázquez, A., Soria-Olivas, E. Figueiras-Vidal, A. R. (2006). Support Vector Machines for Nonlinear Kernel ARMA System Identification. *IEEE Transactions on Neural Networks*, 17(6), 1617-1622.
- Martínez-Ramón, M., Xu, N. Christodoulou, C. (2005). Beamforming using support vector machines. *IEEE Antennas and Wireless Propagation Letters*, 4, 439-442.
- Navia-Vazquez, A., Pérez-Cruz, F., Artes-Rodríguez, A., Figueiras-Vidal, A. R. (2001). "Weighted least squares training of support vector classifiers leading to compact and adaptive schemes" *IEEE Trans. on Neural Networks*, Vol. 12, No 5, pp. 1047 – 1059.
- Rojo-Álvarez, J. L., Camps-Valls, G., Martínez-Ramón, M., Navia-Vázquez, A. Figueiras-Vidal, A. R. (2005). Support Vector Machines Framework for Linear Signal Processing. *Signal Processing*, 85(12), 2316-2326.
- Rojo-Álvarez, J. L., Martínez-Ramón, M., Cumplido, M. dePrado, Artés-Rodríguez, A. Figueiras-Vidal, A. R. (2004). Support Vector Method for Robust ARMA System Identification. *IEEE Trans. on Signal Processing*, 52(1), 155-164.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*, Springer-Verlag, NY, USA.
- Vapnik, V. (1998). *Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control*. John Wiley & Sons.

Section V: Intelligent Systems and Medical diagnosis/imaging

Chapter 17

A Method of Clustering a Self Organising Map Neural Network to Determine Subgroups of Non-Specific Lower Back Pain Patients

Sam Empson, Chris McCarthy, Ranjit Lall, Evor Hines, Mark Leeson and
Daciana Iliescu

17.1 INTRODUCTION

Low back pain (LBP) and disability in particular, continue to be significant healthcare problems in western society (Waddell, 2004). LBP is a common condition with a lifetime prevalence of 58% (Walsh et al., 1992; Papageorgiou et al., 1995) though more recently, this has been quoted to be as high as 80% in industrialised countries (Walker et al., 2004). In 1998, the UK estimated economic burden due to LBP was £10.6 million (Maniadakis and Gray, 2000). LBP disability and costs of work loss and sickness benefits increased dramatically towards the end of the last century (Waddell, 2004).

In the UK, physiotherapists see many patients with LBP, with estimates varying from 9% to 50% of sufferers having sought help from physiotherapists (Maniadakis and Gray, 2000). NSLBP is synonymous with the terms ‘simple LBP’ or ‘mechanical LBP’ (McCarthy et al., 2004) and derives from an initial diagnostic triage process of classifying patients into one of three categories – nerve root irritation/compression, serious pathology and NSLBP (CSAG, 1994; Bigos et al. 1994; Koes et al., 2001). It is still not possible for clinicians to reach a valid pathoanatomical diagnoses for NSLBP, due to the poor correlation of clinical presentation (Kent & Keating, 2005) and diagnostic imaging findings.

Patients referred for conservative treatments with a diagnosis of NSLBP form an extremely heterogeneous group thereby presenting with a large variability in prognosis and potential treatment methods (McCarthy et al., 2004). The belief that several smaller homogeneous subgroups are likely to exist within the heterogeneous groups of NSLBP (McKenzie, 1981; Psitzer, 1987; Leboeuf-Yde et al., 1997; Bouter et al. 1998; McCarthy et al., 2004; Fritz et al., 2006), has prompted efforts to identify relevant subgroups of NSLBP patients (McKenzie, 1981; Quebec Task Force, 1987; Delitto et al., 1995; Maluf et al., 2000; McCarthy et al., 2004) which may effectively direct treatment, improve research efficiency and clinical outcomes (Fritz et al., 2000; Peterson et al., 2003). Some classification systems (Quebec Task Force, 1987; Delitto et al., 1995, Maluf et al., 2000), it may be argued, have been biased by judgmental approach, author experience and biomedical characteristics (McCarthy et al., 2004). Their quality, in terms of methodological development can only be classified as moderate and therefore have limited generaliseability to clinical practice. Statistical approaches to examining patient data, termed 'unsupervised analysis' (Munro, 2003) have greater methodological rigour as personal bias is reduced. Indeed, some studies provide initial evidence that classification guided subgrouping might aid the evaluation process and inform treatment direction (Delitto et al., 1995; O'Sullivan et al., 1997; Riddle, 1998; Fritz et al., 2003, Long and Donelson, 2004; Hicks et al., 2005).

Use of a multi layer perceptron (MLP) neural network has been investigated by Vaughn (2004) to attempt to classify low back pain patients into one of 3 groups. These consist of simple low back pain (synonymous with NSLBP), root pain and abnormal illness behaviour (AIB). The data in the study mainly consisted of results to a clinical examination which was recoded to a set of 48 binary variables. The resulting MLP achieved a classification accuracy of 99% in the training set and a corresponding 78.4% in the test set. In addition a set of rules were extracted from the MLP which demonstrated which variables had a positive or negative association with each group. Another study by Liszka-Hackzell and Martin (2002) has shown that the fusion of a self organising map (SOM) neural network and MLP network can be used to accurately distinguish whether a patient's LBP is acute or chronic based on pain and activity data. The SOM was used to categorise patients based on their pain and activity, then the MLP was used to learn the relationship between a patient's category and health outcome.

With a view to investigating the potential of non-supervised, statistical analysis of patient data, the Warwick Medical School (WMS) have conducted a study which involved carrying out an examination of 300 patients with NSLBP by physiotherapists. The study resulted in demographic data, clinical outcomes to question and physical tests and self-reported questionnaire data. The purpose of this analysis was to investigate the grouping characteristics of the clinical data. Separate analysis will be undertaken using the self-reported questionnaire data.

The data was analysed using intelligent systems engineering (ISE) techniques to investigate the possibility that homogeneous groups of signs and symptoms existed within the data.

17.2 BACKGROUND TECHNIQUES

17.2.1 A Self Organising Map (SOM)

17.2.1.1 Introduction to SOMs

A self organising map (SOM) is a type of unsupervised artificial neural network (NN) designed by Kohonen (2001) and is hence also known as a Kohonen network. The SOM has a number of data analysis applications, including modelling, variable correlation, vector projection and cluster detection (Vesanto et al. 2000b). The SOM consists of an input layer and a map layer, see Figure 1. Every input vector is connected to every map neuron and the weights of these connections are used to find which vectors are ‘similar’ to which areas of the map. The SOM algorithm iteratively adjusts the weights to make the map layer fit the input data as closely as possible. As the algorithm is based on Euclidean distance measures, it can only accept ordinal variables. This is because for categorical variables the distance between classes has no meaning. For example, when considering the set of hair colours [blonde, red, brown, black], there are no ‘high’ or ‘low’ values. Hence any attempt to cluster this variable with ordinal variables would fail as any conclusion that, say, hair colour was positively correlated with height would be meaningless.

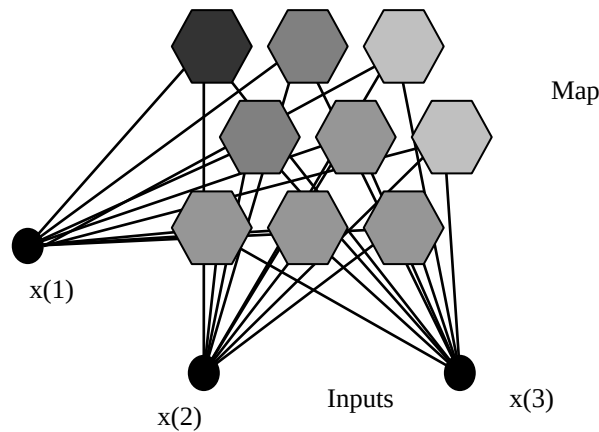


Figure 1: Structure of a SOM showing map layer of neurons connected to all input vectors.

17.2.1.2 Advantages in using a SOM

Some reasons why the SOM may be advantageous are:

- (1) That it is less adversely affected by 'messy' data. So for example, it is better able to detect clusters of non-uniform density, and is less adversely affected by noise caused by 'irrelevant' variables.
- (2) If data is missing from some variables for a sample, this does not preclude it from being assigned to the correct cluster. This occurs because the connection weight to the output layer for the missing variable is set to zero, so it has no effect on the structuring of the map.
- (3) That the SOM is able to plot patient classifiers on the completed map. As these variables are categorical rather than ordinal they cannot be used by the SOM algorithm to create the map.

17.2.1.3 Structure and Operation of SOM

The number of neurons in the input layer corresponds to the number of dimensions of the input data. The size, shape and topology of the map layer can be determined by the user, although they are typically based on the number of data samples available. The structure to be used in this analysis is a 'sheet' type,

as in Figure 1, where the lattice is hexagonal, i.e. all map neurons will be connected to its six neighbours.

In operation the SOM creates a two-dimensional grid of neurons which maintains the topology of the potentially much higher dimensional input data set and so can be used for finding clusters in high dimensional data. The neurons are assigned m-dimensional ‘prototype vectors’, where m is the number of variables in the data set. The components of these vectors represent possible values of each variable. The prototype vectors can be seen as prototype cluster centres.

The steps which the SOM algorithm follows are:

- (1) The prototype vectors are assigned initial values which can either be randomly assigned or based on properties of the data set. A description of this process is provided in section “Conditions required to allow Comparison of SOM Clustering Results”.
- (2) Next a competitive ‘batch’ algorithm is run which involves presenting a group of samples to the map layer and identifying the prototype vectors with the smallest Euclidean distance to each sample. The presentation of one batch of samples to the algorithm is termed an epoch. The size of each batch is either equal to the number of map units or the number of samples, whichever is smaller. An alternative ‘sequential’ method involves updating the map every time after a single sample has been presented. The batch method was used in this work as it has been shown by Kohonen (2001) to be much more computationally effective. However Nöcker (2006) has shown that use of the batch method can have undesired effects on the SOM. This can be seen where clusters of equal size and density appear severely distorted on the SOM.
- (3) After all members of a ‘batch’ have been presented to the map layer, the prototype vectors identified as best matching units (BMUs) (i.e. the prototype vectors with the smallest Euclidean distance to each batch member) and their neighbours are then adjusted to bring the map units closer to the samples. This is done by replacing the prototype vector of each map unit by the weighted average of the data samples that were in its neighbourhood. This includes samples whose BMU was the map unit being updated, as well as samples whose BMU was in the neighbourhood of this unit. The weighting of samples for neighbouring map units decreases with distance from the unit following a Gaussian distribution. The ‘reach’ of this function is also decreased with time, so initially changes are made based on many samples, but this reduces linearly with time as the map becomes

organised. The rule for updating the prototype vector m of map unit i , for the samples x which have BMU j , is given by:

$$m_i(t+1) = \frac{\sum_{j=1}^n h_{ic}(t)x_j}{\sum_{j=1}^n h_{ic}(t)} \quad (1)$$

where h is the Gaussian distributed weighting component which depends both on the distance from the BMU of sample x to unit m as well as the number of epochs, t , the algorithm has completed.

- (4) Another batch is then chosen and the process returns to step 2 until a predetermined number of epochs have been exceeded.

17.2.1.4 SOM Training

The training process is carried out in two stages. The coefficients for each stage are described in Table 1. These coefficients are generated by the algorithm to be good values based on the size of the data set and the size of the SOM. In turn the size and shape of the SOM is determined based on the size of the data set and its eigenvalues and corresponding eigenvectors. This is explained further in section “Conditions required to allow Comparison of SOM Clustering Results”.

The training process involves two distinct phases of training. In both stages the ‘neighbourhood distance’ reduces linearly for the duration of the training. The ‘neighbourhood distance’ defines a boundary from an input vector, measured using an Euclidean distance. ‘Prototype vectors’ within the boundary will be adjusted however ‘prototype vectors’ outside it will not. The first stage of training is called ‘rough’ training where the initial neighbourhood distance is large and training length is short. This results in large numbers of ‘prototype vectors’ being adjusted with the aim of getting the SOM into approximately the right shape. The second phase of training is called ‘fine-tuning’, where initial neighbourhood distances are low and the training length is longer. The resulting smaller adjustments of ‘prototype vectors’ will ‘fine-tune’ the ‘prototype vector’ coefficients.

17.2.1.5 SOM U-Matrix and Component Plane Visualisations

The resulting two dimensional map layer represents the multi-dimensional data set while maintaining its topology (i.e. inter-sample distances) and can be visualised in a number of ways. The most common forms are the unified distance

Table 1: SOM training coefficients for ‘rough’ and ‘fine-tuning’ stages.

	Rough	Fine-tuning
Initial neighbourhood distance	$\frac{ms}{8}$	$\max(\frac{ms}{32}, 1)$
Final neighbourhood distance	$\max(\frac{ms}{32}, 1)$	1
Training length (epochs)	$10 \times \frac{munits}{dlen}$	$40 \times \frac{munits}{dlen}$

where

$munits$ =number of map units

$dlen$ =number of samples

ms =width or height of map layer, whichever is greater

matrix (u-matrix) and component planes. These allow the underlying structure of the data to be examined. These forms of SOM visualisation are known as ‘cell’ visualisations, the two main alternative forms of visualisation are ‘mesh’ and ‘graph’. Let us consider these in turn:

- (1) ‘Mesh’ visualisations involve plotting the map prototype vectors on a three dimensional membrane. This can be achieved by reducing the dimensionality of the vectors using principle component analysis (PCA). Vesanto et al. (2000) describe these plots to be more efficient than ‘component planes’ when only a few variables are of interest. The distance between neurons in the plot represents the distance in the typically higher dimensional prototype vector subspace. This can be used to visualise the structure of the map but it must be remembered that accuracy is potentially reduced through the use of PCA. This may occur because the PCA process involves data loss through the removal of components with the smallest contribution to the output. An example ‘mesh’ visualisation can be seen in Figure 2.
- (2) ‘Graph’ visualisations simply involves plotting extra information on top of each cell in a ‘cell’ visualisation. This could include extra information such as cluster membership or properties of the data samples for which the neuron is a BMU. An example ‘graph’ visualisation can be seen in Figure 3.

The ‘cell’ form of visualisation will be used predominantly in this work as the author’s experience indicates that it is the clearest way to visualise the SOM in

the context of this work. For a more general review of forms of SOM visualisation see Vesanto (1999).

The SOM output can be shown using tone indexed planes. These consist of a single unified distance matrix ('u-matrix') and have one 'component' plane per variable. The 'u-matrix' plane represents the distance between neighbouring map units of the SOM.

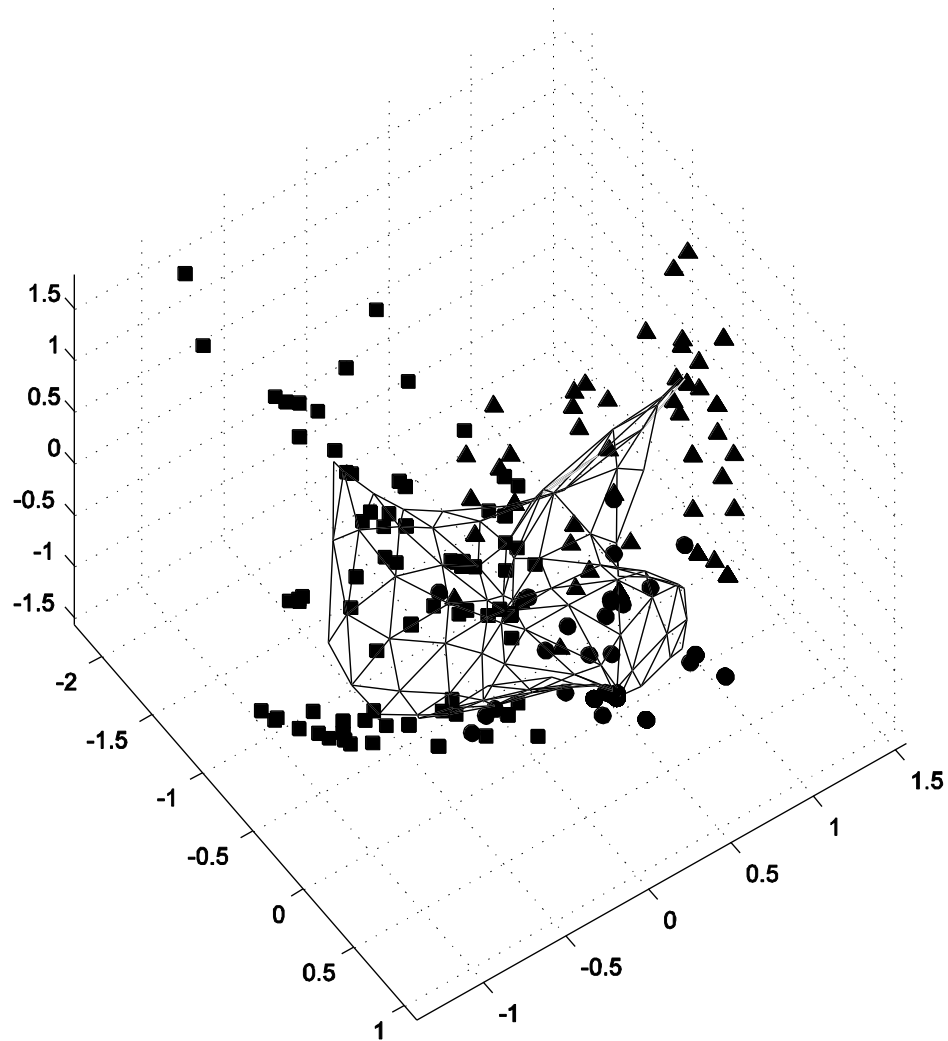


Figure 2: Example 'mesh' visualisation showing a PCA projection of a SOM map layer (grid) and distribution of data (points) showing cluster membership (point shape).

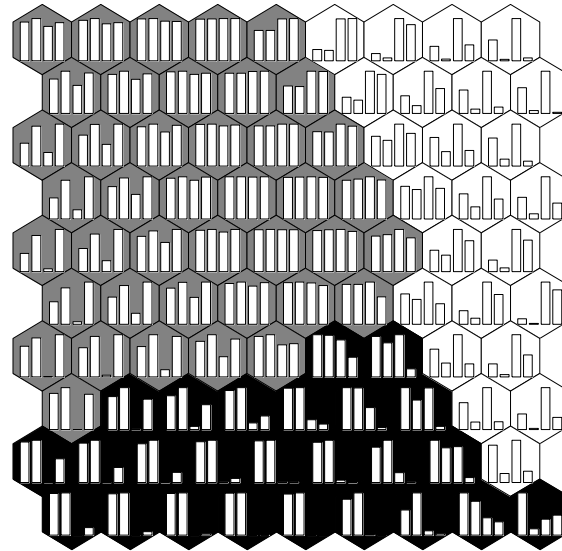


Figure 3: Example 'graph' visualisation showing cluster membership of map units (cell background tone) and prototype vector values of variables (bar charts).

As an example, a SOM which has dimensions 5x1 (see Figure 4), will have map units

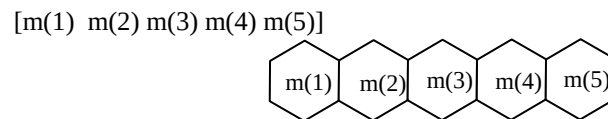


Figure 4: Example map layer containing 5 neurons.

This will lead to the 9x1 u-matrix

$[u(1) \ u(1,2) \ u(2) \ u(2,3) \ u(3) \ u(3,4) \ u(4) \ u(4,5) \ u(5)]$

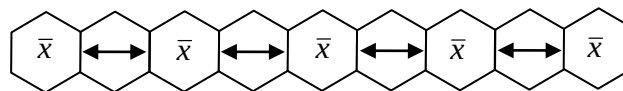


Figure 5: Representation of how map neurons create a unified distance matrix (u-matrix).

where $u(i,j)$ denotes the distance between map units $m(i)$ and $m(j)$ and $u(k)$ is the mean of the surrounding values, e.g. $u(3) = (u(2,3) + u(3,4))/2$. This can be seen visually in Figure 5.

One of the results of this process is that there are more cells in a u-matrix than there are map units. U-matrices can be used to indicate areas of clustering

that may exist in the data set. For a 5 neuron map, 4 (double-headed) cells are added and take the value of the distance between neurons. The values of the remaining (x-bar) cells take the value of the mean of the surrounding distance measure cells.

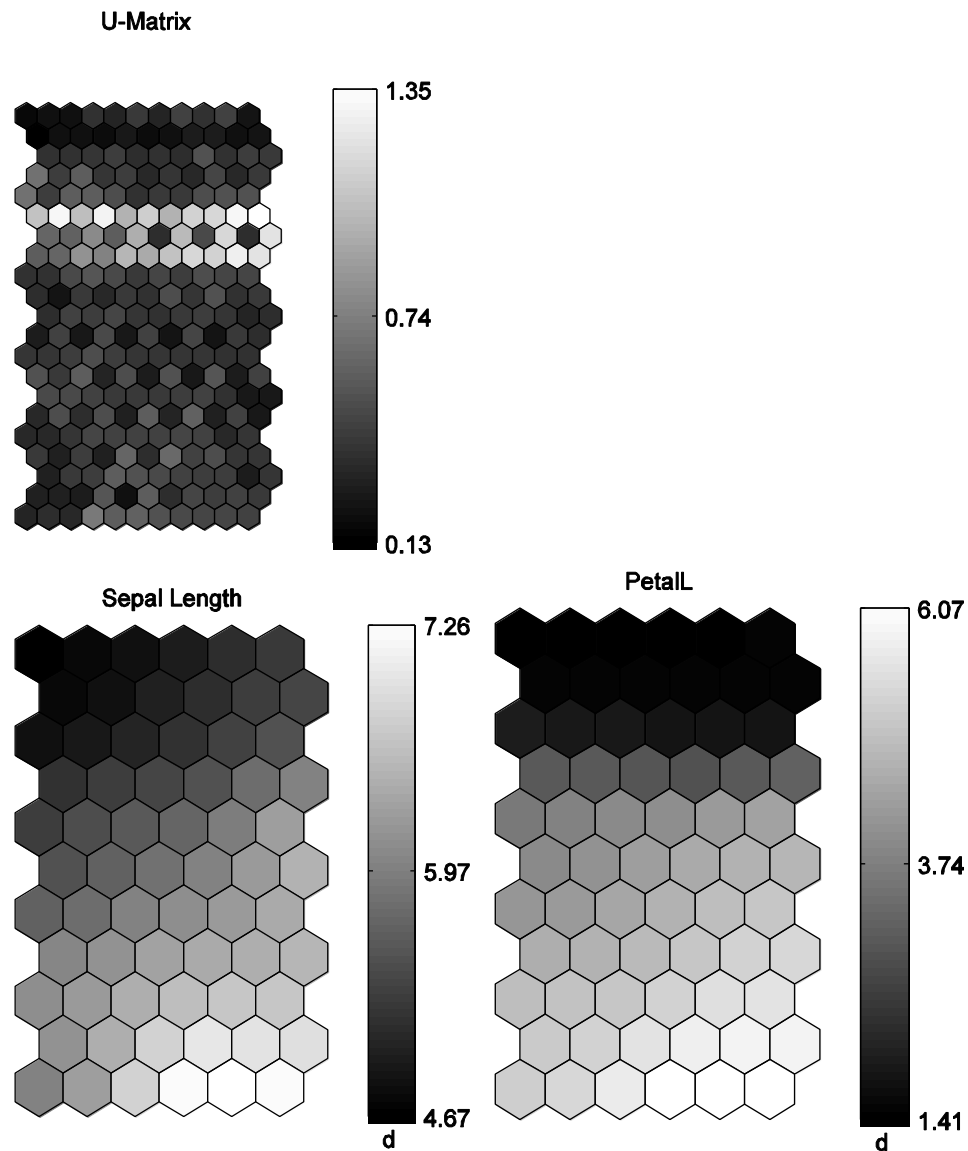


Figure 6: (a) Sample U-matrix of a SOM demonstrating two distinct clusters (b) Sample component

plane of a SOM for a sepal length variable. (c) Sample component plane of a SOM for a petal length variable. Data source: Anderson (1935).

An example u-matrix (Figure 6(a)) and two example component planes (Figure 6(b)) and 6(c)) are provided, which use data from Anderson's (1935) well known iris study. The set consists of 4 measurements of 150 iris flowers from 3 classes of iris: 50 setosa, 50 versicolor and 50 virginica. The measurements are of the length and width of the flowers' sepals and petals. The 'u-matrix' shows two clusters with a clear horizontal cluster boundary of white and pale grey cells. The scale to the right indicates the distance between neurons, hence a cluster can be visualised in terms of the cells with low distances and cluster edges can be visualised in terms of cells with high distances. Each cell in a component plane represents a single map unit and its tone represents a value of that variable. As each component plane is a 'slice' of the output, two planes showing similar tone in the same area of the map indicates that the variables are related. An example where the two variables are positively correlated can be seen in Figure 6(b) and 6(c). In this case it shows that iris flowers which have long sepals also have long petals.

17.2.1.6 Clustering Data using a Self-Organising Map (SOM)

The approach to be used is a two-level clustering method. This involves training a self-organising map (SOM) on the data set. The resulting prototype vectors are considered to be prototype clusters. As the number of prototype vectors is much greater than the number of clusters expected, they can then be clustered using methods such as k-means or fuzzy clustering algorithms. Vesanto and Alhoniemi (2000) suggest that the main benefit of this approach is computational efficiency, however they also state that noise reduction is an additional benefit.

An alternative approach is to set up a one-dimensional SOM with the same number of neurons as the number of expected clusters. After the SOM algorithm has run, each input vector is then assigned to the neuron to which it has the smallest Euclidean distance. Mangiameli et al. (1996) have shown that this approach outperforms a number of direct hierarchical clustering methods. They reported that the advantage is especially pronounced when the data is 'messy'; that is where the data includes imperfections such as high dispersion, outliers, irrelevant information and non-uniform cluster densities. These attributes would be expected to apply to experimental data such as the ones examined in this

work. Bacao et al. (2005) also show this method to outperform k-means clustering and say this is because the k-means method is prone to finding local minima and hence does not properly search the whole data-space.

17.2.2 K-Means (also known as C-Means) Clustering (KM)

KM clustering is a partitive clustering algorithm which aims to minimise intra-cluster distances. The objective function the algorithm attempts to minimise is

$$f = \sum_{i=1}^c \sum_{j=1}^n u_{ij} d_{ij} \quad (2)$$

Under the constraints

$$\sum_{i=1}^c u_{ij} = 1 \quad \text{for all } j = 1, 2, \dots, n \quad (3)$$

Where

$$u_{ij} \in \{0, 1\} \quad (4)$$

This means that every sample will be assigned to one cluster ($u_{ij}=1$ indicates that sample i is a member of cluster j . For this sample, for all other values of j , $u_{ij}=0$). The distances from each sample to their cluster centre is then summed and the centres are adjusted to minimise this value.

A problem with this method is that the final results depend heavily on the initial starting position of the cluster centres. In the implementation of the k-means algorithm used here, the centres are set to randomly chosen data points from the data set. This does not allow the method to be repeatable as different results will be obtained for different runs due to the possibility of finding local minima. An improved result may be achieved by averaging the results from a number of runs of the algorithm, however this approach would still be adversely affected by the results when local minima were found. Hence we do not use this approach here.

17.2.3 Fuzzy C-Means (FCM) Clustering

This approach to clustering is derived from the KM method, however it is based on the idea of partial membership instead of each element being assigned to just

one cluster. It is similar to KM in that it provides partitioning results however it also provides extra information in the form of values of degrees of membership.

The clustering problem can be formulated as follows:

Consider a finite set of elements $X = \{x_1, x_2, \dots, x_n\}$ which exist in p -dimensional Euclidean space $\mathfrak{R}^p$ such that $x_j \in \mathfrak{R}^p, j = 1, 2, \dots, n$. The problem is to partition the data set into c fuzzy sets, where c is a given number of clusters, while reaching a given objective, which is typically to optimise an *objective function*. The result of a fuzzy clustering can be expressed by a *partition matrix* such that

$$U = [u_{ij}]_{i=1..c, j=1..n}, \quad (5)$$

where u_{ij} describes the degree to which element x_i is a member of cluster i . However the assumption $u_{ij} \in \{0,1\}$ for KM is relaxed to $u_{ij} \in [0,1]$ for FCM. This allows the membership values, u , to take any value in the range 0,1; so long as the total membership to all clusters of each element sums to 1, i.e.

$$\sum_{i=1}^c u_{ij} = 1 \quad \text{for all } j = 1, 2, \dots, n. \quad (6)$$

Additionally, each cluster must be non-empty and different from the entire set, i.e.

$$0 < \sum_{j=1}^n u_{ij} < n \quad \text{for all } i = 1, 2, \dots, c. \quad (7)$$

The resulting *objective function* for FCM clustering was developed by Bezdek:

$$J(u_{ij}, v_k) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|x_j - v_i\|^2, \quad m > 1 \quad (8)$$

where v_j can be considered the central vector of the j th cluster. In comparison to the KM objective function, a power m has been introduced to the partition variable u_{ij} . This sets how ‘fuzzy’ the resulting clusters will be. In our implementation we set the value of m to 2; Höppner and Klawonn (2003) describe this value as typical and they discuss the implications of altering it.

The aim of the FCM algorithm is to minimise the objective function. To achieve this, first we differentiate the objective function in (8) with respect to

v_i (for fixed u_{ij} , $i=1,\dots,c$, $j=1,\dots,n$) and to u_{ij} (for fixed v_i , $i=1,\dots,c$), whilst obeying condition (6), and hence obtain

$$v_i = \frac{1}{\sum_{j=1}^n (u_{ij})^m} \sum_{j=1}^n (u_{ij})^m x_j, \quad i = 1, 2, \dots, c, \quad (9)$$

$$u_{ij} = \frac{\left(\frac{1}{\|x_j - v_i\|^2} \right)^{\frac{1}{m-1}}}{\sum_{k=1}^c \left(\frac{1}{\|x_j - v_k\|^2} \right)^{\frac{1}{m-1}}}, \quad i = 1, 2, \dots, c; j = 1, 2, \dots, n. \quad (10)$$

The steps followed by the FCM algorithm are summarised as:

- (1) Select a number of clusters c ($2 \leq c \leq n$) and exponential weight m ($1 < m < \infty$). Choose an initial partition matrix $U^{(0)}$ and a termination criterion ε . Set the iteration index l to 0.
- (2) Calculate the fuzzy cluster centres $\{v_i^{(l)} | i=1, 2, \dots, c\}$ by using $U^{(l)}$ and Eq. (9).
- (3) Calculate the new partition matrix $U^{(l+1)}$ by using $\{v_i^{(l)} | i=1, 2, \dots, c\}$ and Eq. (10).
- (4) Calculate $\Delta = \|U^{(l+1)} - U^{(l)}\| = \max_{i,j} |u_{ij}^{(l+1)} - u_{ij}^{(l)}|$. If $\Delta > \varepsilon$ then set $l = l + 1$ and go to step 2. If $\Delta \leq \varepsilon$, then stop.

(Lin and Lee, 1996)

This process can be described as follows:

- (1) Choose initial cluster centres. This is typically achieved by either setting all centres to the centre of the data-space or by choosing random data points.
- (2) Iteratively adjust the cluster centres while calculating new membership values for all data points.
- (3) Stop the algorithm when a certain error level, ε , has been reached. ε is typically chosen to be 0.001, although using 0.01 works well in most cases (Babuska, 2001).

The degrees of membership corresponds to the possibility that the sample is a member of that cluster. The algorithm works in a similar way to the SOM in that it starts with a set of prototype vectors (one for each cluster centre) which are updated iteratively to try to best represent the data. Should the data be required to be 'crisply' clustered, the result can be 'defuzzified' by assigning each sample to the cluster for which it has the highest membership value. Results from FCM clustering have been shown by Höppner et al. (1999) to be better than those obtained from KM clustering. The reason why a defuzzified FCM clustering result will produce a different result to KM clustering is that an early cluster assignment in the KM method may preclude that sample being considered a member of any other cluster (Garge et al. 2005). The FCM clustering algorithm does not allow this 'premature' assignment as it delays the decision of crisp cluster assignment until the algorithm has finished.

17.2.4 Fuzzy Clustering Validity Indices

The success of the clustering depends heavily on the choice of the expected number of clusters (Babuska, 2001). A number of clustering indices exist which can be used as guidance as to how many clusters are present in a data set. A set of possible indices is presented in Figure 7. The indices were again calculated using Anderson's (1935) well known iris data set. A description of the indices used has been described by Balasko et al. (2005).

The indices should be interpreted as follows:

- (1) For the partition coefficient (PC) and classification entropy (CE), the indices naturally decay and grow with cluster number respectively and so local minima and maxima respectively should be used to find 'good' values. In this case there is a minor 'kink' in both graphs at 4 clusters.
- (2) For the partition index (SC), separation index (S) and Xie and Benn index (XB), a low value indicates a good choice, and where two or more cluster numbers have similar index values, the smaller number of clusters should be chosen. In this case the indices for SC and S do not reduce significantly after 5 clusters, and XB after 6.
- (3) The Dunn (DI) and alternative Dunn (ADI) indices should also be minimised, so the DI indicates evidence of the presence of 3 clusters and ADI indicates 6.

As the evidence of five clusters is supported by two of the indices, one would draw the conclusion that five clusters may be present however this would be asserted with a low level of confidence as the other indices did not support this.

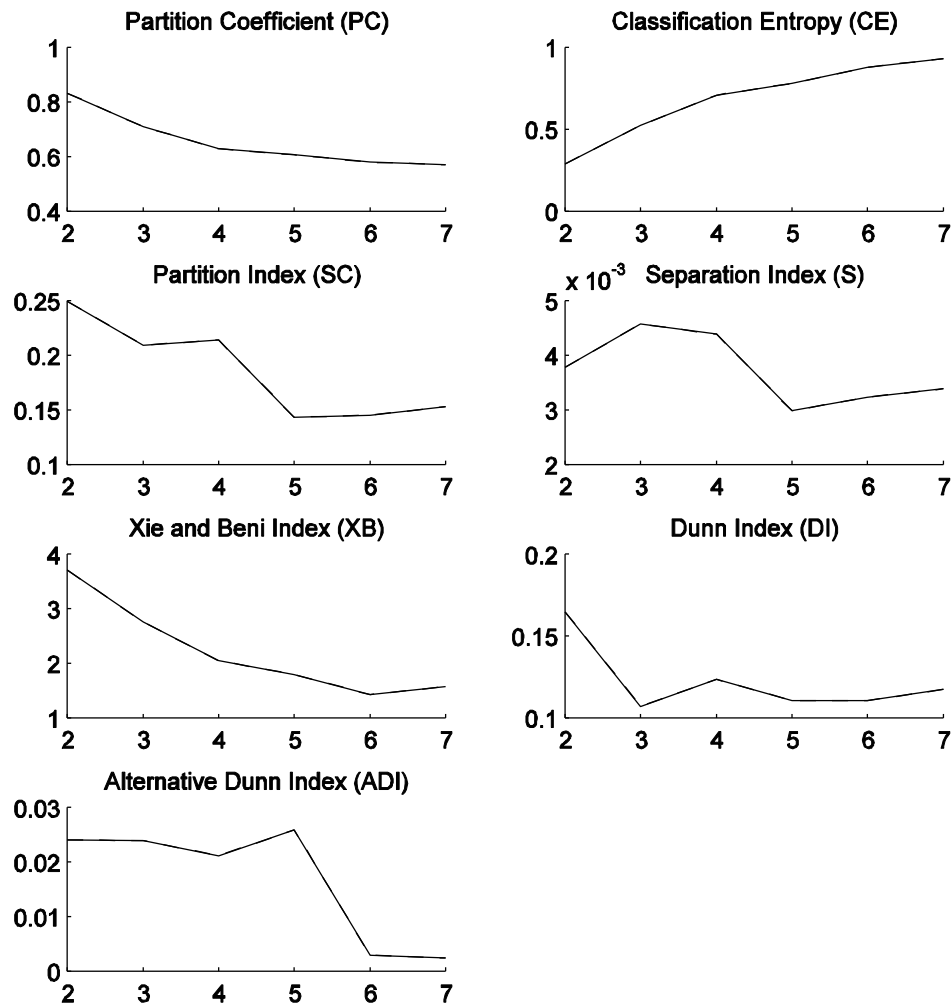


Figure 7: Sample clustering indices showing the number of clusters versus the index.

17.2.5 Cohen's Kappa Coefficient

This measures the agreement above chance between two judges. The coefficient

is given as

$$\kappa = \frac{p_o - p_c}{1 - p_c} \quad \text{Cohen (1960) (11)}$$

Where

p_o = the proportion of units in which judges agreed

p_c = the proportion of units for which agreement is expected by chance

The implication from this coefficient is that the maximum value is +1, which would indicate perfect agreement. A value of zero would show that the agreement observed is exactly that as would be expected by chance.

17.2.6 Collection of Data using a Body Chart

The majority of this study concerns itself with an analysis of inter-rater 'Body Chart' data, so an explanation is provided as to how the data was collected.

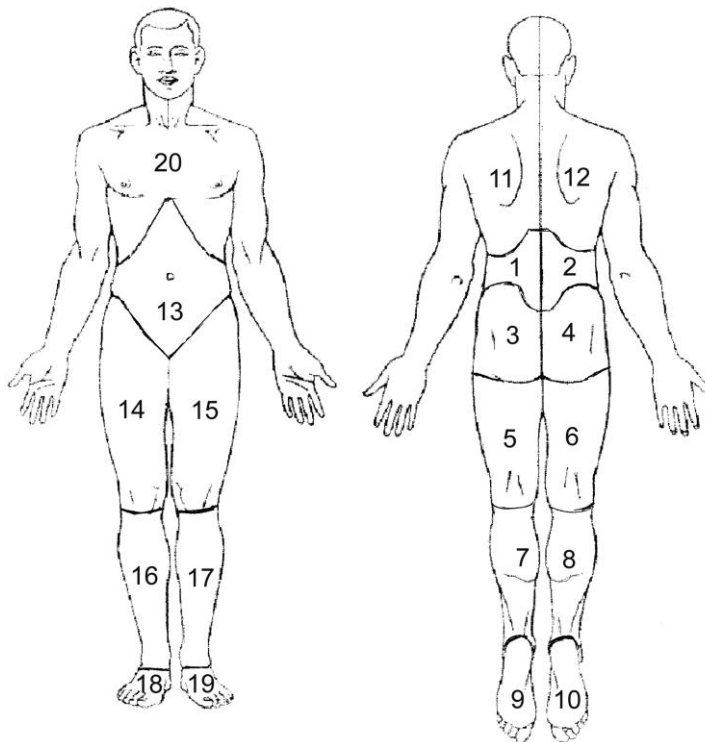


Figure 8: Body Chart used to show in which areas patients feel pain.

The chart (see Figure 8) consists of both an anterior and posterior view of a body which has been divided into 20 regions. The chart is completed by a trained physiotherapist who indicates in which areas the patient feels pain by shading that corresponding area of the chart. As this is an inter-rater study, the patient is then examined by a second physiotherapist. The two examinations are carried out a few hours apart on the same day. This provides two data sets for each patient. The two sets of data were then randomly assigned to be either a member of 'set 1' or 'set 2'; this was to remove any bias caused by individual therapists or time of day or the patient remembering their answers to the previous examination a couple of hours earlier. Throughout this work, the data from set one will be referred to as 'Body Chart data, set 1' or 'BC1' and the data from set 2 as 'Body Chart data, set 2' or 'BC2'.

This availability of inter-rater data allows a more rigorous analysis to be carried out as the agreement between results from each rater can be assessed.

17.3 METHODOLOGY

17.3.1 Outline of Approach of 'two-level' Clustering

In clustering a SOM and extracting the patient's 'identity' from the clustering results, the aim is not only to separate the data set into distinct clusters but also to ascertain how many clusters are present in the data set. With this in mind, the SOM will initially be examined qualitatively. The SOM for each rater's data will then be clustered using both KM and FCM clustering algorithms. The patients will be extracted from each cluster found and the agreement between the two raters will be determined using Cohen kappa as described in section "Cohen's kappa Coefficient". This will be carried out for differing numbers of clusters. This 'level of agreement' will be used to find the best number of clusters to choose to represent this scenario and also the best clustering algorithm for this scenario.

17.3.2 Conditions Required to Allow Comparison of SOM Clustering Results

The aim of this analysis is to determine whether the two sets of data will create

SOMs, which when clustered, are ‘similar’. This will mean that corresponding clusters in both data sets will be similar shapes and sizes and hence corresponding map units will be assigned to the same cluster. Similarity will be calculated by finding the Cohen’s kappa coefficient of agreement of the cluster assignment between data sets. The SOMs will be clustered with a range of possible numbers of clusters and this similarity measure will then be used to determine which number of clusters is assessed to be the best.

In order for this to be successful, both SOMs will need to have map units in the same area of the map which also have similar prototype vectors. This is because the clustering algorithm uses the prototype vectors of the SOM to carry out the clustering. This can be seen visually in Figure 9 which shows the grid of two SOMs, including the indices of the map units. For the clustering results to be comparable, the prototype vector of map unit 1 of BC1 will have to be most similar to map unit 1 of BC2. If however, map unit 1 of BC1 is most similar to unit 45 of BC2, then although the structure of data sets BC1 and BC2 may be the same, this would not be apparent from the clustering of the map. So for example, units [1, 11, 2, 12] may create a cluster in map BC1, but the equivalent cluster containing units with similar prototype vectors may appear as units [45, 55, 65, 46] in BC2 (shown as grey cells in Figure 9). This situation would render any direct unit to unit comparison meaningless.

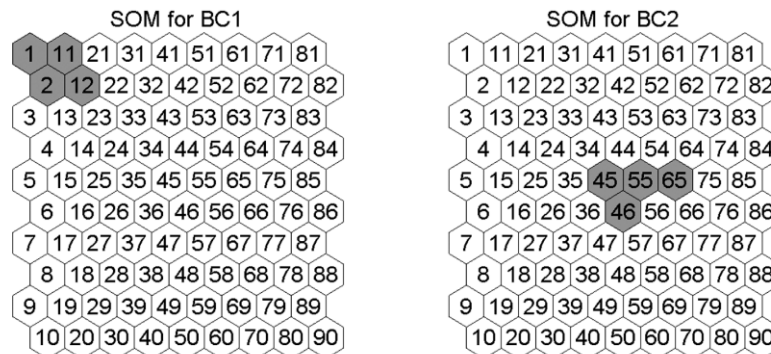


Figure 9: SOM grid showing indices of map units and possible clusters of map units having similar prototype vectors.

In order to prevent this happening, when comparing SOMs, the initial settings for the first SOM will be used to initialise both SOMs. We will describe how this is achieved in the next section.

17.3.3 Initialising SOMs to Allow Direct Comparison of Clustering Results

The aim is to initialise the SOM for body chart 2 (BC2) with the settings used by the SOM for body chart 1 (BC1), hence a direct comparison of the clustering results of the map units will be possible. By default the algorithm determines the optimum size, shape and initial prototype vector values of the map. However these will need to be overridden by ensuring that both data sets use SOMs which have been initialised to the same settings; hence facilitating direct unit to unit comparison.

Let us consider the process in 3 stages:

- (1) Firstly the algorithm decides on the number of map units using the formula

$$m = 5\sqrt{n}, \text{ where } m = \text{number of map units and } n = \text{number of samples}$$

For our data set of 295 patients, this gives an optimum number of map units of 86.

- (2) Secondly the shape (i.e. ratio of width to height) is determined. This is based on the ratio of the two largest eigenvalues of the training data. The eigenvalues are calculated by first finding the auto-correlation matrix of the data set, then finding the two largest eigenvalues and their corresponding eigenvectors for this matrix. As the height and width are forced to this ratio, the number of map units is secondary to this ratio, but is kept as close to the calculated number as possible.
- (3) Finally the initial prototype vectors of the units need to be set. One option is to use randomly initialised vectors; however the algorithm is also able to base the initial weights on the data set to reduce the time it takes to reach an optimum. It does this by initially assigning the mean of each variable to that variable's prototype vector, then linearly adjusting the values across the two eigenvectors of the eigenvalues calculated earlier. To use a dynamics analogy, the two largest eigenvalues indicate the two natural frequencies of the data set which will give the greatest response and the eigenvectors show how the variables will behave at these frequencies. An example initial weighting for one variable is shown in Figure 10.

Clustering SOM agreement can be assessed in two ways:

- (1) The 'best' number of clusters can be determined by evaluating the agreement between the clustering result for each map unit from each rater. Each clustering algorithm will assign each map unit to a cluster, and the agreement

between raters will be compared using Cohen's kappa coefficient.

- (2) Alternatively the agreement can be measured between the clustering result of each patient, after they have been extracted from the SOM.

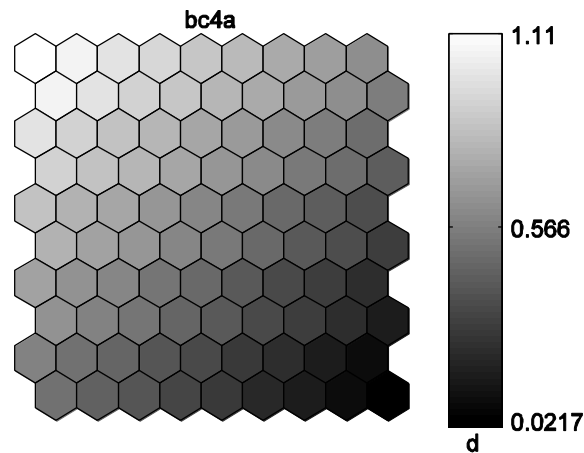


Figure 10: A typical example of an initial weighting for one variable.

17.3.4 Extraction of the Patient's Identity and Corresponding Cluster from the SOM

Once the SOM has been clustered, the patient's identity will need to be extracted from the SOM to determine which cluster the patient has been assigned to.

Let us consider the 4 steps taken to do so in this process:

- (1) Consider a single cluster of the SOM map containing a number of SOM map units.
- (2) For each SOM map unit in this cluster, find all data samples whose BMU is this map unit.
- (3) Assign all data samples found in step 2 to the cluster considered in step 1.
- (4) Return to step 1 until all clusters in the SOM map have been considered.

An example showing how the process is applied to three map units is shown in Figure 11. The diagram shows how data samples (labelled with letters) are assigned to the SOM map unit which is its BMU and then extracted based on the cluster membership of that unit. This example only shows the result for 3 units, whereas in practice, the process is carried out for the whole map.

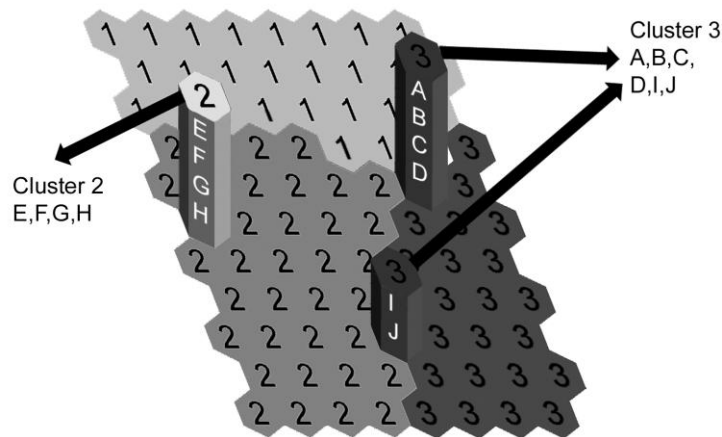


Figure 11: Diagram showing how patients are assigned to clusters using the 'two-level' clustering method.

17.3.5 Methodology Applied to Back Pain Data Set

With a view to improving the current knowledge of NSLBP, the Warwick Medical School (WMS) have conducted a study which involved carrying out an examination of 300 patients with NSLBP by physiotherapists. The study collected demographic data, clinical outcomes to question and physical tests and results of a self-reported questionnaire. The purpose of the study was to investigate the potential of 'unsupervised' statistical methods for identifying clinically recognisable groups where any potential physician bias is removed.

A subset of the data involving the 'Body Chart' was then analysed using the 'two-level' clustering technique described in this section. The aim was to search for homogeneous groups of patients based purely on the areas of the body in which pain was experienced.

17.4 RESULTS

17.4.1 Initial Appraisal of SOM Output

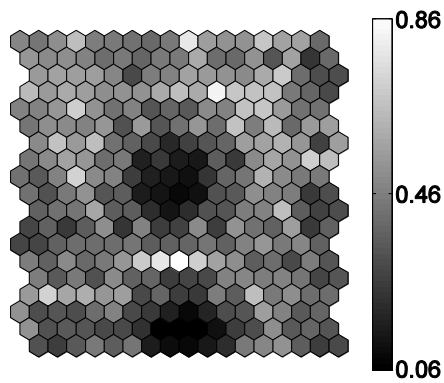


Figure 12: u-matrix of SOM of data set 1

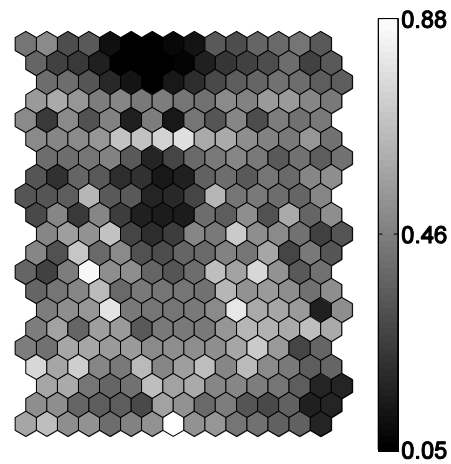


Figure 13: u-matrix of SOM of data set 2

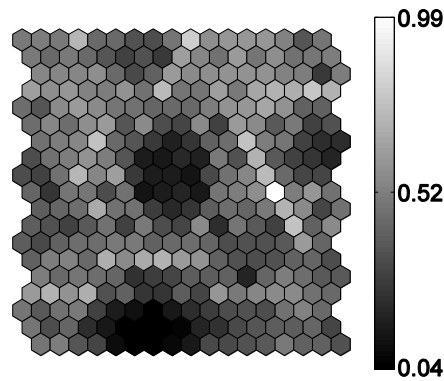


Figure 14: u-matrix of SOM of BC2, initialised
to BC1

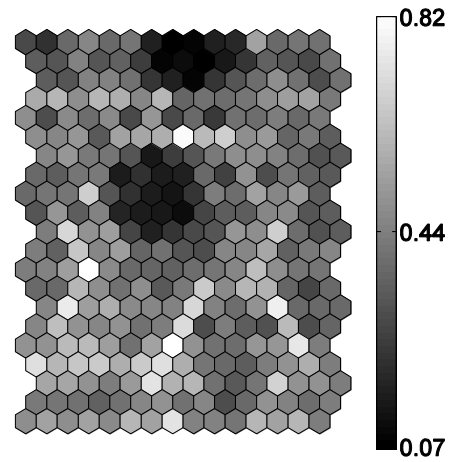


Figure 15: u-matrix of SOM of BC1,
initialised to BC2

A look at the u-matrices of SOMs for the two sets (Figures 12 and 13) can be useful in visualising the structure of the data. They show that both data sets

indicate two clear clusters (areas of black and very dark grey) however it is not clear what is happening in the remainder of the map. Lines of white or pale grey cells indicate cluster boundaries, but the SOM does not provide sufficient information for one to be conclusive about the clustering of the data across the whole map.

An additional way of visualising the SOM is to create pie charts of the neurons in the map. This visualisation contains the prototype vector values of each map unit as a pie chart. Each slice represents the weight of connection from a variable; hence each pie will have 20 slices (since the body chart has 20 regions). An examination of the initial charts showed that 8 out of the 20 variables dominated the charts, so the remaining 12 variables have been coloured black to make the charts easier to evaluate (see Figures 16 and 17). This can also be used to find how many clusters there are by deciding how many different ‘types’ of chart there are. The plots suggest that there are 3 or 4 types of pie chart. These have been indicated on the plots by adding possible region boundaries. Figure 16 shows the charts for the results from BC1 which suggests there are 4 types of cluster – mainly middle back (region 3), some upper back (region 4), mid and low back (region 2), mid back and low back and rear thigh (region 1).

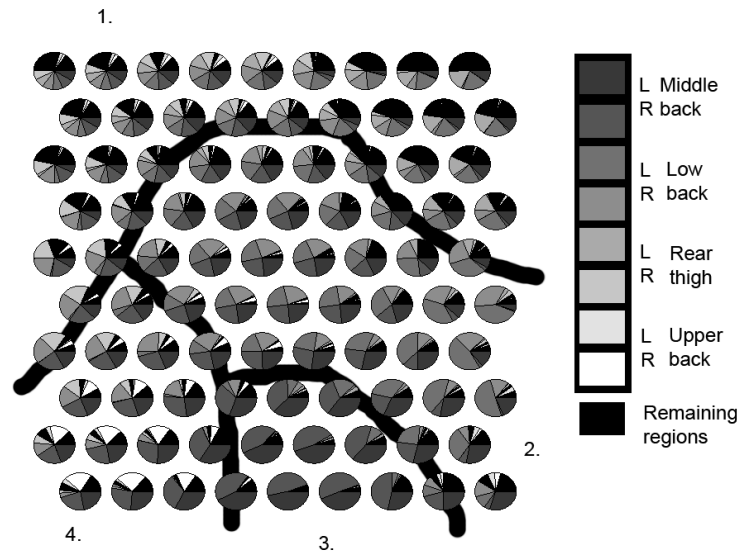


Figure 16: Pie charts for BC1 including suggested clusters.

Figure 17 shows the charts for the results from BC2 which suggests 3 types of cluster — mainly middle back (region 1), mid and low back (region 2), mid back and low back and rear thigh (region 3). In addition there are a handful of units (bottom right) which appear to just be low back.

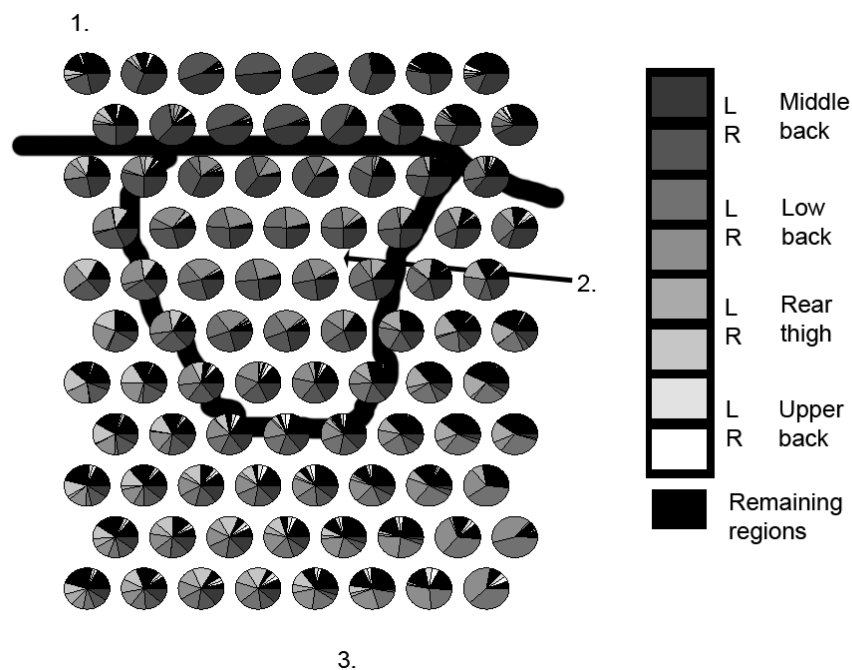


Figure 17: Pie charts for BC2 including suggested clusters.

17.4.2 SOM Clustering

From looking at the u-matrices of the SOMs (Figures 12 and 13), two problems are made immediately obvious if comparisons between maps are to be made, as described in section “Conditions required to allow Comparison of SOM Clustering Results”. Firstly the data is clustered in different areas of the SOM for each data set. The two data sets were expected to contain similar data, as both sets contain data which was taken from the same patients. Had the two data sets been identical, the two clustering results should be very similar, where the only variation is caused by the SOM algorithm. As the two data sets should have some differences, the corresponding clustering should still be similar and it is this

similarity that is being exploited to find ‘good’ numbers of clusters. However Figures 12 and 13 do not suggest that they are at all similar due to the differences in positions of clustering. This suggests that the ‘prototype vectors’ of a SOM map unit in SOM 1 bears no evident relation to the ‘prototype vectors’ of the SOM map unit in the same position in SOM 2. This means that any clustering carried out on both SOMs could not objectively be compared unit by unit.

The second problem is that the SOM maps are different shapes and sizes, which would also make unit to unit comparisons inappropriate. As the size and shape of SOMs are based on the training data, the optimum number of SOM map units for both sets was 86; as they have the same number of samples. However as the shape is based on the eigenvectors of the data set (described in section “Initialising SOMs to Allow Direct Comparison of Clustering Results”), the resulting SOMs were different shapes. The SOM for rater one’s data (BC1) was automatically set to have a [10x9] grid and the SOM of rater two’s data (BC2) was automatically set to have a [11x8] grid. A SOM will therefore need to be set up which is trained on the data from BC2, but where the SOM has the same dimensions and initialisations as the map created from BC1. Having done this, the results for one of the variables (components) can be seen in Figure 18.

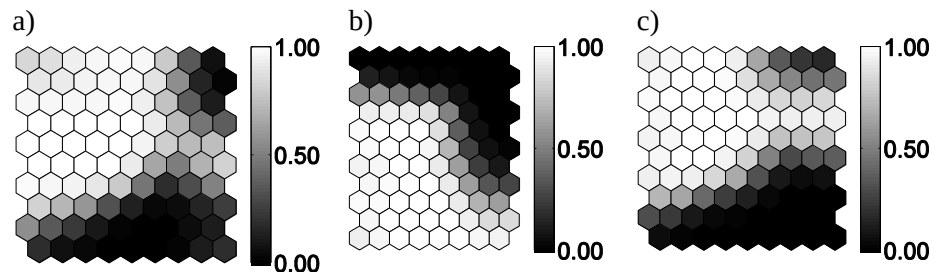


Figure 18: Component plane of variable 4 from the trained SOM of Body Chart data for: a) BC1, b) BC2, c) BC2 initialised with BC1

The reason for the differences in u-matrices is made clear here, that is because the variables are not clustered in the same areas of the map for the two data sets (see Figure 18(a) and (b)). Figure 18(a) and (c) show the results after training from sets 1 and 2 respectively, after each started with the same initial prototype vectors (shown in Figure 10). The results are fairly similar, which is a good initial indication that the two sets contain similar data.

The above process has also been carried out to initialise BC1 to the settings of BC2, resulting in two pairs of SOMs. The two pairs of u-matrices can also be

compared to check for agreement, Figures 12 to 15. The two SOM pairs are very similar with clusters (areas of dark cells) and cluster borders (lines of white/pale grey cells) in similar areas of the SOM.

The approach described has produced two SOMs which have map units which are 'similar' to map units in the same position in the other map; this thus allows direct unit to unit comparison between the maps to be made. It is now possible to carry out clustering on both maps and the results can be sensibly compared. Clustering will be carried out based on the unit's vector prototypes (i.e. connection weights to each variable) for the completed SOMs using both KM and FCM clustering algorithms. The aim of the dual approach is to increase the diversity of types of analysis and the result from the one which is seen to perform best will be used. Both clustering algorithms will be run independently on each SOM and on completion will assign each map unit to a cluster.

17.4.3 Optimal Recoding using Cohen's Kappa Coefficient

In order to make comparisons possible, the numbering of the clustering results of one of the SOM maps will have to be recoded such that the SOM maps correspond; for example the cluster in the top left in both maps will have to have the same number. This will be achieved using an algorithm which calculates Cohen's kappa coefficient for every permutation of one map when the other map is kept constant. The permutation with the highest kappa agreement is then accepted as the optimal recoding. As the time the algorithm takes to execute increases factorially with number of clusters, this is only feasible for this data set for recoding up to seven clusters.

In addition, the analysis outlined above was also carried out to compare BC2 to BC1, which had been initialised to the settings of BC2. The resulting clustering and kappa plots are shown in Figures 19 to 28. The cluster results should be considered in pairs such that both pair members' SOMs were initialised to the same settings, but each member clustered data from SOMs constructed from different data sets (e.g. Figures 19 and 20). In the case that clusters are present, both members should create clusters in similar areas of the map, when the clustering algorithm attempts to find that many clusters. When the clustering algorithm searches for the 'incorrect' number of clusters, the agreement between members should be lower as differences between the two data sets will cause a disproportionate difference in the clustering result. This

effect can be seen for the ‘2 Cluster’ results in Figures 19 and 20. As a comparison, the results for 3 clusters are very similar. This suggests that as both data sets agree about the location of the cluster centres, it is likely that 3 clusters are present. In addition to comparing the cluster results qualitatively, the kappa coefficient between map pairs can be seen in Figures 23 and 28 for maps clustered using the KM and FCM algorithms respectively.

17.4.4 Clustering Performance of Fuzzy Clustering Algorithm

It should be noted that the indices should be used as guides, rather than absolute indicators as described in section “Fuzzy Clustering Validity Indices”.

Figure 29 shows the clustering indices for the SOMs trained on data sets BC1 and BC2 respectively. For many of the indices there appears to be a ‘kink’ at either 3 or 4 clusters, supporting the premise that 3 or 4 clusters are present in each SOM.

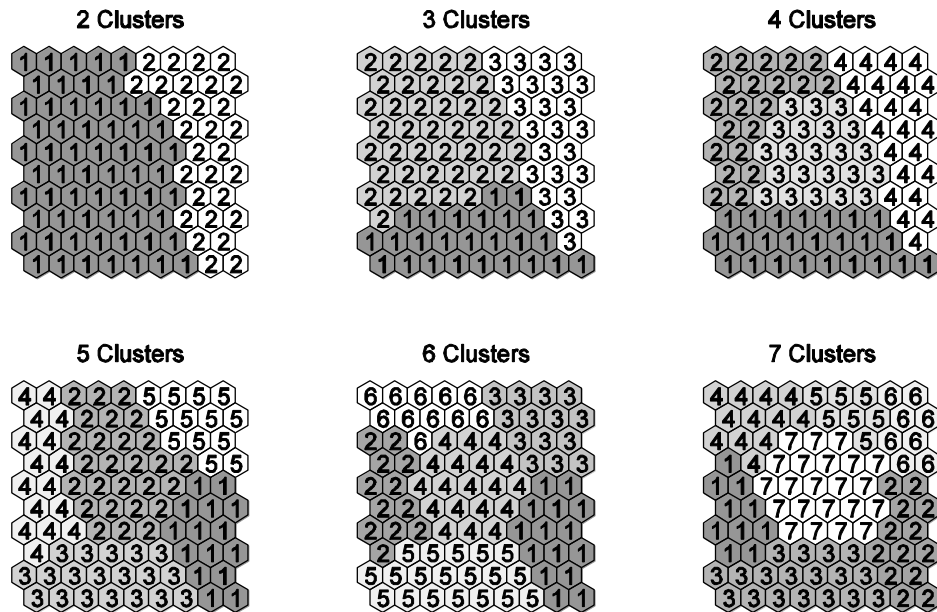


Figure 19: Result from KM clustering of SOM neurons of BC1.

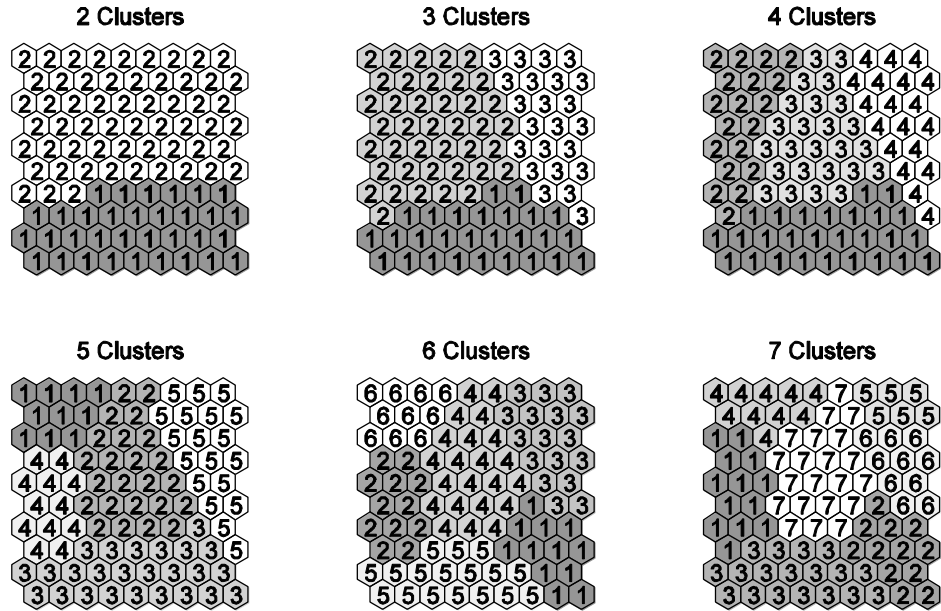


Figure 20: Result from KM clustering of SOM neurons of BC2, initialised to BC1.

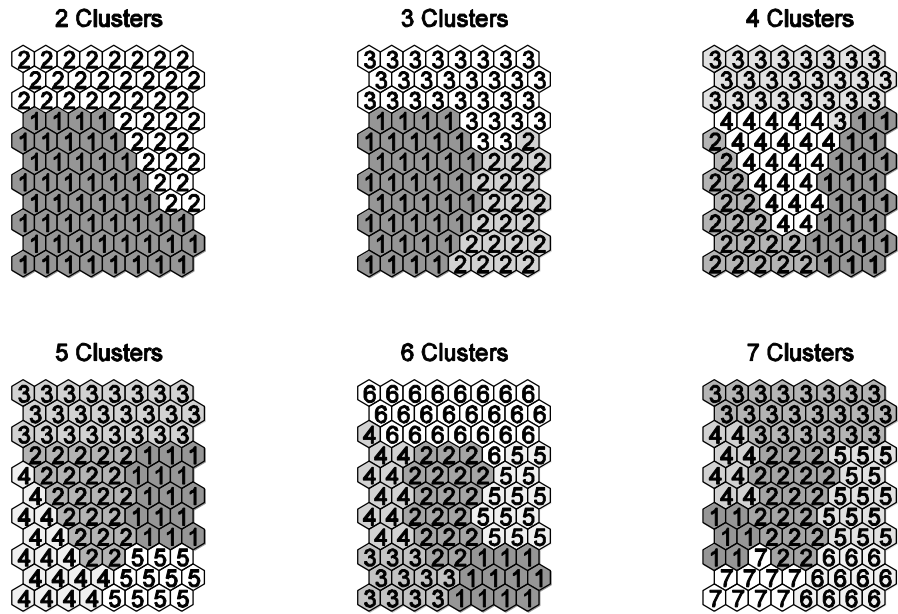


Figure 21: Result from KM clustering of SOM neurons of BC2.

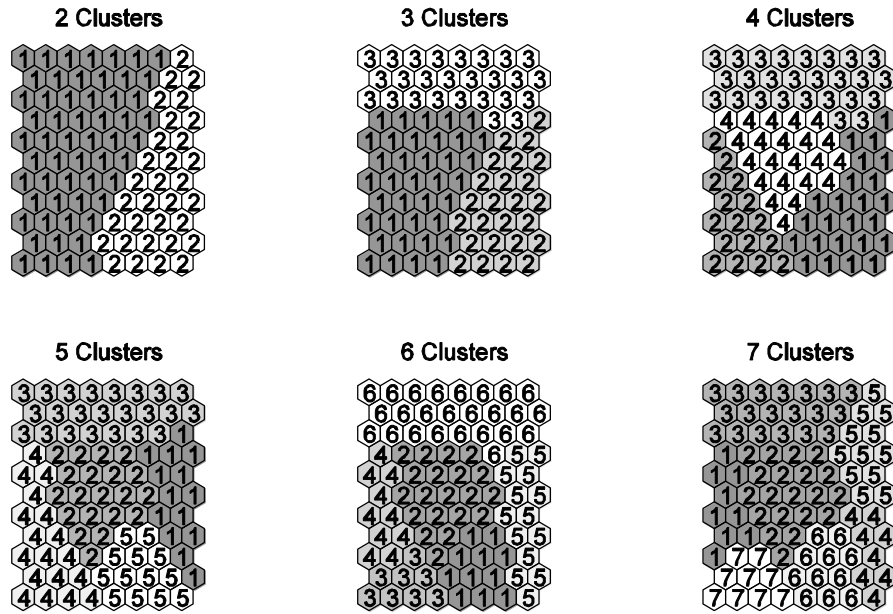


Figure 22: Result from KM clustering of SOM neurons of BC1, initialised to BC2.

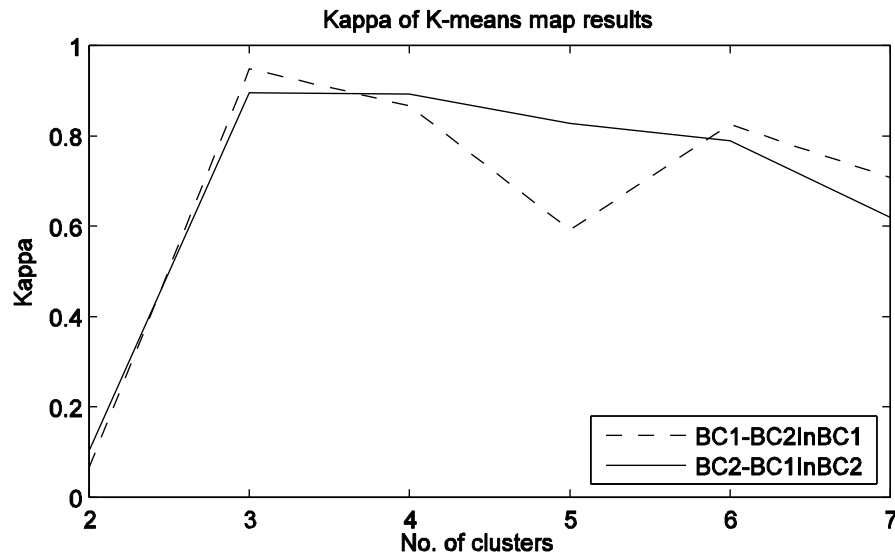


Figure 23: Kappa coefficient (high value shows good agreement) for varying numbers of clusters for both [BC1 – BC2, initialised with BC1] and [BC2 – BC1, initialised with BC2]. KM algorithm used.

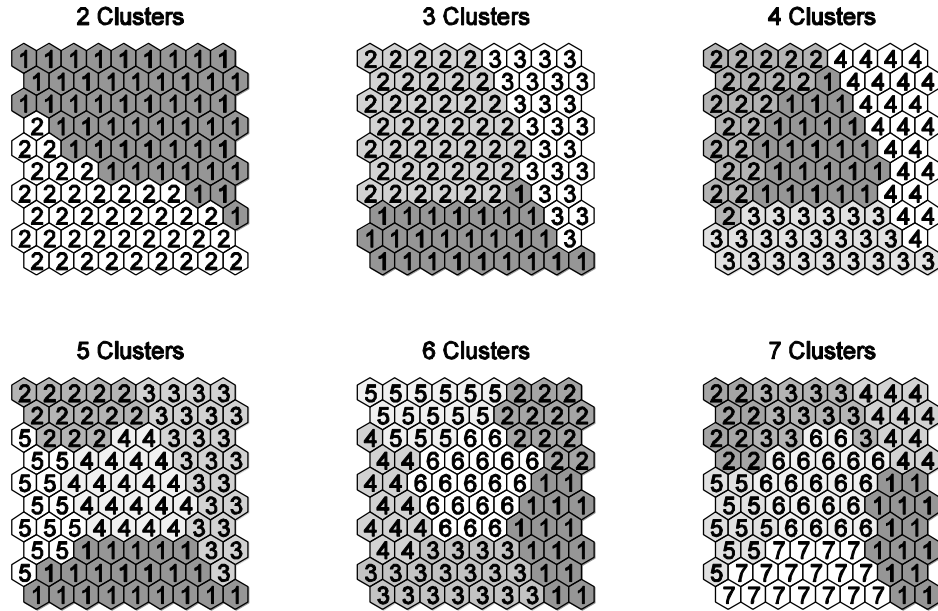


Figure 24: Result from defuzzified FCM clustering of SOM neurons of BC1.

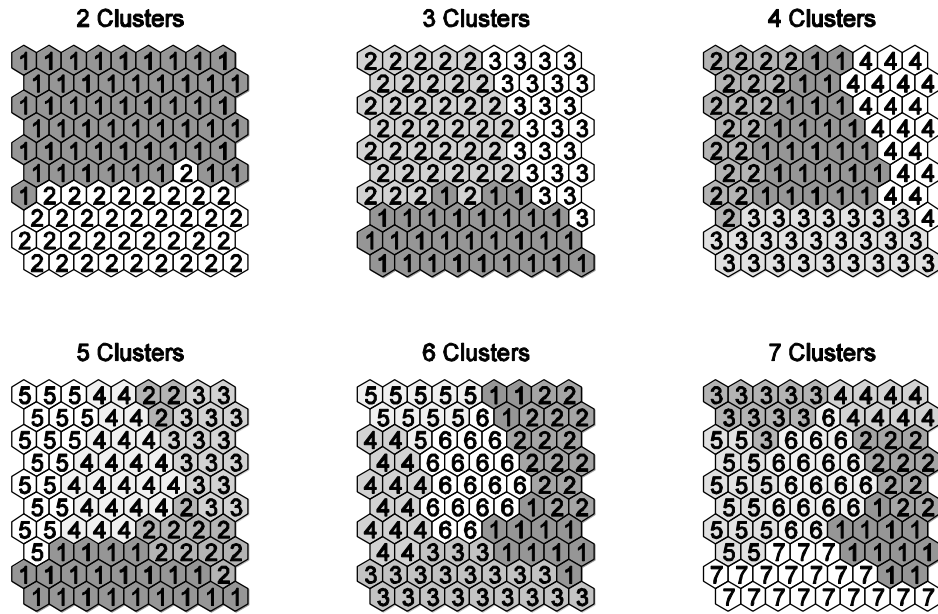


Figure 25: Result from defuzzified FCM clustering of SOM neurons of BC2, initialised to BC1.

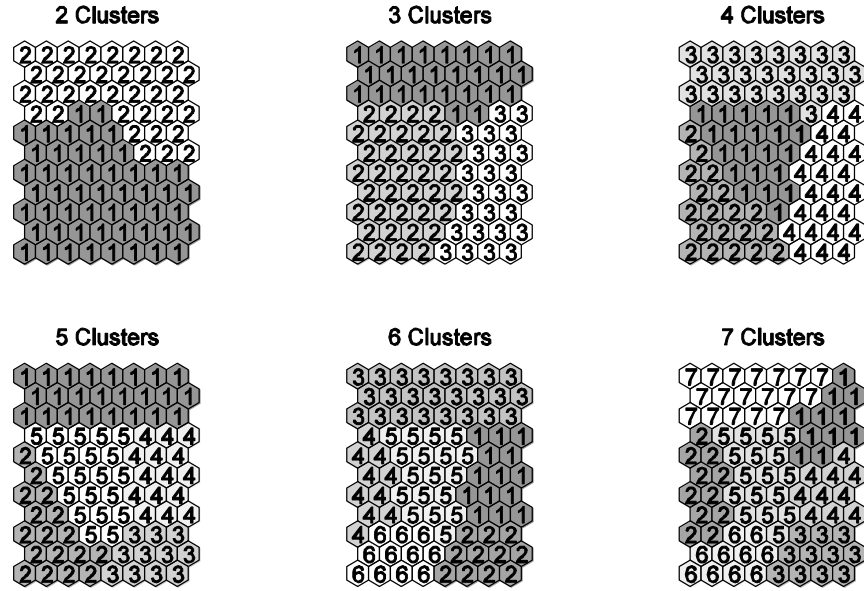


Figure 26: Result from defuzzified FCM clustering of SOM neurons of BC2.

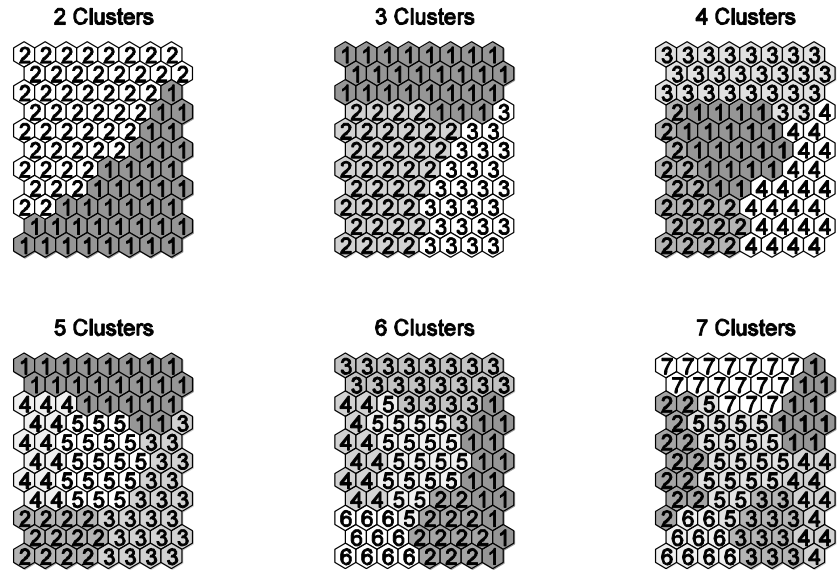


Figure 27: Result from defuzzified FCM clustering of SOM neurons BC1, initialised to of BC2.

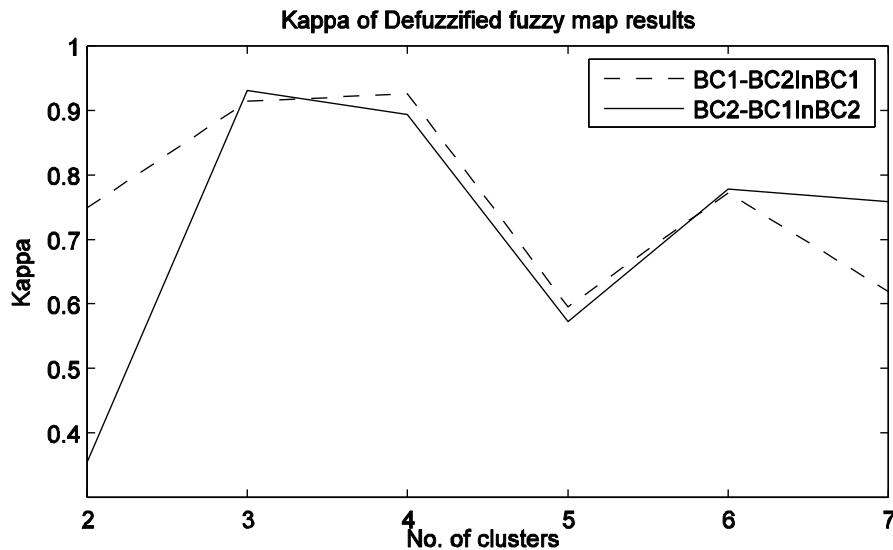


Figure 28: Kappa coefficient for varying numbers of clusters for both [BC1 – BC2, initialised with BC1] and [BC2 – BC1, initialised with BC2]. Defuzzified results from using FCM algorithm used.

The kappa results from both KM and FCM clustering algorithms respectively, indicate the highest agreement for either 3 or 4 clusters; see Figures 23 and 28.

17.4.5 Clustering Indices from ‘extracted patients’

An additional cluster measure can be calculated using the extracted data. The ‘Cohen kappa’ agreement between cluster results from BC1 and BC2 on *patients* for a range of cluster numbers can be examined, see Figures 30 and 31. These results also indicate that the number of clusters used should be 3 or 4 as the highest agreement between BC1 and BC2 occurs at these values.

17.4.6 Examination of Resulting Clusters

The body chart data was only a small subset of the data collected during the study. Using the clusters identified, it is possible to look at other properties of patients who are members of these clusters to look for symptoms of recognised conditions. Staff at the WMS manually explored the data and found that the 3 clusters where the map had been KM clustered created clinically recognisable groups. These were:

- (1) Pain area – back, buttock and leg. These patients have features of hyper-vigilance.
- (2) Pain area – just low back pain. Typical NSLBP group with joint and stiffness pain.
- (3) Pain in the pelvic ‘sacroiliac joint’.

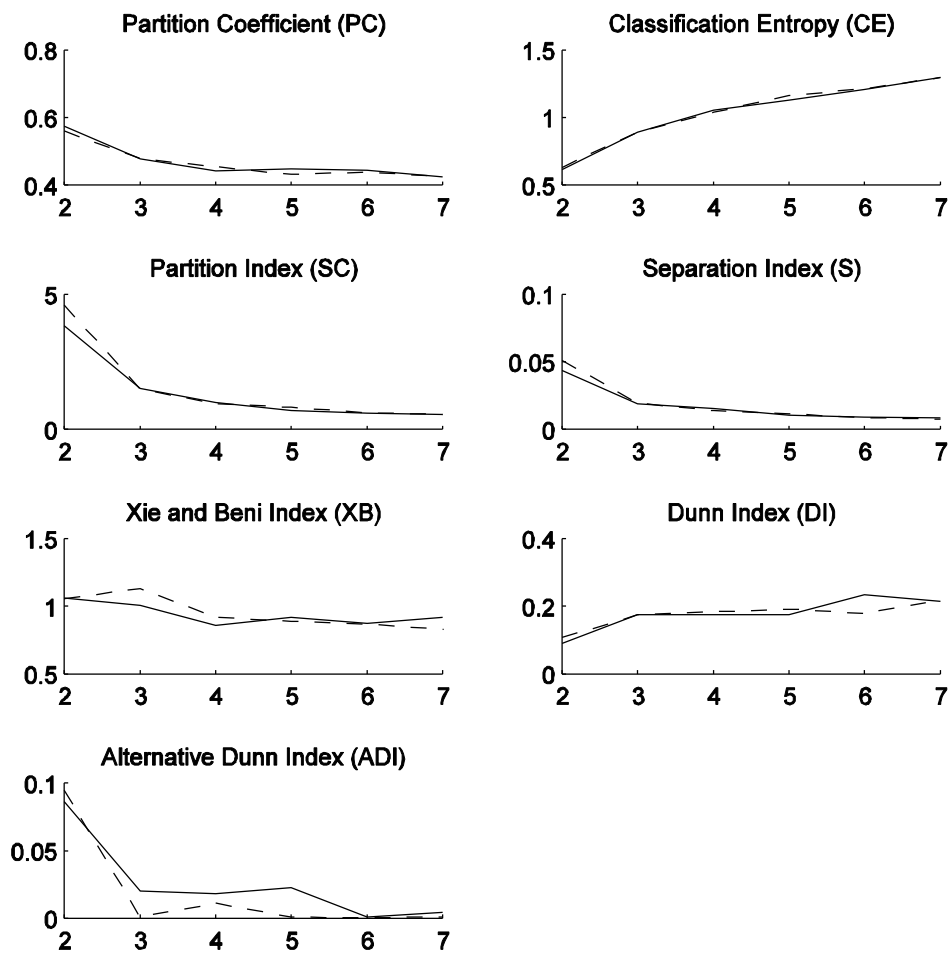


Figure 29: Clustering indices vs number of clusters for data sets BC1 and BC2.

The pain profiles of the three groups are shown in Figure 32. The tone of the pain region represents the percentage of people in that cluster who had pain in that

area, black being 100%, white being 0%. The results from both data sets using the KM clustering algorithm were very similar, so only the profiles created from set 2 are shown. Also the anterior views are not shown as they have very low levels of pain and minimal variation between groups.

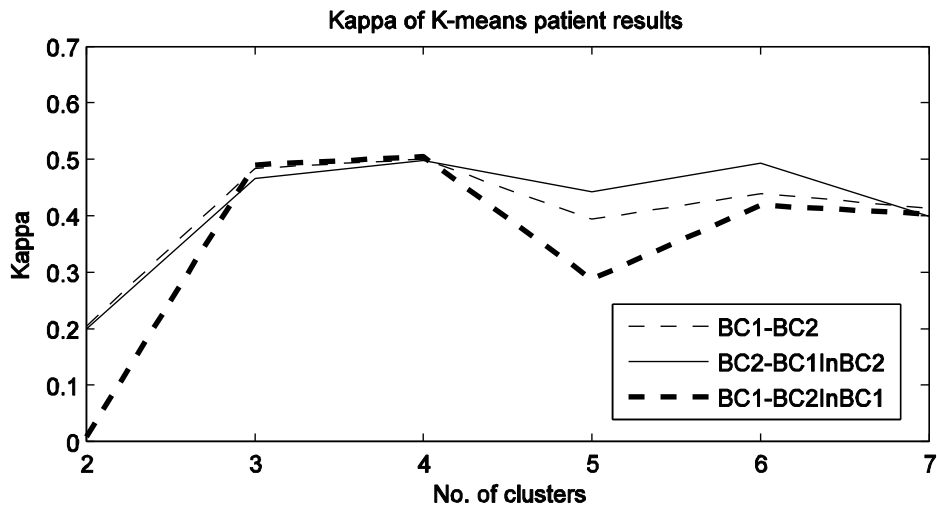


Figure 30: Kappa agreement for patients which have been extracted from a KM clustered SOM for a range of numbers of clusters.

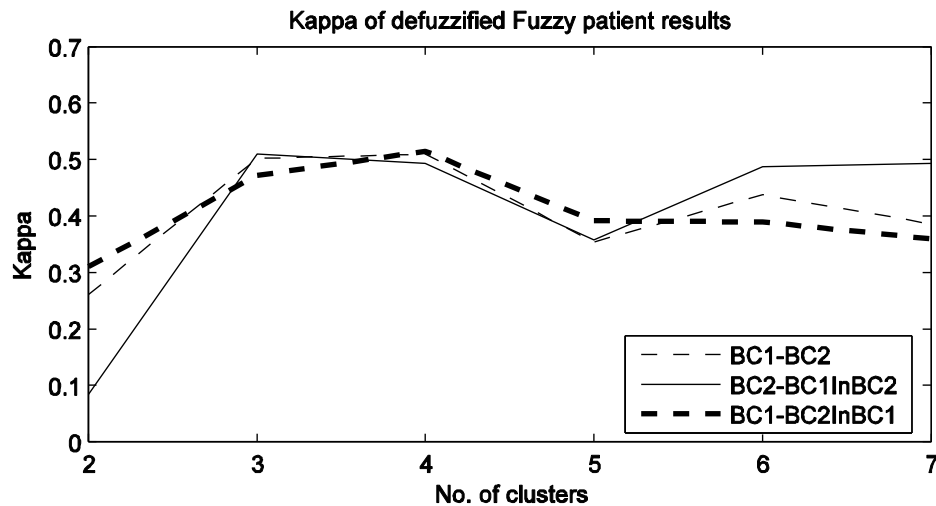


Figure 31: Kappa agreement of patients which have been extracted from a defuzzified fuzzy clustered SOM for a range of numbers of clusters.

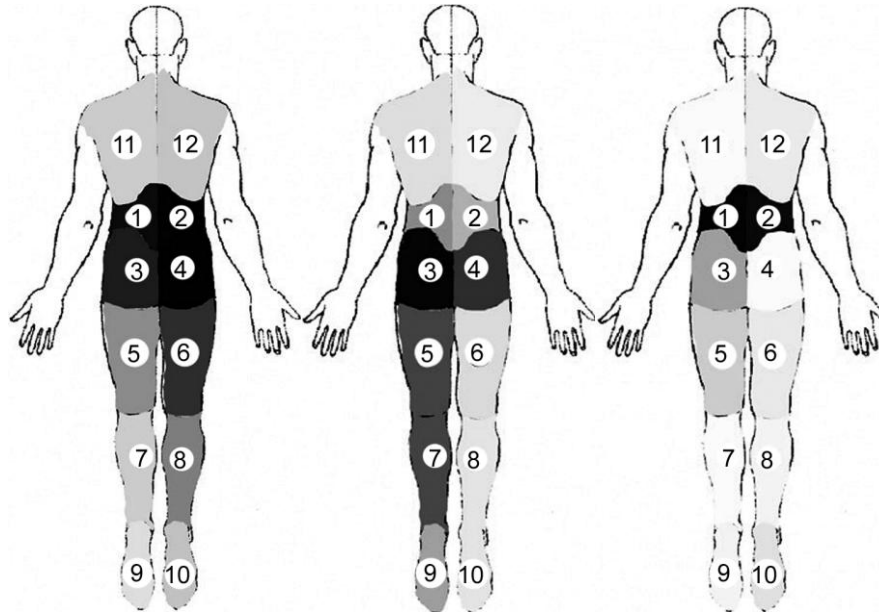


Figure 32: Posterior view of pain profiles of 3 groups identified by KM clustering of data set BC2:
(a) Hypervigilant group, (b) typical NSLBP, (c) sacroiliac pain.

17.4.7 Validation of Results by Comparison with Conventional Clustering Methods

As a means to determine the success of the ‘two-level’ clustering method, when compared to conventional methods, the KM and FCM clustering algorithms were also applied directly to the data. That is, for the ‘two-level’ clustering method, the clustering was carried out on a SOM trained on the patient data, however in this case the clustering is carried out on the patient data directly. The clustering of the patients were also optimally recoded to get best agreement, as described for the SOM maps in section “Optimal Recoding using Cohen’s Kappa Coefficient”. The optimal kappa values of agreement can be seen in Figure 33. This shows that the final agreement for patients clustered between the two raters is better when using the SOM clustering and extraction method when compared to direct clustering of the data for both the KM and FCM clustering algorithms.

17.4.8 'Intelligent' Extraction of 'Patients' from Fuzzy Clustered SOM

Although the fuzzy clustering method outperforms the k-means clustering method when applied to the SOM, a large amount of information is lost when the fuzzy result is defuzzified. By colouring the cells based on the possibilities calculated, the following plots, Figures 34 to 37, show how 'fuzzy' the actual results were.

Each figure shows the defuzzified result followed by a 'fuzzy' plot for each cluster. In the fuzzy plots, the tone represents the degree of membership for each map unit to that cluster, where a black cell indicates a membership of 1 and white indicates 0.

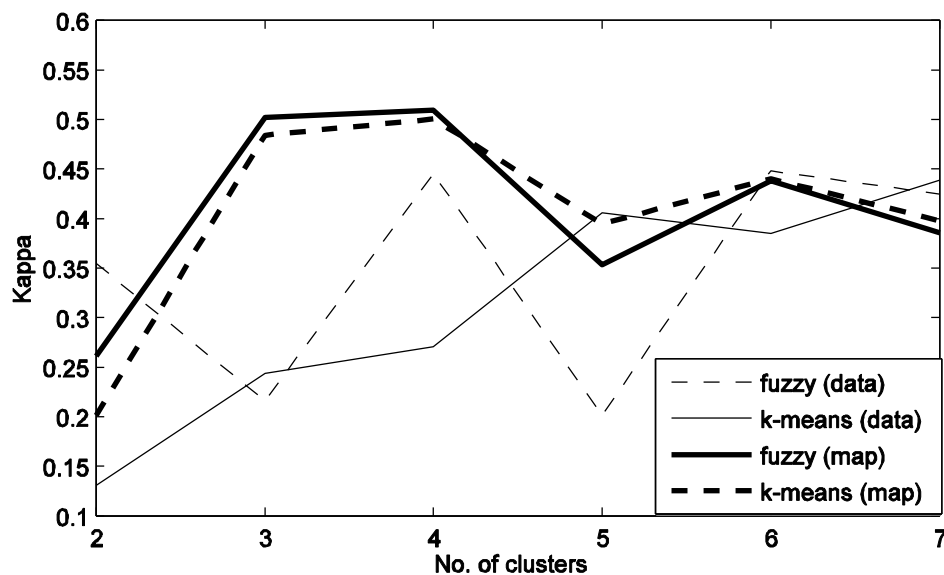


Figure 33: Comparison of kappa agreement of clustered patients from direct clustering of raw data (data) and SOM clustering and extraction (map) methods.

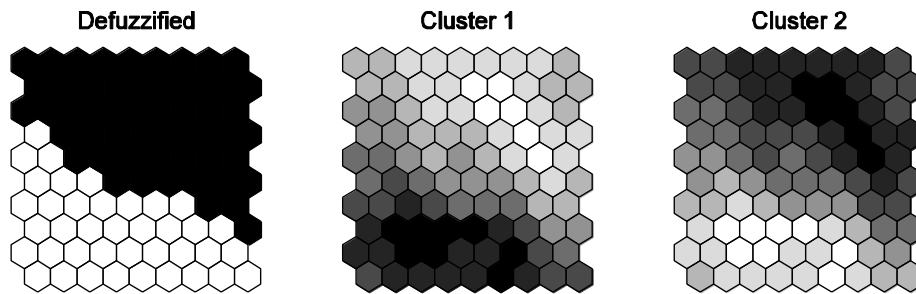


Figure 34: 'Fuzziness' of 2-cluster FCM result for BC1.

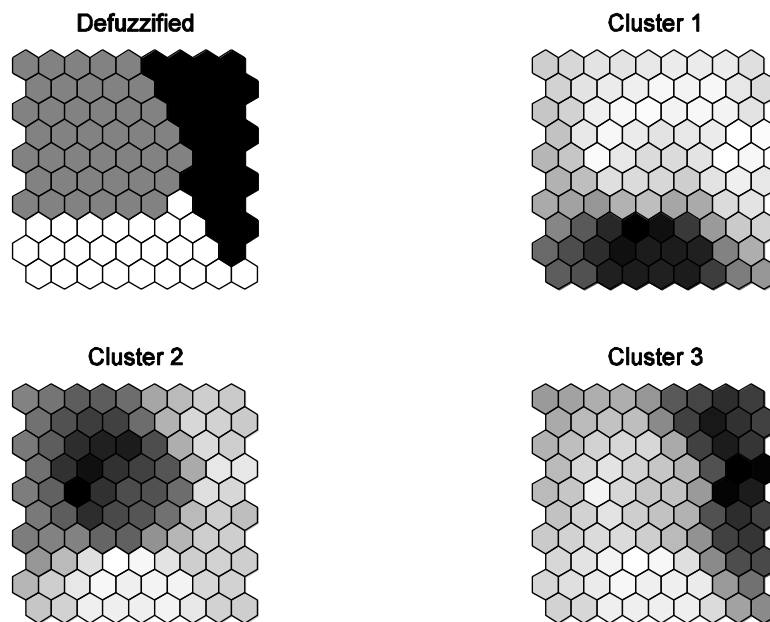


Figure 35: 'Fuzziness' of 3-cluster FCM result for BC1.

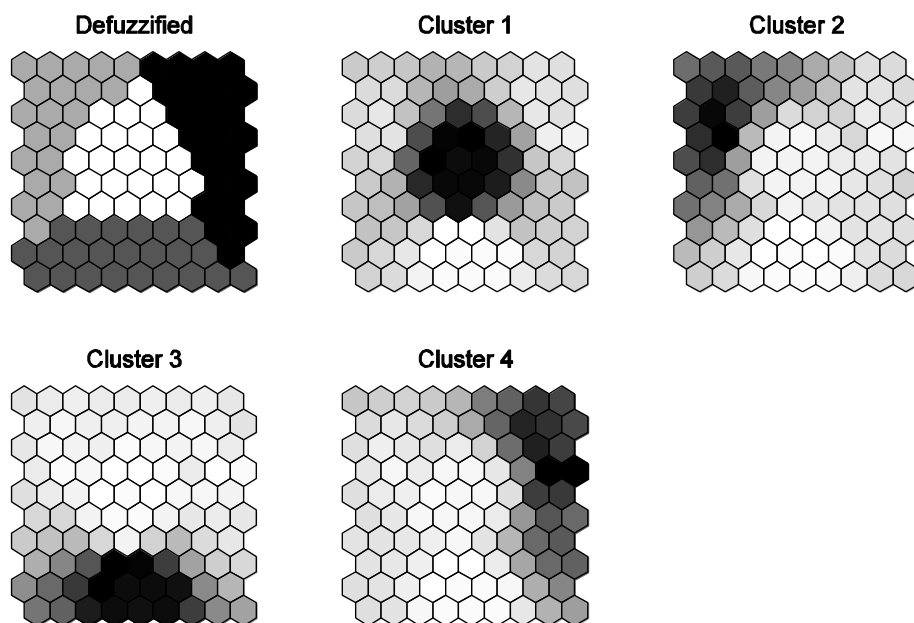


Figure 36: 'Fuzziness' of 4-cluster FCM result for BC1.

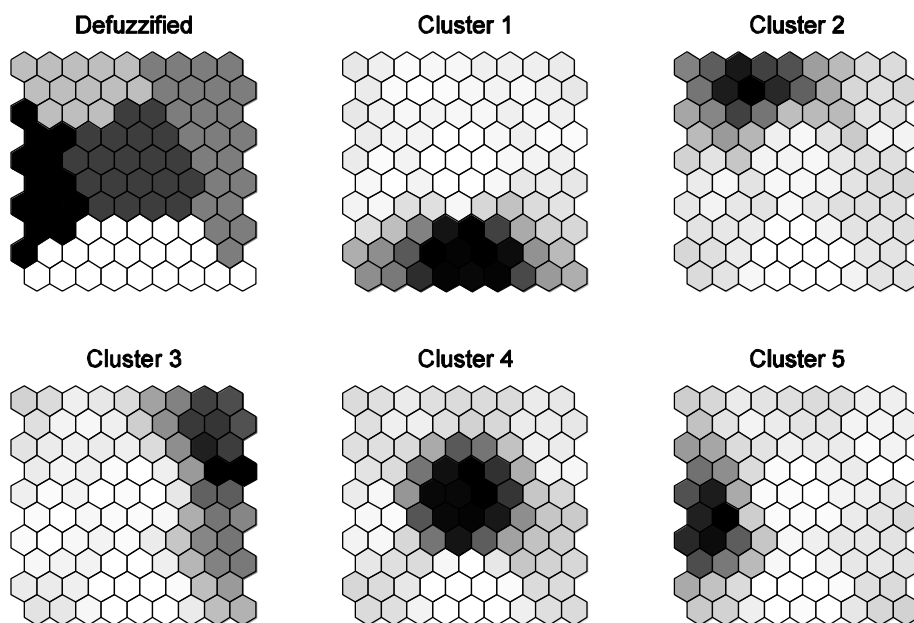


Figure 37: 'Fuzziness' of 5-cluster FCM result for BC1.

To explain the ideas behind how the patients are ‘intelligently’ extracted the clustering result for 2 clusters will be discussed. This scenario was chosen because the cluster memberships are in the simplest state. However it should demonstrate how the process can be adjusted to include more clusters.

When examining, say, the 2 cluster result in Figure 34, the two clusters are not distinct, as there are a large number of neurons which have a similar degree of membership to both cluster 1 and 2. This can be seen where units in the fuzzy plots which correspond to map units near the cluster border on the defuzzified plot have a similar tone in both clusters. As these neurons on the border of clusters clearly do not belong to just one cluster, it suggests that when patients are extracted from these cells, some of them may be being assigned to the ‘wrong’ clusters. A simplified demonstration of this ‘mis-assignment’ is shown in Figure 38(a). The plots use the 8 patients whose BMU was cell 1 of the SOM of BC1 (i.e. top left most cell in Figure 34). They plot the distance from each patient to each cluster centre using the ‘2 Cluster’ result and their resulting cluster memberships. In this case the defuzzified result placed the cell in cluster 2. For this reason, all patients assigned to this map unit were assigned to cluster 2, despite the fact that Figure 38(a) shows that 2 of the patients were in fact closer in Euclidean space to the centre of cluster 1 than the centre of cluster 2. In practice any extraction method must also take into account the ‘possibilistic’ space in addition to the Euclidean space. This is because in reality, clusters may be of non-uniform density and varying size, so a member of a large cluster may in fact be closer in Euclidean space to the centre of the smaller cluster than the larger cluster. However it would be incorrect to assign this member to the smaller cluster.

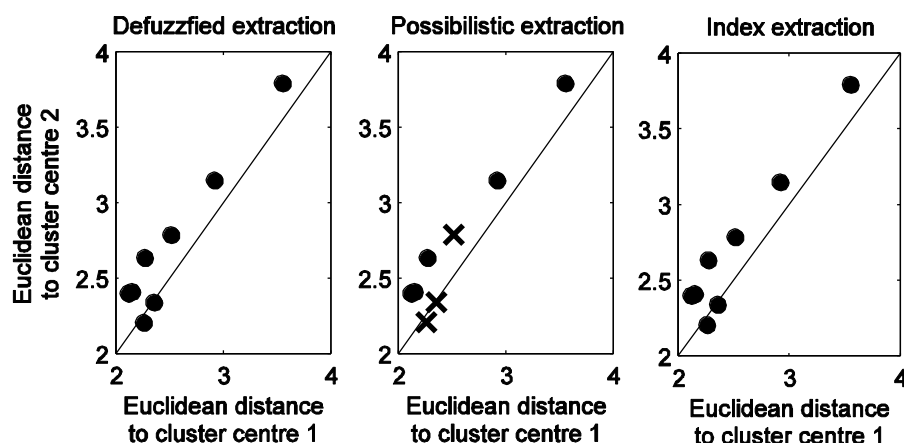


Figure 38: Example of (a) defuzzified, (b) possibilistic and (c) index ‘patient extraction’ methods using cell 1 of map BC1 from the ‘2 Cluster’ result. A cross indicates membership of cluster 1 and a circle of cluster 2.

To overcome this problem, it is suggested that the algorithm be revised to assign patients to clusters depending on the possibilities assigned to that neuron.

One way to achieve this would be to create ‘quotas’ such that the percentage of patients assigned to a neuron related to the possibility that that neuron was a member of that cluster. To use the example of cell 1 of the SOM of BC1 again, the cell was assigned a possibility of being a member of cluster 1 of 42% and a member of cluster 2 of 58%, so 42% (3 patients) of the patients who had that cell as a BMU were assigned to cluster 1 and 58% (5 patients) of them assigned to cluster 2. The result of this process can be seen in Figure 38(b). This process was achieved by measuring the Euclidean distance from each patient in that cell to each cluster centre. This process was then repeated for all neurons in a map. However a problem with this approach is that it assumes that the Euclidean space is directly related to the possibilistic space and as a result assumes that cluster boundaries are equidistant from all cluster centres in Euclidean space as they are in the possibilistic space. This assumption would only be correct if all clusters were of equal size and density and as this is unlikely, the distribution of patients should also involve possibilistic information.

As the clustering algorithm was applied to the map, possibilistic data for individual patients does not exist, other than the data inherited from its parent neuron. As a result of this, an index was devised which consists of the Euclidean

distance from a patient to a cluster centre which has been weighted using the possibility value that the patient's BMU is a member of that cluster. The result of using this index can be seen for cell 1 as before in Figure 38(c). In this case, all patients are assigned to cluster 1, which suggests that the cluster boundary in Euclidean space does not lie in the same space relative to the patients as the cluster boundary in possibilistic space.

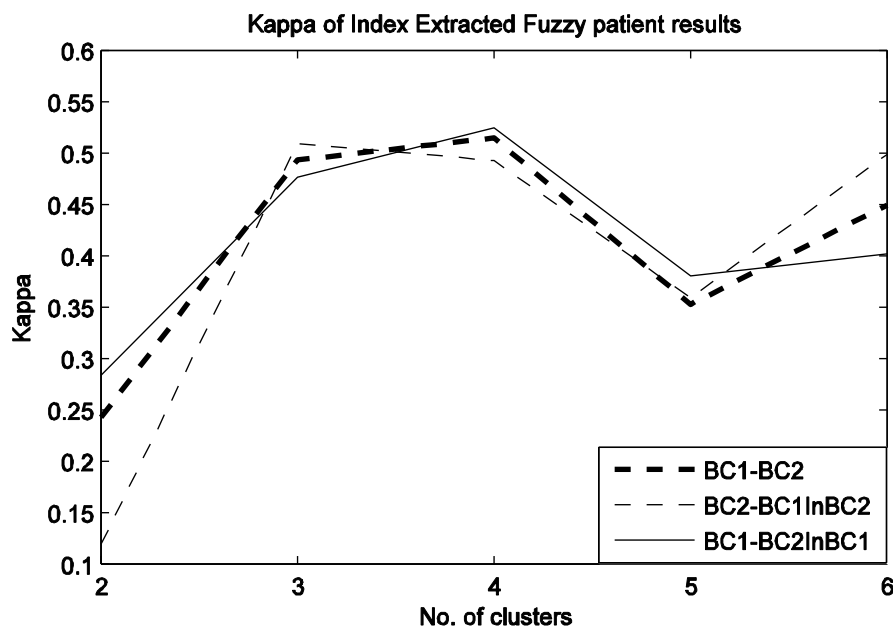


Figure 39: Kappa agreement of patient clustering from 'intelligent' extraction of patients from a SOM.

The idea was implemented and the kappa result for the resulting clustering can be seen in Figure 39.

This result can be considered a success, as the maximum kappa level of agreement has improved when compared to the result using the defuzzified method ($k = 0.5245$ vs $k = 0.5149$, e.g. see Figure 33). The success of the result is heavily dependant on the way the patients are distributed to each cluster from single neurons. A brief description of how the algorithm works follows, carried out for all neurons:

- (1) Find all patients who have this neuron as a BMU.
- (2) Calculate the Euclidean distance from each patient to all cluster centres.

- (3) For each cluster present, create an index for each patient by multiplying a patient's Euclidean distance to the cluster centre by the complement of the possibility that the neuron is a member of this cluster.
- (4) Assign the patient(s) to the cluster(s) for which they have the lowest index.

Note: The reason for using the complement of the possibility ($1-p$, where p is the possibility) is to force the value relating to possibility to increase with distance from a cluster centre. This is to ensure that patients with a low Euclidean distance to a cluster centre, and high membership possibility (and hence low possibility complement) have a low index. If the possibility value were used instead of its complement, these patients would have a similar index to those with a high Euclidean distance but low membership possibility and hence would not be separated.

17.5 CONCLUSIONS

The body chart data strongly supports the existence of 3 homogeneous subgroups. Not only were these groups strongly identified by the ISE techniques but also the groups were subsequently found to have properties of existing clinical NSLBP grouping descriptions. The three groups identified were: typical NSLBP, sufferers of hypervigilance and sufferers with pain in the sacroiliac joint. This demonstrates that the fusion of a SOM neural network and a clustering method such as the KM or FCM algorithms provide a valid method for searching for the presence of clusters within NSLBP data sets. It was also demonstrated that this 'two-level' method of clustering outperformed conventional 'one-level' clustering methods, such as using KM or FCM algorithms in isolation.

Furthermore an exciting novel approach to 'two-level' SOM/FCM clustering has been introduced. This so called 'intelligent extraction' involved combining possibilistic data from the FCM clustering result with Euclidean distance data from a trained SOM map layer to produce a result which was shown to perform better than either the 'two-level' clustering result using the KM algorithm or that using the defuzzified FCM results.

References

- Anderson, E. (1935) "The Irises of the Gaspé Peninsula", *Bull. American Iris Society*, 59:2-5
- Babuska, R. (2001) "Fuzzy and Neural Control DISC Course Lecture Notes (October 2001)" Delft University of Technology, pp64-65.
- Bacao, F., Lobo, V., Painho, M., (2005) "Self-organizing Maps as Substitutes for K-Means Clustering". In: V.S. Sunderam, G. van Albada, P. Sloot, J. J. Dongarra (Eds.), *International Conference on Computational Science 2005. Lecture Notes in Computer Science*, Vol. 3516. Springer-Verlag Berlin Heidelberg, pp476-483
- Balasko, B., Abonyi, J. and Feil, B. (2005) "Fuzzy Clustering and Data Analysis Toolbox For Use with Matlab" Software documentation. University of Veszprem, Hungary, pp13–16.
- Bigos S, Bowyer O, Braen G, et al. (1994) "Acute Low Back Problems in Adults. Clinical Practice Guideline", No. 14, AHCPR Publication No. 95-0642. Rockville, MD: Agency for Health Care Policy and Research, Public Health Service, U.S. Department of Health and Human Services. pp1-160
- Bouter, J., Van Tulder, M. and Koes, B. (1998) "Methodological issues in low back pain research in primary care", *Spine*, 23, pp2014-2020
- Clinical Standards Advisory Group (CSAG) (1994) *Back Pain*, London, HMSO
- Cohen, J. (1960) "A Coefficient of Agreement for Nominal Scales". In *Educational and Psychological Measurement*, Vol 20, No. 1, pp 37-48.
- Delitto, A., Erhard, R. and Bowling, R. (1995) "A treatment based classification approach to low back syndrome: identifying and staging patients for conservative management", *Physical Therapy*, 75, pp470-489
- Fritz, J and George, S. (2000) "The use of a classification approach to identity subgroups of patients with low back pain: inter rater reliability and short term treatment outcomes", *Spine*, 25, pp106-114
- Fritz, J., Brennan, G. and Clifford, S. (2006) "An examination of the reliability of a classification algorithm for sub grouping patients with low back pain", *Spine*, 31 (1), pp77-82
- Fritz, J., Delitto, A. and Erhard, R. (2003) "Comparison of a classification based approach to physical therapy and therapy based on clinical practice guidelines for patients with acute low back pain: a randomised controlled trial", *Spine*, 28, pp1363-1372
- Garge, N. R., Page, G. P., Sprague, A. P., Gorman, B. S. and Allison, D., B (2005) "Reproducible Clusters from Microarray Research: Whither?" In *BMC Bioinformatics 2005*, 6(Suppl 2):S10. Published online
- Hicks, G., Fritz, J. and Delitto, A. (2005) "Preliminary development of a clinical prediction rule for determining which patients with low back pain will respond to a stabilization exercise program", *Archives of Physical Medicine and Rehabilitation*, 86, pp1753-1762
- Höppner, F. and Klawonn, F (2003) "What is Fuzzy About Fuzzy Clustering? -- Understanding and Improving the Concept of the Fuzzifier". In: M.R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, C. Borgelt (eds.): *Advances in Intelligent Data Analysis V.*, Springer, Berlin (pp254-264)

- Höppner, F., Klawonn, F., Kruse, R. and Runkler, T. (1999) "Fuzzy Cluster Analysis", J. Wiley & Sons, Chichester, England, pp189
- Kent, P and Keating, J. (2005) "Classification of Nonspecific Low Back Pain: What Methods do Primary Care Clinicians Currently Use?", *Spine*, 30 (12), pp1433-1440
- Koes, B., Van Tulder, M., Ostelo, R., Kim, B and Waddell, G. (2001) "Clinical guidelines for the management of low back pain in primary care: An international comparison", *Spine*, 26, pp2504-2602
- Kohonen, T. (2001) "Self-Organizing Maps", Third edition, Springer, Berlin
- Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V. and Saarela, S (2000) "Self Organization of a massive document collections". In *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, Vol 11, No 3 pp 574-585, May 2000.
- Leboeuf-Yde, C., Lauritsen, J and Lauritzen, T. (1997) "Why has the search for causes of low back pain largely been nonconclusive?", *Spine*, 22, pp877-881
- Lin, C-T. and Lee, C. S. G (1996) "Neural Fuzzy Systems – A Neuro-Fuzzy Synergism to Intelligent Systems". Prentice Hall, 1st edition, pp187.
- Liszka-Hackzell, J. J. and Martin, D. M. (2002) "Categorization and Analysis of Pain and Activity With Low Back Pain Using a Neural Network Technique." In *Journal of Medical Systems*, Vol. 26, No. 4, pp337-347.
- Long, A. and Donelson, R. (2004) "Does it matter which exercise? A randomised controlled trial for low back pain", *Spine*, 29, pp2593-2602
- Maluf, K., Sahrman, S. and Van Dillen, L. (2000) "Use of a classification system to guide nonsurgical management of a patient with chronic low back pain", *Physical Therapy*, 80, pp1097-1111
- Mandiakis, N. and Gray, A. (2000) "The economic burden of low back pain in the UK", *Pain*, 84, pp95-103
- Mangiameli, P., Chen, S. K., & West, D. (1996) A Comparison of SOM neural network and hierarchical clustering methods. In *European Journal of Operational Research*, Vol 93, pp 402-417.
- McCarthy, C., Arnall, F. and Strimpakos, N (2004) "The biopsychosocial classification of non specific low back pain: a systematic review", *Physical Therapy Reviews*, 9, pp17-30
- McKenzie, R. (1981) "The lumbar spine. Mechanical diagnosis and therapy". Spinal publications, Waikanae New Zealand, pp27-80
- Munro, R. (2003) "Classification and analysis in supervised mixture-modelling" Technical Report, No 536, School of Information Technologies, University of Sydney, NSW, Australia
- Nöcker, M., Mörch, F. and Ultsch, A (2006) "An algorithm for fast and reliable ESOM learning." In *ESANN 2006, 14th European Symposium on Artificial Neural Networks*, Bruges, Belgium, April 26-28, 2006, Proceedings. pp 131-136
- O'Sullivan, P., Phyt, G. and Twomey, L. (1997) "Evaluation of specific stabilization exercise in the treatment of chronic low back pain with radiographic diagnosis of spondylolysis or spondylolitheses", *Spine*, 22, pp2959-2967
- Papageorgiou, A., Croft, P and Ferry, S. (1995) "Estimating the prevalence of low back pain in the general population: evidence from the South Manchester Back Pain Survey", *Spine*, 20, pp1889-1894
- Peterson, T, Laslett, M. and Thorsen, C. (2003) "Diagnostic classification of non specific low back pain. A new system integrating patho-anatomic and clinical categories", *Physiotherapy*

- Theory and Practice, 19, pp213-237
- Riddle, D. (1998) "Classification and low back pain: a review of the literature and critical analysis of selected systems", *Physical Therapy*, 78, pp708-737
- Shlens, J. (2005) "A Tutorial on Principle Component Analysis" version 2. Systems Neurobiology Laboratory, Salk Institute for Biological Studies La Jolla, CA, US. Retrieved August 22, 2007, from <http://www.cs.cmu.edu/~elaw/papers/pca.pdf>
- Vaughn, M. L., Cavill, S. J., Taylor, S. J., Foy, M. A. and Fogg, A. J. B. (2004) "A full explanation facility for a MLP network that classifies low-back-pain patients and for predicting MLP reliability." In Abraham, A., Jain, L. C. and Zwaag, B. J. van der (eds.), *Innovations in Intelligent Systems (Studies in Fuzziness and Soft Computing Series)*, Springer, pp381-412.
- Vesanto, J. (1999) "SOM-Based Data Visualization Methods" In *Intelligent Data Analysis*, Vol 3 No 2, pp 111-126
- Vesanto J. (2000) "Using SOM in Data Mining", Licentiate thesis, Helsinki University of Technology, Espoo Finland
- Vesanto, J. & Alhoniemi, E. (2000) Clustering of the Self-Organizing Map. In *IEEE Transactions on Neural Networks*, Vol 11, Number 3, pp. 586–600.
- Vesanto, J., Himberg, j., Alhoniemi, E. and Parhankangas, J. (2000) "SOM Toolbox for Matlab 5", Software documentation, Helsinki University of Technology, Finland
- Waddell, G. (2004) "The Back Pain Revolution", 2nd edition, London, Churchill Livingstone
- Walker, B., Muller, R and Grant, W. (2004) "Low Back pain in Australian Adults. Prevalence and Associated Disability", *Journal of Manipulative and Physiological Therapeutics*, 27, (4), pp238-244
- Walsh, K., Cruddas, M and Coggon, D. (1992) "Low back pain in eight areas of Britain", *Journal Epidemiology Community Health*, 46, pp227-230
- Windham, M. P. (1981) "Cluster Validity for Fuzzy Clustering Algorithms". In *Fuzzy Sets and Systems Vol 5*, North-Holland Publishing Company, pp177-185

Chapter 18

Learning the Relevant Features of Ventricular Fibrillation from Automatic Detection Algorithms

Felipe Alonso-Atienza, Alfredo Rosado-Muñoz, José Luis Rojo-Álvarez and
Gustavo Camps-Valls

18.1 INTRODUCTION

Ventricular fibrillation (VF) is a pathological process among the ventricular arrhythmias. These, in turn, are included in the general set of cardiac arrhythmias. Any type of cardiac arrhythmia produces alterations in the cardiac rhythm and its process of electrical activation. Although the origin of the arrhythmias may be originated by different causes, all of them can be seen as a consequence of alterations in the cellular electrophysiology of the heart. Ventricular arrhythmias are frequently observed and can either be well-tolerated or produce a blood pumping threat (Macfarlane & Veitch, 1989). Even the well-tolerated ventricular fibrillation can eventually produce sudden cardiac death due to ventricular tachycardia and fibrillation. Therefore, the life of the patient is at risk, and a rapid detection of ventricular arrhythmias is essential in order to take appropriate therapeutic actions. Autopsy studies have shown that, in most cases of sudden cardiac death, an arrhythmogenic cardiac disorder is the main cause, with no evidence of pathological abnormalities of the heart, hence proving that VF can unleash a sudden and irreversible cardioelectric degeneration process with lethal consequences (Doolan, Langlois, & Semsarian, 2004). Nonetheless, even though the sudden death of cardiac origin in people not showing signs of any previous pathology is still a usual cause of death, the initial causes have not yet been fully clarified.

In general, VF is generated and supported by multiple disorganized activation fronts, following different re-entry circuits and constantly changing. Initially, the number of circuits is small, and they involve a wide area of the myocardium.

Later, circuits are smaller and the number of wave fronts increases, causing the heart pumping of blood to stop. It seems to be demonstrated that, for the VF to persist, a certain heart area must be fibrillating. Despite of some authors consider that VF is a chaotic and disorganized pathology, not following any well-defined electrical activation pattern, other studies support that a certain degree of organization exist during VF (Jalife, Gray, Morley, & Davidenko, 1998). This last consideration promotes the search of a processing system being capable of performing detection, and even prediction, of VF episodes in patients.

Nowadays, the only effective treatment to revert VF is the electrical defibrillation of the heart (Beck, Pritchard, Giles, & Mensah, 1947). Defibrillation consists in the external application of a high energy electrical stimulus on the torso of the patient. If the patient has been surgically implanted an implantable cardioverter defibrillator (ICD), the electrical discharge is provided by epicardic or endocardic electrodes within the torso and the heart (Mirowski, Mower, & Reid, 1980). Clinical and experimental studies have demonstrated that the success of defibrillation is inversely related to the time interval between the beginning of the VF episode and the application of the electrical discharge (Capucci et al., 2001). This close relation of success in recovery with time was already accepted during the 80's (Yakaitis, Ewy & Otto, 1996), and together with the invention of the microprocessor (Thakor, 1984), they have impelled the development of automatic devices, which are capable of real-time detection of VF without a need for the support or presence of qualified personnel. On the one hand, it is very important to be able to distinguish between fibrillatory and not fibrillatory rhythms (AAMI, 1993), since defibrillation can be harmful when incorrectly applied. On the other hand, defibrillation turns out to be the only method of restoring the normal cardiac rhythm, and hence, the decision of the application of the electrical shock is crucial. The reliability of a VF detector is usually defined according to two indicators: *Sensitivity*, which is the percentage of correct detection when VF exists, and *specificity*, which is the percentage of correct detection of non-VF rhythms. The former is important for detecting all VF episodes when they actually exist, and the later gives the reliability of the detector in terms of false positives, i.e., detection of VF when it does not exist.

Due to the intrinsic characteristics of VF (lack of organization, irregularity, etc.), automatic detection is not a trivial task, especially considering the existence of similar pathologies, such as ventricular tachycardia (VT), where the required therapy is not the same as for VF. Specifically, some types of VT can be recovered by using drugs, and others require a low-energy synchronized electrical

stimulation, the so called cardiac pacing. For those patients without precedents of cardiac pathology in which VF can appear suddenly, the only monitoring system is the electrocardiogram (ECG) recording by using surface leads. Additionally, due to the unpredictable appearance of VF episodes, ECG monitoring is often accomplished by using external emergency equipments in scenarios such as airports, sport arenas, or shopping centers. According to these considerations, the development of robust algorithms for VF detection using surface ECG is an important task, given that it will provide a fast diagnosis to be able to supply the best therapy to the patient in the shortest possible time.

A wide number of studies have searched for adequate parameters that, when measured in the ECG, yield an accurate VF detection. Algorithms for ICD detection have an extremely limiting requirement on simplicity, due to the battery discharge that would require, and hence detection in current devices is mostly made basing on the cardiac cycle length. This is an extremely simple criterion, which yields a high sensitivity at the expense of decreased specificity, but guaranties that shock is not inhibited during a VF episode, hence preserving the patient's security. Aiming to yield enhanced detection capabilities, further research has focused on characterizing additional features from the ECG, by measuring them either in the time domain or in the frequency domain. Additionally, there is no physical model of the heart being detailed enough to explain all the arrhythmia mechanisms and origin, and hence, methods that are capable of learning the relevant features from samples have been explored in this setting. Statistical Learning Algorithms (such as neural networks, clustering, or in general, data mining procedures) have been thoroughly analyzed, aiming to build VF detectors from the knowledge that these methods can extract from databases of VF episodes in a representative number of patients. However, this has led to a high number of suitable features proposed in the literature, and more, given the nonlinear nature of the majority of the machine learning algorithms, the information that is contained in those variables usually remains hidden in a complex equation, which is called a 'black-box' solution. These two facts have raised an increasing interest for the use of variable selection procedures in this setting during the last years. Moreover, given that the nature of VF has not yet been fully established, the usefulness of variable selection methods could go beyond the obtention of more accurate VF detectors, and might help researchers to have more knowledge about the VF initiation and perpetuation mechanisms.

The main objective of this chapter is to give a review on feature selection strategies that have been used for VF detection in automatic detection techniques, focusing on algorithms that learn the relevant information from examples, this is, from patient databases. The outline of the chapter is as follows. In the next section, a general taxonomy of feature selection methods is presented, together with a brief introduction to the most well known algorithms for VF automatic detection. Then, two novel methodologies for feature selection in the VF problem are proposed. Afterwards, experimental results in these two kinds of feature selection methods in a patient database are compared. Finally, a discussion on the topic is included, with emphasis in the open issues to the date on feature selection for VF detection.

18.2 THE FEATURE SELECTION PROBLEM IN VF DETECTION

18.2.1 Filter, Wrapper and Embedded Methods

Recent applications, such as high resolution image processing or spectrometry measurement analysis, motivates the use of supervised learning algorithms to infer knowledge from a large amount of datasets (Guyon, Weston, Barnhill & Vapnik, 2002). Intuitively, the learning task will benefit from increasing the number of available examples. However, the discrimination process is strongly affected by the relevance of the input variables, and consequently the learning procedure can perform badly in situations with many irrelevant examples. Two approaches have emerged in order to cope with this problem, which are called feature extraction (FE) and feature selection (FS) techniques. On the one hand, FE techniques obtain a new dataset with reduced number of features or variables (expressed as combinations of the original ones) that encodes all the information contained in the original data (Liu & Motoda, 1998). On the other hand, the purpose of FS techniques is to find a subset of the input variables that best describes the underlying structure of the data as well or better than the original features (Salcedo-Sanz, Camps-Valls, Perez-Cruz, Sepulveda-Sanchis & Bousoño-Calzon, 2004). The FS approach is, therefore, a particular case of the FE problem.

FS can be applied to both supervised and unsupervised methods. In this chapter, we focus on the supervised classification problem where each training example $\mathbf{x}_i \in R^n$ consists of n explanatory hvariables, also known as *features* or *attributes*. The potential benefits of FS are improving the generalization capacity of the classifier, reducing the computational burden associated with the learning

procedure, and providing a better understanding of the underlying processes that generated the data (Guyon & Elisseeff, 2003). The main drawback of the FS technique is the search of the optimum set of relevant features, since all possible features subset configurations (2^n , with n is the number of variables) need to be evaluated, which is computationally intractable. For this reason, the search in the feature space is conducted heuristically according to some relevance criterion. FS techniques can be divided into three major categories, depending on how they combine the search of the optimum feature subset and the construction of the classification model: Filter methods, wrapper methods, and embedded methods.

Filter methods (Blum & Langley, 1998) evaluate the relevance of each variable by individually examining the intrinsic properties of the data. Variables are ranked according to a predefined relevance score, so that low-scored variables are removed. Selected variables then constitute the input space of the classifier. Examples of filter methods are (Salcedo-Sanz, et al. 2004) χ^2 -test, principal/independent component analysis, mutual information techniques, correlation criteria, Fisher's discriminant scores, or clustering. Filter methods are computationally easy and fast. However, they do not usually take into account the existence of nonlinear relationship among features, and their classification performance can be reduced.

Wrapper methods (Kohavi & John, 1997) use the performance of a (possibly nonlinear) classification algorithm as quality criterion for evaluating the relevant information conveyed by a subset of features, i.e., a search procedure in the whole feature space is defined, and different candidate subsets are scored according to their classification performance. The subset of features which yields the lowest classification error is selected. Using a wrapper method often requires to define: (a) a classification algorithm; (b) a relevance criterion to assess the prediction capacity of a given subset of features; and (c) a searching procedure in the space of all possible subsets of features. Then, for a given classifier, the evaluation of a specific subset is usually performed by cross-validation techniques or by splitting the data between training, validation, and test sets. The (usually heuristic) searching procedures can be divided into two types, namely, randomized and deterministic search methods. Examples of randomized methods are genetic algorithms or simulated annealing (Kohavi & John, 1997). On the other hand, deterministic methods, also called greedy strategies, perform a local search in the feature space and are computationally advantageous and robust against overfitting. The most common deterministic algorithms are forward and backward selection

methods. Starting with an empty set of features, forward selection methods progressively add those variables that lead to the lowest classification error until the prediction performance is not longer improved. Backward selection methods start with the full set of features, and progressively eliminate those variables with the lowest discrimination capacity. Wrapper methods usually outperform filter strategies in terms of classification error. However, they are computationally intense and can suffer from overfitting if reduced data sets are available.

Finally, *embedded methods* combine the training process with the search in the feature space. For the particular case of the so-called nested methods, the search procedure is guided by estimating changes in the objective function (e.g., classifier performance) for different subsets of features. Together with backward and forward selection techniques, nested methods constitute very efficient schemes for FS.

18.2.2 On the Suitable Domain for VF Detection

VF detection has been investigated by means of FE in different signal representations, such as time, frequency, and time-frequency domains. Time-domain methods are mainly based on morphological analysis (Mirowski et al., 1980), Threshold Crossing Interval (TCI) analysis technique (Chen, Clarkson, & Qi, 1996), autocorrelation function (Aubert, Denys, Ector, & De Geest, 1982), adaptive analysis (Finelli, Jenkins, & DiCarlo, 1993), or with (multivariate) autoregressive (AR) models (Ge, Hou, & Xiang, 2007). Results show typically low rates of false positive rejection for certain signals, suggesting that this is a limited working domain. Other temporal FE methods used in the literature consider working with *fractal* features (Wang, Zhu, Thakor, & Xu, 2001), chaos features (Zhang, Zhu, & Gu, 1996), eigenvectors (Ubeyli, 2008), or Lyapunov exponents (Guler & Derya-Ubeyli, 2005). Recently, new approaches for VF detection in the time domain using phase space reconstruction have improved the detection performance (Amann, Tratnig, & Unterkofler, 2007), improving specificity but at the expense of reduction in sensitivity.

Regarding the frequency domain, the ECG signal reveals significant differences between sinus and fibrillatory rhythms. In the case of sinus, the spectrum is wide (up to 25 Hz) with a great number of harmonics, whereas in the case of VF, the spectrum is concentrated in a narrow band between 3 to 9 Hz, containing a maximum value known as the dominant frequency (DF) (Clayton, Murray, & Campbell, 1995). The DF and some other parameters related to the energy measurement in frequency bands have been used for VF detection (Barro, Ruiz, Cabello, & Mira, 1989). It is important to note that the mentioned studies remark

the existing problems in the cases of VF and VT discrimination. Some other studies (Kuo & Dillman, 1978) assume a sinusoidal-like signal for VF, and use their similarity factor as a VF detection statistic.

Lately, a common approach to FE considers the joint time and frequency domains through the use of the discrete wavelet transform (Khadra, Abdallah, & Nashash, 1998). For instance, in (Henriques et al., 2007), time and frequency features are stacked and a simple feature reduction is applied to alleviate the collinearity problem. These problems can be circumvented by using scale-frequency domains like the wavelet transform, as illustrated in (Guler & Derya-Ubeyli, 2005). Certainly, time-frequency distributions have been applied to biological signals in general (Akay, 1998), and to VF signals in particular (Afonso & Tompkins, 1995). These seminal papers focus on extracting some VF characteristics in the time-frequency domain for different distributions, but no detection algorithm is proposed. A similar study is done in (Clayton, Campbell, & Murray, 1997), concluding that low cross-term distributions are appropriate for VF detection. Different parameters and algorithms are proposed in (Rosado et al. 1999, 2001) using the Pseudo Wigner-Ville (PWV) distribution.

However, as noted in (Pardey, 2007), the isolated analysis of temporal, spectral, or time-frequency based parameters is suboptimal. In (Unterkofler, 2005), a comparative analysis was made between individual (temporal and spectral) methods proposed in the literature, using three public databases of ECG recordings with non-selected segments, which showed that the algorithms under study exhibited a performance far below the previously reported. The development of detection algorithms based in the joint consideration of temporal and spectral algorithms has shown its performance in ICD applications (Aliot, Nitzsché, & Ripart, 2004), as well as in predicting the success of defibrillation shock (Jekova, Mougeolle, & Valance, 2004). In this setting, in (Rosado-Muñoz, 2000) several classification schemes were proposed for combining temporal, spectral, and time-frequency characteristics obtained from the ECG for automatic VF detection.

18.2.3 Machine Learning Methods for VF detection

The use of machine learning and data mining methods for VF detection has been successfully presented in several works. Essentially, the proposed systems apply a dimensionality reduction or FS step to the extracted features, and then a (non-linear) classifier is applied. In this subsection, the literature is briefly reviewed.

On the one hand, the potentially huge amount of computed features in a given domain, e.g. AR or wavelet coefficients, gives rise to a high input space dimensionality that must be further reduced. Typically, this is done by using principal component analysis (PCA) (Ceylan & Ozbay, 2007) or correlation analysis (Henriques et al., 2007) as filter methods. Other authors include a clustering step before classification, implemented with k -NN, fuzzy c -means, or self-organizing maps (SOM) (Ozbay, Ceylan, & Karlik, 2006). The same problem occurs when working with body surface potential maps. Here, the high number of spatial features are reduced with the Karhunen-Loève transform of the data matrix of the QRST map (De Ambroggi & Corlan, 2001).

On the other hand, a wide variety of classification methods have been proposed. In (J. Lee et al., 2005), a linear discriminant analysis (LDA) is applied with good accuracy rates. A quadratic approach is used in (Ge et al., 2007). The mostly used approach considers different topologies of artificial neural networks (NN), such as the multilayer perceptron (MLP), radial basis neural networks (RBNN), or probabilistic neural networks (PNN) (Clayton, Murray, & Campbell, 1994; Ceylan & Ozbay, 2007; Ubeyli, 2008). Despite their promising results, it seems clear that their performance is highly dependent on the input parameters. More recently, (Pardey, 2007) proposed the joint use of parameters TCI and CM as the input layer for an artificial neural network, showing an increase in the discrimination capabilities with respect to the individual results reported in the original papers. Given the high number of potential pathologies to be identified, a common approach in the literature consists of developing dedicated classifiers to each one, and then combine them using mixture of neural network experts (Guler & Derya-Ubeyli, 2005; Ubeyli, 2008). Recently, the use of the support vector machine (SVM) classifiers has been presented (Kostka & Tkacz, 2006). In (Osowski, Hoai, & Markiewicz, 2004), a committee of experts formed by several SVMs was proposed for the recognition of 13 heart rhythm types including VF, and confirmed the reliability of the proposed approach. An updated comparison among neural networks and SVM can be found in (Ubeyli, 2008).

18.3 FEATURE SELECTION METHODOLOGIES FOR VF DETECTION

This section is devoted to analyze two novel methodologies for FS in the VF detection problem. Firstly, we review the data mining scheme proposed in (Rosado, Camps-Valls, et al., 2002). Secondly, a novel SVM-based FS is presented (Alonso-Atienza, Camps-Valls, Rosado-Muñoz, Rojo-Álvarez, & García-Alberola, 2006).

18.3.1 VF Feature Selection with Filter Methods

The proposed scheme for VF feature selection relies on the pooled evaluation of both filter and wrapper methods. A common practice to evaluate the (linear) relationship between each of the n input features $\mathbf{x} = \{x^1, x^2, \dots, x^n\}$ and output y , or among pair-wise features (x^i and x^j) is the use of the correlation function. This is a good method to remove redundant features and to evaluate relationships, but fails when working with low number of samples, or when the assumed linear relationship is not present. When data is considered as the realization of random processes, it is possible to compute the relevance of variables with respect to each other by means of the mutual information (MI) function, which is defined as the difference between entropy of x^i and the conditional entropy of x^i given x^j . The MI function is suitable for assessing the information content of features in tasks where methods like the correlation are prone to mistakes. In fact, the MI-function measures a general dependence between features, instead of a linear dependence offered by the correlation function. However, computing the MI requires a large enough dataset and a careful histogram parametrization. Finally, using Fisher's discriminant scores is only appropriate when solving linear problems, which is not the case in VF, in which a complicated (potentially non-linear) relation between extracted features and the pathology is observed. In the proposed scheme, we develop a pool decision based on several standard data mining techniques:

- *Wilks' lambda criterion.* The Wilks' Lambda is a reflectance of variable importance to perform linear discrimination. The smaller the Wilks' Lambda, the more important the variable. An F -test and significant p -value are also provided as measures of the importance of a variable. Therefore, the smallest Wilks' Lambda and the greatest significance relate the most important variables in a set.
- *Principal Component Analysis.* PCA provides a preliminary insight in data distribution by inspecting the most relevant features associated to the highest eigenvalued principal components. PCA is extensively used in variable selection and dimensionality reduction, even though linearity is assumed.
- *Self-Organizing Maps.* Clustering data is a useful technique to identify homogeneous groups of variables or cases. The usual methods are based on calculating geometric distances between patterns such as k -means and hierarchical clustering. This processing pursues that similar input patterns self-organize in the output space. Rather than offering a ranking of relevance variables, the SOM yields qualitative information about the

structure itself of the data. A useful technique for assessing clusters confidence is to inspect the quantization error of a SOM (Kohonen, 2001) with the classical hierarchical Ward cluster algorithm (Michie, Spiegelhalter, & Taylor, 1994).

- *Classification and Regression Trees*. CART are binary decision trees, which have two branches at each internal node. CART provides fast performance and reliable results (Michie et al., 1994), and can find flexible non-linear relations among independent and dependent variables that are encoded in the tree with *if-then* piecewise linear rules. This method does not only yield good performance but also allows us to get qualitative non-linear information about the data structure by analyzing the surrogate and main splits (Rosado, Camps-Valls, et al., 2002).

The diversity of methods and guiding principles permits capturing different aspects and underlying physical processes in the signal, and thus the pooled FS is robust (Rosado, Camps-Valls, et al., 2002). Methodology performance will be shown in Section 4.

18.3.2 VF Feature Selection with a SVM Embedded Method

As previously mentioned SVM classifiers constitute an interesting approach for the detection of VF based on combination of temporal, spectral and time-frequency features, since they have shown an excellent performance in a number of practical applications in terms of minimal classification error probability. This combination of features is likely to be redundant, so the relative relevance of each of the input space features needs to be quantified. A number of procedures have been proposed for input space FS in linear and nonlinear SVM classifiers, but these methods do not provide with a clear cut-off statistical test. In (Alonso-Atienza et al., 2006) we proposed a Gaussian kernel SVM together with non-parametric bootstrap resampling to provide with a distribution independent statistical criterion for FS.

An advantage of SVM classifiers is its nonparametric nature. Given that their optimizing criterion is the maximum margin in the separating hyperplane, they are not sensitive to the statistical distribution of input features. According to this property, the use of the (nonparametric version of the) bootstrap resampling (Efron & Tibshirani, 1998) for creating sequential procedures of input space variable selection, in connection with SVM classifiers, is an interesting issue. In (Rojo-Álvarez, Arenal-Maíz, & Artés-Rodríguez, 2002), bootstrap resampling is proposed for tuning the optimal SVM free parameters in reduced data sets. In (Alonso-Atienza et al., 2006), bootstrap techniques are proposed and tested for

comparing the relevance of disjoint subsets of the input feature space. In this section, we present the basis to provide with a backward input space selection procedure in SVM classifiers based on bootstrap Confidence Intervals (CI). The statistical criterion to be used is the difference in Error Probability P_e between: (1) the input feature space (complete model), and (2) a reduced model that only considers a subset of the input space features. This approach lies in eliminating one by one the irrelevant features of the input space, until a subset of only-significant input variables is present. This will ensure that the performance of the final SVM classifier is not significantly different from the complete model trained classifier.

18.3.2.1 SVM for classification

SVM binary classifier is a sampled-based statistical learning algorithm based on constructing maximum margin separating hyperplanes in a reproducing kernel Hilbert space (Vapnik, 1995). Be $\mathbf{V}$ a set of N observed and labeled data, $\mathbf{V} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in R^n$ and $y_i \in \{-1, +1\}$. Be $\phi(\mathbf{x}_i)$ a nonlinear transformation to a (generally unknown) higher dimensional space R^B , where a separating hyperplane is given by $\langle \phi(\mathbf{x}_i), \mathbf{w} \rangle + b = 0$. We know that $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ is a Mercer's kernel, which allows us to calculate the dot product of pairs of vectors transformed by $\phi(\mathbf{x}_i)$ without explicitly knowing the nonlinear mapping. Two commonly used kernels are the linear, given by $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, and the Gaussian, given by $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$.

Under these conditions, the problem is to minimize

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (1)$$

with respect to $\mathbf{w}, b$, and ξ_i , and constrained to $\xi_i \geq 0$, and to $y_i(\langle \phi(\mathbf{x}_i), \mathbf{w} \rangle + b) - 1 + \xi_i \geq 0$ for $i = 1, \dots, N$, where ξ_i represent the losses, C represents a trade-off between margin and losses; and $(\cdot)$ expresses the dot product. By using the Lagrange Theorem, (1) can be rewritten into its dual form, and then, the problem consists of maximizing

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i y_i \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (2)$$

constrained to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^N \alpha_i y_i = 0$, where α_i are the Lagrange multipliers corresponding to primal constraints. Gaussian kernel width σ and parameter C are free parameters that have to be previously settled. Methods such as cross-validation or bootstrap resampling can be used for this purpose.

18.3.2.2 Bootstrap resampling to estimate SVM performance

In (Alonso-Atienza et al., 2006), a bootstrap resampling based method for FS was proposed, which is here briefly presented according to the principles in (Rojo-Álvarez et al., 2002). A dependence estimation process between pairs of data in a classification problem, where the data are drawn from a joint distribution

$$p(\mathbf{x}, y) \rightarrow \mathbf{V} \quad (3)$$

can be estimated by using a SVM. The estimated coefficients are

$$\hat{\alpha} = [\alpha_1, \dots, \alpha_N] = s(\mathbf{V}, C, \sigma) \quad (4)$$

where $s(\cdot)$ is the operator that accounts for the SVM optimization, and it depends on the data ($\mathbf{V}$) and on the values of C and σ . The empirical risk for the current coefficients is defined as the training error fraction of the machine

$$\hat{R}_{emp} = t(\hat{\alpha}, \mathbf{V}) \quad (5)$$

where $t(\cdot)$ is the operator that represents the empirical risk estimation.

A *bootstrap resample* is a new data set obtained from the training set according to the empirical distribution, i.e., it consists of sampling with replacement the observed pairs of data:

$$\hat{p}(\mathbf{x}, y) \rightarrow \mathbf{V}^* = \{(\mathbf{x}_1^*, y_1^*), \dots, (\mathbf{x}_N^*, y_N^*)\} \quad (6)$$

Therefore, $\mathbf{V}^*$ contains elements of $\mathbf{V}$ appearing zero, one, or several times. The resampling process is repeated $b = 1, \dots, B$ times. A partition of $\mathbf{V}$ in terms of resample $\mathbf{V}(b)^*$ is

$$\mathbf{V} = (\mathbf{V}_{in}^*(b), \mathbf{V}_{out}^*(b)) \quad (7)$$

where $\mathbf{V}_{in}^*(b)$ is the subset of samples included in resample b , and $\mathbf{V}_{out}^*(b)$ is the subset of non-included samples. SVM coefficients for each resample are given by

$$\hat{\alpha}^* = s(\mathbf{V}_{in}^*(b), C, \sigma) \quad (8)$$

The actual risk estimation for the resample is known as its *bootstrap replicate*, and can be obtained by taking

$$\hat{R}^*(b) = t(\hat{\alpha}^*, \mathbf{V}_{out}^*(b)) \quad (9)$$

Therefore, its normalized histogram for the B resamples approximates the empirical risk density function. A proper choice for B is typically from 500 to 1000 resamples.

Now, we consider a reduced version of the observed data ($\mathbf{W}_u$), in which the u^{th} variable is removed in all the available observations. If we perform a parallel resampling procedure according to the complete set resampling,

$$\hat{p}(\mathbf{x}, y) \rightarrow \mathbf{W}_u^* = \{(\mathbf{x}_1^*, y_1^*), \dots, (\mathbf{x}_N^*, y_N^*)\}_{u_{removed}} \quad (10)$$

then the bootstrap replication of the actual risk in the incomplete model is

$$\hat{R}_u^*(b) = t(\hat{\alpha}^*, \mathbf{W}_{out}^*(b)) \quad (11)$$

18.3.2.3 Hypothesis testing for ranking variables

Let R_c and R_u be the actual risk of a SVM classifier for the completed model and for the uncompleted model when the u^{th} variable is removed, respectively. Without loss of generality, an adequate risk measurement in a classification task is the classification error probability, denoted by P_e . Under these assumptions, the relevance of variable u can be evaluated by comparing the error probability between the complete feature dataset ($P_{e,c}$) and the uncompleted model ($P_{e,u}$). Then, an hypothesis test can be defined to evaluate the relevance of variable u :

- H_0 : variable u is not relevant;
- H_1 : variable u is relevant.

To assess these statements the following test statistic can be used:

$$\Delta P_e = P_{e,u} - P_{e,c} \quad (12)$$

and according to this statistic, the hypothesis test can be reformulated as:

- $H_0 \equiv \Delta P_e = 0$: variable u is not relevant;
- $H_1 \equiv \Delta P_e \neq 0$: variable u is relevant.

The distribution of this test statistic is generally unknown since the dependence process between pairs of data $p(\mathbf{x}, y)$ is not available, and consequently the exact actual risk cannot be computed. However, as stated before, the performance of a SVM classifier can be estimated by using bootstrap resample techniques. In this case, the test statistic is redefined as:

$$\Delta P_e^*(b) = P_{e,u}^*(b) - P_{e,c}^*(b), b = 1, \dots, B \quad (13)$$

which allows us to quantify changes in the SVM performance due to the elimination of variable u . Using bootstrap methods, the bootstrap distribution of test statistic $\Delta \mathbf{P}_e^*$ can be calculated, from which the CI is obtained. Then, for given significance level, H_0 is accepted if the CI does not overlap 0. Otherwise, H_0 is rejected. This procedure to evaluate the relevance of variables can be considered as an embedded method for FS.

18.3.2.4 The Backward Input Space Feature Selection Algorithm

Note that complex interactions among the input variables can be expected whenever a nonlinear SVM model is built, such as collinearity (for the nonlinear case, co-information or redundant information), irrelevant or noisy variables, and subsets of variables being relevant only when interacting among them. Under these situations, the CI of the test statistic associated to relevant variables may also overlap zero. Nevertheless, even if it has not been defined a statistic associated to the CI, it is still possible to propose a backward selection procedure. In order to eliminate one variable at each step, we can remove the variable either with the widest or with the smallest CI overlapping zero.

The bootstrap-SVM backward selection algorithm is as follows:

- (1) Start with all features of the input space.
- (2) Generate B bootstrap resamples of the complete model $\mathbf{V}$.
- (3) For each bootstrap resample b , calculate the error probability of the complete model $P_{e,c}^*(b)$.
- (4) For feature, as in (3), generate B bootstrap samples of the uncomplete model $\mathbf{W}$.
- (5) For each bootstrap sample and for each feature, calculate the error probability of the uncomplete model $P_{e,r}^*(b)$.
- (6) For each variable, compute the bootstrap statistic:

$$\Delta P_e^*(b) = P_{e,r}^*(b) - P_{e,c}^*(b) \quad (14)$$

and calculate the 95% CI.

- (7) If there is any feature with a negative CI, eliminate that feature. Otherwise, if there is any feature with a zero overlapping CI, then:
 - remove the variable with wider CI overlapping 0, or
 - remove the variable with smaller CI overlapping 0.
- (8) If there is any feature for which:

$$\mathbf{P}_{e,r}^* < \mathbf{P}_e^* \quad (15)$$

then error probability of the complete model is redefined as:

$$\mathbf{P}_e^* = \mathbf{P}_{e,r}^* \quad (16)$$

If every feature has a not zero-overlapping CI, end, otherwise go to (4).

18.3.2.5 Nonlinear model example

We illustrate here the performance of our FS algorithm on a synthetic set of data. Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_7\}$ be a set of random variables where $\mathbf{x}_1$ and $\mathbf{x}_2$ define a XOR problem¹: $x_{1,i} = z + N(0, \sigma_{12})$ and $x_{2,i} = z + N(0, \sigma_{12})$, being z a random variable such as $z \in \{-2, +2\}$ and the probability of $z = 2$ or $z = -2$ is equal, for $i = 1, 2, \dots, N$. From $\mathbf{x}_3$ to $\mathbf{x}_5$ different noisy variables are introduced: $x_{3,i} = N(0, 3.5)$, $x_{4,i} = U(-0.5, 0.5)$ and $x_{5,i} = R(1) - 1$, respectively. Collinearity is introduced with $\mathbf{x}_6$ and $\mathbf{x}_7$, defined as $x_{6,i} = 3(x_{1,i} + x_{2,i}) + N(0, 2)$ and $x_{7,i} = 2(x_{1,i} + x_{2,i})2 + N(0, 2)$, respectively. Together with $\mathbf{x}_1$ and $\mathbf{x}_2$, note that both $\mathbf{x}_6$ and $\mathbf{x}_7$ are also relevant features (in weak sense) (Kohavi & John, 1997), since they contain discriminatory information and therefore they can contribute to classification performance.

Table 1 shows the results of the proposed FS algorithm applied to the nonlinear problem defined by variables $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_7\}$. The performance of our algorithm has been evaluated in three different situations, where the associated error probability for the XOR problem (theoretical P_e) has been tuned according to the

¹We denote $N(\mu, \sigma)$ to be a normal distribution with mean μ and standard deviation σ . We also denote $U(a, b)$ to be a uniform distribution in the interval (a, b) , and $R(r)$ a rayleigh distribution with $r_{rms} = \sqrt{2}\sigma$.

parameter σ_{12} . The algorithm has been run for 10 random trials, so symbol (†) indicates that the selected features were obtained 8/10 times, while symbol (‡) corresponds to a variable (or variables) that has been selected 6/10 times. Otherwise, selected features were obtained 10/10 times. Wider CI criterion performs better than smaller CI in terms of P_e . In the case of wider CI criterion, the optimum subset of features is selected in most of the trials, which leads to the minimum error probability. As the value of σ_{12} increases, performance differences between selection criteria decreases, and in these situation the error probability is close to the theoretic value then manifesting the complexity of the scenario under study.

Table 1: Performance of the FS algorithm in a nonlinear problem ($N = 1000, B = 500$).

	$\sigma_{12} = 0.5$	$\sigma_{12} = 1$	$\sigma_{12} = 2$
Smaller CI	$\mathbf{x}_7$	$(\mathbf{x}_1, \mathbf{x}_2)^\dagger$	$\mathbf{x}_7$
Wider CI	$(\mathbf{x}_1, \mathbf{x}_2)^\dagger$	$(\mathbf{x}_1, \mathbf{x}_2)^\ddagger$	$(\mathbf{x}_1, \mathbf{x}_2)^\dagger$
$\mathbf{P}_{e,u}^*$ smaller CI	$5.6(2.0, 11.0) \cdot 10^{-3}$	$4.6(3.4, 5.9) \cdot 10^{-2}$	$0.32(0.29, 0.35)$
$\mathbf{P}_{e,u}^*$ wider CI	$7.6(0.0, 100.0) \cdot 10^{-5}$	$4.6(3.4, 5.9) \cdot 10^{-2}$	$0.29(0.25, 0.37)$
Theoretic P_e	$6.3 \cdot 10^{-5}$	$4.5 \cdot 10^{-2}$	0.32

It is worth noting that for each value of σ_{12} the SVM free parameters (C , σ) have been obtained only once at the beginning of the algorithm. We have observed that the SVM does not need to be retrained if the values of C and σ are chosen correctly. In fact, we attribute the variability of selected variables presented in Table 1 to the difficult task to appropriately determine the values for all the SVM free parameters.

18.4 EXPERIMENTAL RESULTS

In this section, we examine the proposed FS algorithms in Section 3 in the problem of VF detection. For this purpose, input features were extracted from the PWV distribution calculated on ECG signals. In this extraction procedure, a number of temporal, spectral, and time-frequency parameters have been defined. Based on these parameters we analyze the performance of our FS algorithms. We first study the combined strategy of classical and modern filter methods. Then,

we examine the SVM-based FS algorithm. This section concludes with a comparison between both approaches.

18.4.1 Data collection and pre-processing

Signals employed for training and classification were taken from the AHA ECG Arrhythmia Database (8200 series) (AHA, 1998) and the MIT database (MIT-BIH, 2002). A total of 25 patient recordings were analyzed, each containing an average of 30 minutes (1800 seconds) of continuous ECG, from which approximately 100 minutes corresponded to VF. Data contained normal sinus rhythms and several different pathologies, including those closely related to VF, such as VT and ventricular flutter. These pathologies were classified into four groups, which appeared with different prior probabilities in the database: NORMAL ($p_1=40.25\%$), for normal sinus ECG; VT ($p_2=8.84\%$), for ventricular tachycardia including their variants (regular VT, polymorphic VT or “torsade de pointes”); VF-FLUTTER ($p_3=10.66\%$), for VF signal and flutter, both having the same application therapy (electric shock); and OTHERS ($p_4=40.25\%$), comprising the rest of arrhythmias. It is essential to remark that VT usually appears as a previous stage to VF, and for this reason discrimination between VF and VT is a complex issue.

Segment lengths of 128 samples and 125Hz sampling frequency were used, giving a 1.024s window for the segment analysis. This segment length was chosen to contain at least one QRS complex (if existing in the analyzed signal). A general signal pre-processing was done. Firstly, the mean value of the signal was subtracted. Secondly, a low pass 40 Hz filtering was made, providing us with a free noise signal from the 50 Hz or 60 Hz electrical network interference and other high frequency components that were not relevant for the analysis.

18.4.2 Time-frequency parametrization

A full set of parameters was obtained, mainly in the time-frequency domain. Time-frequency distribution of a signal shows the evolution of its spectral components along time, and therefore it provides with joint information of both time and frequency domains. Based on this time-frequency analysis combined-domain parameters can also be defined. Apart from the set of parameters extracted in the time-frequency domain, two time domain parameters were also used, due to their ability to easily reject non-VF rhythms.

A total of 27 parameters were considered, 25 of them extracted from the PWV distribution and 2 of them obtained directly from the time domain. Some basic statistics of the extracted parameters are shown in Table 2. A full detailed description of parameters can be found in (Rosado et al., 1999; Rosado, Guerrero, Bataller, & Chorro, 2001).

Table 2: Some basic statistics of the calculated parameters.

Variable	Normal	Other	VT	VF-Flutter
MINIMFREC	0.73 ± 0.49	0.63 ± 0.38	0.64 ± 0.35	0.64 ± 0.34
MAXIMFREC	21.9 ± 7.7	20.1 ± 7.3	15.4 ± 7.8	14.1 ± 5.6
PMXFRQ	5.51 ± 3.16	4.01 ± 2.47	2.80 ± 2.00	2.56 ± 1.24
AREA (50%)	133 ± 107	126 ± 96	186 ± 136	173 ± 110
LFREC (50%)	9.47 ± 5.01	7.81 ± 3.68	5.49 ± 3.89	4.66 ± 2.17
LTMP (50%)	13.83 ± 11.65	15.43 ± 12.65	33.81 ± 20.73	34.97 ± 21.80
MINFREC (50%)	3.28 ± 3.35	3.07 ± 3.85	2.44 ± 1.75	2.36 ± 1.55
MAXFREC (50%)	12.76 ± 6.04	10.88 ± 4.65	7.94 ± 4.32	7.03 ± 2.13
NAREAS (50%)	1.46 ± 0.81	1.55 ± 1.59	1.76 ± 0.93	1.64 ± 0.86
DISPERSION (50%)	5.72 ± 6.04	5.96 ± 6.57	8.80 ± 7.88	9.89 ± 7.93
TMY (50%)	158.6 ± 72.2	158.3 ± 62.6	292 ± 124	251 ± 113
TE	$6.5E+08 \pm 1.0E+09$	$1.9E+09 \pm 5.1E+10$	$1.2E+10 \pm 2.0E+11$	$9.9E+08 \pm 1.7E+09$
TEH	$5.8E+07 \pm 1.0E+08$	$3.7E+08 \pm 1.7E+10$	$2.9E+09 \pm 7.2E+10$	$1.9E+07 \pm 1.0E+08$
TEL	$5.0E+08 \pm 7.2E+08$	$1.3E+09 \pm 2.9E+10$	$7.4E+09 \pm 1.0E+11$	$8.7E+08 \pm 1.4E+09$
QTEL	75.4 ± 10.6	76.7 ± 11.6	84.8 ± 9.5	86.0 ± 10.1
QTEH	12.1 ± 9.9	7.9 ± 7.2	4.0 ± 6.0	2.8 ± 4.4
CT8	3.6 ± 1.6	3.8 ± 1.5	6.3 ± 1.3	6.0 ± 1.3
TSNZ	1048 ± 635	1104 ± 638	1596 ± 477	1496 ± 492
TSNZL	686 ± 338	723 ± 332	1250 ± 345	1197 ± 346
TSNZH	172 ± 232	161 ± 226	125 ± 177	104 ± 182
QTL	68.53 ± 9.70	68.86 ± 10.17	79.25 ± 10.01	81.21 ± 10.66
QTH	15.02 ± 10.04	12.30 ± 8.69	6.43 ± 7.61	5.35 ± 7.02
MDL8	93.4 ± 44.3	86.7 ± 39.5	69.2 ± 36.8	64.0 ± 25.4
VDL8	99.2 ± 43.8	87.9 ± 38.6	50.3 ± 29.9	46.7 ± 21.2
CURVE	0.112 ± 0.123	0.134 ± 0.117	0.038 ± 0.202	-0.008 ± 0.208
VR	8.2 ± 6.7	6.0 ± 5.0	1.6 ± 3.4	1.5 ± 1.1
RatioVar	1.6 ± 0.5	1.8 ± 0.5	2.5 ± 0.6	2.7 ± 0.4

18.4.3 Learning Relevant VF Features with Filter Methods

We followed a thorough analysis of the 25 parameters extracted by means of standard data mining techniques: PCA, SOM, and CART. These filter methods allows as to better understand the key factors and drivers of VF.

18.4.3.1 Principal Component Analysis

The analysis of the first three principal components (67% of explained variance) suggested that variables *TE*, *TEL*, *TEH*, and *CURVE* are very poor representations of the data (Figure 1(a)). Accordingly, variables *TSNZL*, *CT8*, *QTL*, *LFREC*, *MAXFREC*, and *LTMP* contain relevant information in the first principal component. However, when working with high inter- and intra-subjects variability, it becomes necessary to assess global PCA results performing individual PCA. Individual PCA is illustrated in Figure 1(b). In this case, we made estimations on individual PCA and extracted “mean” information over the population. Therefore, we can conclude that it is necessary to use, in average, six principal components to explain the 90% of variance and 9 for the 95%. Additionally, we inspected the correlation matrix between eigenvectors as a function of the patient, and found that most of them were highly correlated. This procedure helped us to identify three patients characterized by significantly different features from the rest of the population. Consequently, results on these patients are undertaken in a cautious way.

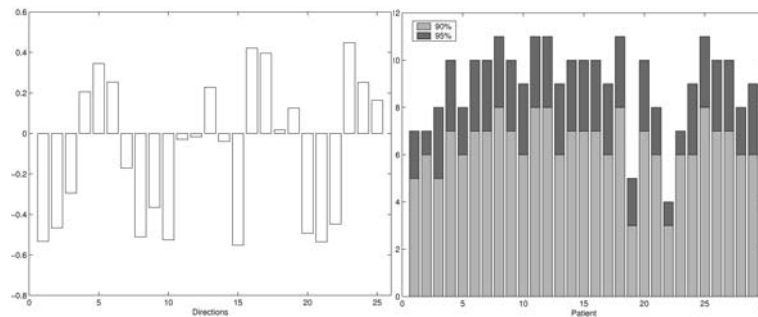


Figure 1: (Left) Eigenvector components of the highest eigenvalue (34% of explained variance). (Right) Number of necessary principal components to explain a given % of variance.

18.4.3.2 Self Organizing Maps

Visualization of SOM clustering determined relevant features and data distributions. The optimal partition was constituted by five clusters. We visualized the SOM clustering for every feature, as illustrated in Figure 2. Several conclusions can be drawn. First, variables TE, TEL and TEH scattered uniformly in all clusters out of VT, which indicates that these parameters do not contain discriminant information except for VT separation. Second, variable TSNZ and xtttQTL traced smooth border lines for a cluster mainly formed with VT and VF, which suggests that these variables contain high discrimination power of those pathologies. Third, variable TEL revealed excellent prognosis capabilities since itself determined cluster #5, which contained the highest number of VT samples. This issue was not captured by any previous statistical method. Finally, it is worth to note that variables AREA and MAXFREC defined specific clusters which were highly correlated with tachycardia episodes (figure not shown).

We additionally analyzed the quantization error to assess the identified clusters. The quantization error is a measure of how good the data vectors from the source data set are matched by a specific node. It is computed by the average of the squared distance of all data records associated with a node. Averaging over the quantization errors of all nodes yields the quantization error of the map. The map is well adapted if the quantization errors are very small and equally distributed over the map. Table 3 relates to quantization and shows the results of this test. We can observe that clusters #1-#4 contained similar and lower variability with respect to cluster #5. This may be due to the presence of badly-fitted VT samples which were mainly associated with #5. However, despite of being a weak boundary in terms of SOM theory, it helped us to identify features associated with VT patterns.

Table 3: Analysis of the quantization error for each cluster.

Cluster	Mean	STD DEVIATION	Max
1	0.073	0.051	0.338
2	0.050	0.044	0.268
3	0.138	0.067	0.437
4	0.106	0.051	0.307
5	0.374	1.139	5.607

18.4.3.3 Classification Trees

Analysis of surrogate and main splits in CART yields valuable information on the relevance of input features. Table 4 shows different ranking of variables according to these measurements and the classical Wilks' Lambda statistical analysis. Similar results are obtained in the parameter relevance list. Confidence on this analysis could be ensured, since classification rates of the best CART achieved recognition rates higher than 83% in all classes, hence suggesting that the underlying differences between classes had been captured.

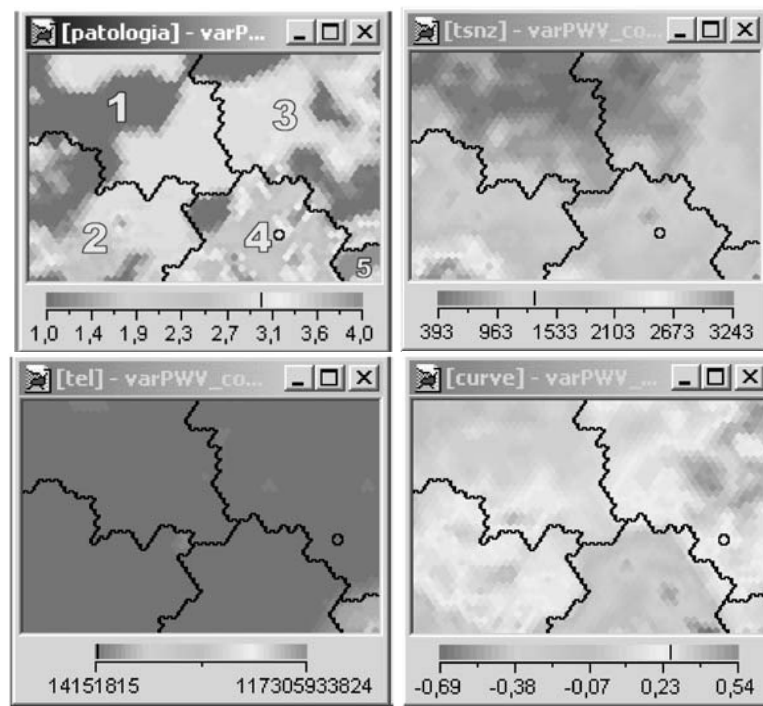


Figure 2: SOM clustering for (top): pathology (left) and TSNZ (right); (bottom): TEL (left) and CURVE (right).

Table 4: Analysis of surrogate and main splits in CART and statistical analysis through Wilks' Lambda.[†]

Variable	Main Splits Score	Variable	Surrogate Splits Score	Variable	Wilks' lambda
TSNZL*	100.00	TSNZL*	100.00	TSNZL*	.671
LTMP*	89.29	LFREC*	63.52	CT8*	.710
CT8	79.93	TE*	43.75	TMY*	.724
MAXFREC	79.80	TEL	32.03	QTL*	.747
TEL	78.18	CURVE*	31.03	VDL8*	.805
TE*	77.69	QTEL	30.21	QTEL*	.807
LFREC*	72.78	MAXIMFREC*	17.47	LTMP*	.807
TMY*	62.33	MAXFREC	17.34	LFREC*	.819
TSNZ	61.63	TEH	16.45	QTEH*	.822
MAXIMFREC*	54.83	QTL*	13.37	MAXFREC*	.834
QTEH	54.25	TMY*	6.53	QTH	.842
QTEL	49.38	PMXFREC*	4.58	CURVE*	.849
QTL*	46.43	MDL8	4.38	PMXFREC	.863
PMXFREC*	39.40	QTEH	4.10	MAXIMFREC	.875
QTH*	36.15	AREA	2.96	TSNZ	.911
TEH	33.80	TSNZ	2.59	DISPERSION	.924
CURVE*	32.49	LTMP*	2.47	MDL8	.937
TSNZH	23.58	QTH*	1.90	NAREAS	.947
MDL8	21.44	VDL8	1.44	MINFREC	.964
MINFREC	21.03	TSNZH	1.37	AREA	.966
NAREAS	18.72	DISPERSION	0.81	TSNZH	.986
VDL8	18.11	MINIMFREC	0.78	MINIMFREC	.988
AREA	7.44	MINFREC	0.71	TEL	.997
MINIMFREC	3.79	NAREAS	0.25	TE*	.998
DISPERSION	1.65	CT8	0.00	TEH	.999

[†] *: Selected Parameters.

18.4.3.4 Combining individual selections

A pool of all methods considered 10 parameters to be significant, namely: TSNZL, LTMP, TE, LFREC, TMY, MAXIMFREC, QTL, PMXFREC, QTH, and CURVE. The decision on kept parameters was taken based on CART, PCA, SOM, Wilks' lambda and correlation analysis. All discrimination models and PCA were developed in MATLAB environment (Mathworks, Inc). SOM and CART processing were carried out by using shareware implementations from <http://www.eudaptics.com/> and <http://www.salford-systems.com/>, respectively. Statistical analysis was carried out with SPSS.

We performed discrimination with all features and with the ones selected in the previous process using the detection tree previously presented in (Rosado et

al., 1999). Patterns were randomly assigned to two sets, two thirds for training (38605 patterns) and the rest for validating (19302 patterns) the discrimination tool. The criterion used to choose the best model was the sum of sensitivity and specificity rates applied to the validation set in order to obtain well-balanced models. Detection using the reduced set of parameters (sensitivity ‘VF-Flutter’ 88.8%, specificity ‘Normal -Other’ 94.9%, specificity ‘VT’ 76.3%) showed better results compared to a detection tree using classical statistical parameter selection (sensitivity ‘VF-Flutter’ 85.3%, specificity ‘Normal-Other’ 95.3%, specificity ‘VT’ 73.9%). Using the proposed methodology, 10 parameters were used, while previous methods gave rise to a more complex detection tree. We can conclude that proposed methodology simplifies the solution and improves discrimination scores.

18.4.4 SVM-based embedded method

Besides the aforementioned filter methods, we also examined the performance of a SVM classifier for detecting VF based on the complete set of temporal, spectral, and time-frequency parameters previously extracted. Once the complete model was characterized in terms of P_e , we applied the FS algorithm defined in Section 2. According to previously presented considerations, wider CI criterion was used to eliminate irrelevant features.

18.4.4.1 SVM Classifier Performance

Given that our purpose was VF detection, the arrhythmia classification problem was simplified by considering a binary output target, this is, aiming to discriminate VF episodes from any other rhythm. The free parameters of the SVM classifier were tuned by using a 5-fold cross-validation strategy. Due to the large amount of available ECG recording samples, the training set was defined as a random subset (20%) of the original data. Those samples being not included in the training set were used as a testset, in order to measure the generalization capabilities of the classifier.

Results are summarized in Table 5. The detection capabilities of VF when using the SVM algorithm can be considered as acceptable, specially taking into account that the complete data base, and not selected segments, had been used, together with unbalance between the examples of each class. Nevertheless, most significant errors were present in a number of VT segments, which were often erroneously classified as VF segments.

Table 5: SVM classifier performance for FV detection.

	Sensitivity	Specificity	Specificity	Specificity	Specificity
	FV-Flutter	NORMAL	OTHERS	TV	global
5-fold	86.4%	97.3%	96.6%	40.0%	90.8%
Test	87.6%	97.5%	97.4%	40.0%	91.4%

Figure 3 shows two application examples of the SVM classifier, which correspond to two different ECG recordings. In the upper panel of each example, the pathology for each segment is represented, whereas the lower panel shows the VF classifier output. Continuous line depicts the target output (VF segments were labeled with $\{-1\}$), and dotted line depicts the soft output of the classifier. Figure 5(a) contains an example in which the algorithm was able to discriminate correctly the VF episodes from all the other pathologies, despite it occasionally failed in VT recordings. Figure 5(b) shows the evolution of the soft output of the classifier towards a VF episode. Note that the transition from sinus rhythm to VF was in this case progressive, and this time interval corresponded to a VT episode which preceded the VF onset.

18.4.4.2 Feature Selection

Table 6 contains the results of the FS algorithm for VF detection. The final set of chosen features are sorted by their relevance order, for temporal (t), frequency (f), and time-frequency (tf) characteristics. The error probability for the complete

Table 6: Results of the FS algorithm for SVM and bootstrap resampling. Chosen variables have been sorted according to the average value of decision statistic $\Delta \mathbf{P}_e^*$. Right column contains the average value and CI of $\Delta thbfP_e^*$ for each variable.

Selected Variables		
CURVE	f	4.95(2.34,7.28)
QTL	f	3.42(0.97,5.91)
MDL8	t	3.30(0.92,5.89)
QTEH	f	2.75(0.28,5.32)
MAXFREC	f	2.65(0.19,5.38)
TSNZ	f	2.60(0.37,4.91)
AREA	tf	2.51(0.09,5.10)
TSNZH	f	2.48(0.19,5.22)
MINFREC	f	2.45(0.09,5.22)
TSNZL	tf	2.42(0.09,5.14)
MINIMFREC	f	2.41(0.19,5.38)
DISPERSION	tf	2.33(0.19,5.15)

feature set was $\mathbf{P}_e^*(\%) = 9.3(7.8, 10.7)$, whereas for the reduced set it was $\mathbf{P}_{e,r}^*(\%) = 11.6(9.3, 14.1)$. Despite the fact of the slight decrease in average error probability with respect to the complete set, the test allows us to consider that the reduction in the feature set has not modified significantly the classifier performance. We should also note that the ratio of spectral and time-frequency parameters of the reduced set increased with respect to the complete set, while the temporal indices were poorly represented in this final selected set of features.

Table 7: Comparative analysis of different methods for FS. Features have been sorted for being able to compare the shared and different results for the three methods.

Method 1	Method 2 (Rosado et al., 2001)	Method 3 (Rosado et al., 2002)
CURVE	CURVE	CURVE
QTEH	QTH	QTEH
TSNZL		TSNZL
QTL		QTL
DISPERSION	DISPERSION	
	TE	TE
	LTMP	LTMP
	TMY	TMY
MINFREC	CT8	LFREC
TSNZ	VDL8	MAXIMFREC
MINIMFREC	VR	PMXFREC
AREA	NAREA	
MDL8	RatioVar	
MAXFREC		
TSNZH		
$\mathbf{P}_{e,r}^*(\%) = 11.6(9.3, 14.1)$	$\mathbf{P}_{e,r}^*(\%) = 9.7(7.8, 11.4)$	$\mathbf{P}_{e,r}^*(\%) = 10.6(8.9, 12.5)$

1.4.5 Filter and embedded methods comparison

Further work has been devoted to analyze the FS process from the set of parameters considered in (Rosado-Muñoz, 2000), which have been used in this study. In (Rosado et al., 2001), up to seven significative parameters were extracted by using classical techniques of discriminant analysis, such as correlation method and the Wilks' Lambda method. From empirical considerations, four more features were included in this set, hence yielding 11 relevant features. Later (Rosado, Camps-Valls, et al., 2002) explored the FS by using data mining schemes, such as principal component analysis a self-organized maps. In this study, a number of 10 features were selected, exhibiting higher performance in sensitivity and speci-

ficity when compared to those ones presented in (Rosado et al., 2001). Table 7 shows a comparison among the results from the SVM and bootstrap FS example (Method 1), and the methods in (Rosado et al., 2001) (Method 2) and in (Rosado, Camps-Valls, et al., 2002) (Method 3). Among the selected features with SVM and bootstrap, only four of them were also included in the set of Method 3, and three of them in the set of Method 2. Note that features QTEH and QTH have been considered as a single variable, given that the correlation between them was higher than 95% (Rosado, Guerrero, Bataller, & Serrano, 2002). Method 2 and Method 3 shared five features. The best method, from a point of view of the error probability in the reduced set, was Method 2. However, the differences with the other methods are minimal, all of them showing good detection capabilities. It can be noted that Method 3 improved Method 2 in terms of sensitivity and specificity, however, the last one had lower error probability. Therefore, the minimum error probability does not guaranty the best performance in terms of sensitivity and specificity. In fact, the VF detection procedure proposed in (Rosado et al., 2001; Rosado, Guerrero, et al., 2002) was made by using a decision tree, in which a first discrimination of the observations relied on temporal features whose relevance was empirically estimated. Despite of the differences in the sets of selected features, their performances were quite similar (Table 7), which shows that, in any case, the original set conveys a great amount of redundant information.

18.5 CONCLUSIONS

This study revised, analyzed and proposed the application of machine learning techniques for the selection of parameters for VF detection. This goal is accomplished by using both filter and wrapper methods. Filter methods allow us to accomplish effective discrimination of VF pathologies by selecting the most relevant features in the database, while wrapper methods optimize the objective criterion (discrimination accuracy) by following a SVM-based approach. We showed that by following the filter approach an improved detection was attained, while the wrapper approach yielded good classification accuracy and extracting information about the relevant features indirectly.

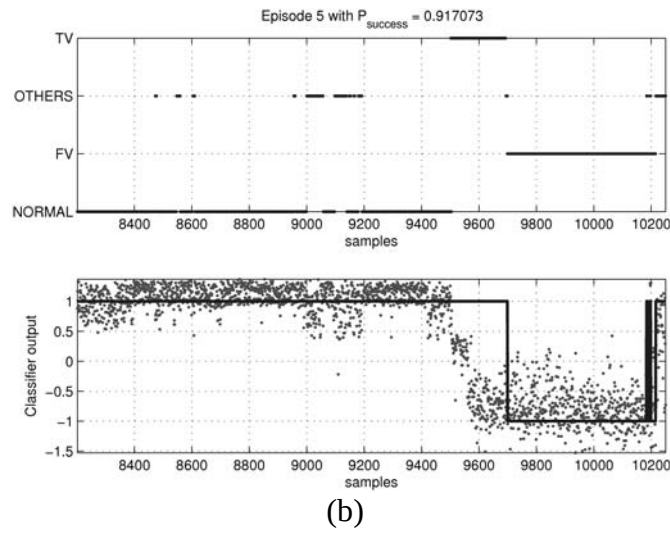
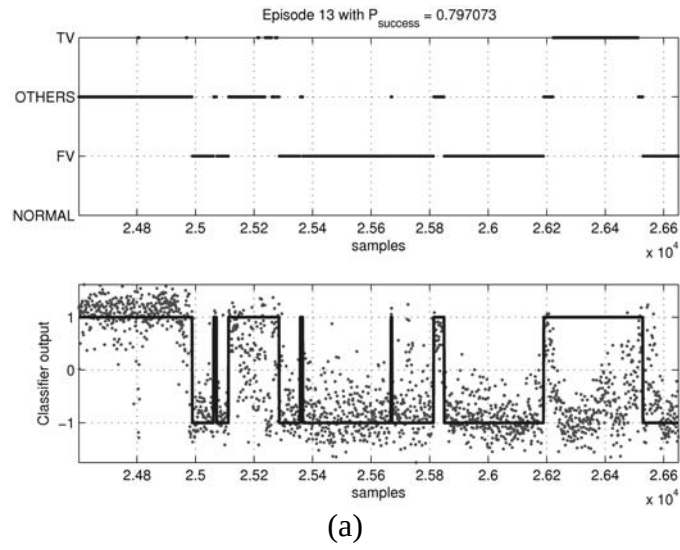


Figure 3: Detection examples of VF episodes with SVM. (a) Episode 13. (b) Episode 5.

References

- AAMI. (1993). Automatic external defibrillators and remote-control defibrillators. (ANSI/AAMI, DF39. Association for the advancement of Medical Instrumentation)
- Afonso, V. X. and Tompkins, W. J. (1995). Detecting ventricular fibrillation. *IEEE Eng. in Medicine and Biology Mag.*, 14, pp. 152-159.
- AHA.(1998). American heart association database. (<http://www.americanheart.org>)
- Akay, M. (1998). Time-frequency and wavelets in biomedical signal processing. IEEE Press.
- Aliot, E., Nitzsché, R. and Ripart, A. (2004). Arrhythmia detection by dual-chamber implantable cardioverter defibrillators. a review of current algorithms. *Europace*, 6, pp. 273-286.
- Alonso-Atienza, F., Camps-Valls, G., Rosado-Muñoz, A., Rojo-Álvarez, J. García-Alberola, A. (2006). Bootstrap feature selection in support vector machines for ventricular fibrillation detection. *Proc. European symposium on artificial neural networks*, pp. 233-238.
- Amann, A., Tratnig, R. and Unterkofler, K. (2007). Detecting ventricular fibrillation by time-delay methods. *IEEE Trans. Biomed. Eng.*, 54, pp. 174-7.
- Aubert, A. E., Denys, B. G., Ector, H. and De Geest, H. (1982). Fibrillation recognition using autocorrelation analysis. *Comp. Cardiol.*, pp. 477-80.
- Barro, S., Ruiz, R., Cabello, D. and Mira, J. (1989). Algorithmic sequential decision-making in the frequency domain for life threatening ventricular arrhythmias and imitative artefacts: a diagnostic system. *J. Biomed. Eng.*, 11, pp. 320-328.
- Beck, C. S., Pritchard, W. H., Giles, W. and Mensah, G. (1947). Ventricular fibrillation of long duration abolished by electric shock. *J. Am. Med. Assoc.*, 135, pp. 985-986.
- Blum, A. and Langley, P. (1998). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97, pp. 245-71.
- Capucci, A., Aschieri, D., Piepoli, M., Rosi, A., Arvedi, M. and Villani, G. (2001). Early defibrillation through first responders triples survivals rates from out of hospital cardiac arrest in an italian community. *Eur. Heart J.*, 22, pp. 242.
- Chen, S., Clarkson, P. M. and Qi, F. (1996). A robust sequential detection algorithm for cardiac arrhythmia classification. *IEEE Trans. Biomed. Eng.*, 43, pp. 1120-1124.
- Clayton, R. H., Campbell, R. W. F. and Murray, A. (1997). Time-frequency analysis of ventricular arrhythmias. *IEE colloquium on time-frequency analysis of biomedical signals* (6/1-6/4).
- Clayton, R. H., Murray, A. and Campbell, R. W. (1994). Recognition of ventricular fibrillation using neural networks. *Med. Biol. Eng. Comput.*, 32, pp. 217-20.
- Clayton, R. H., Murray, A. and Campbell, R. W. (1995). Evidence for electrical organization during ventricular fibrillation in the human heart. *J. Cardiovasc. Electrophysiol.*, 6, pp. 616-624.
- De Ambroggi, L. and Corlan, A. D (2001). New methods of analysis of the ventricular repolarization by using surface potential mapping. *Journal of Electrocardiol.*, 34, 333.
- Doolan, A., Langlois, N. and Semsarian, C. (2004). Causes of sudden cardiac death in young australians. *Med. J. Austral.*, 180(3), 110-112.
- Efron, B. and Tibshirani, R. (1998). *An Introduction to the Bootstrap*. Chapman&Hall.
- Finelli, C. J., Jenkins, J. M. and DiCarlo, L. A. (1993). Detection and identification of cardiac arrhythmias using an adaptive, linear-predictive filter. *Comp. Cardiol.*, pp. 177-180.

- Ge, D. F., Hou, P. and Xiang, X. J. (2007). Study of feature extraction based on autoregressive modeling in ECG automatic diagnosis. *Acta Autom. Sinica*, 33, pp. 462-6.
- Guler, I. and Derya-Ubeyli, E. (2005). A modified mixture of experts network structure for ECG beats classification with diverse features. *Eng. Applic. Art. Intell.*, 18, pp. 845-56.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3, pp. 1157-82.
- Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46, pp. 389-422.
- Henriques, J., Carvalho, P., Gil, P., Marques, A., Rocha, T., and Ribeiro, B. (2007). Ventricular arrhythmias assessment. *Proc 29th IEEE Eng. Med. Biol. Soc.*, pp.3852-55.
- Jalife, J., Gray, A., Morley, G. E. and Davidenko, J. M. (1998). Evidence for electrical organization during ventricular fibrillation in the human heart. *Chaos*, 8, pp. 79-93.
- Jekova, I., Mougeolle, F. and Valance, A. (2004). Defibrillation shock success estimation by a set of six parameters derived from the electrocardiogram. *Physiol. Meas.*, 25, pp. 1179-88.
- Khadra, L., Abdallah, M. and Nashash, H. (1998). Neural wavelet analysis of life threatening ventricular arrhythmias. *IEEE Southeastcon*, pp. 228-229.
- Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Art. Intell.*, 97, pp. 273-324.
- Kohonen, T. (2001). *Self-Organizing Maps (3rd Extended)*. Springer Series in Information Sciences, Vol. 30.
- Kostka, P. and Tkacz, E. (2006). Hybrid feature vector creation for atrial fibrillation detection improvement. *Proc. WC Med. Phys. Biomed. Eng.*
- Kuo, S. and Dillman, R. (1978). Computer detection of ventricular fibrillation. *Comp. Cardiol.*, pp. 2747-50.
- Lee, J., Park, K., Song, M. and Lee, K. (2005). Arrhythmia classification with reduced features by linear discriminant analysis. *Proc 27th AIC Eng. Med. Biol. Soc.*, pp. 1142-4.
- Liu, H. and Motoda, H. (1998). *Feature extraction construction and selection: A data mining perspective*. Kluwer Academic Publishers, Norwell, MA.
- Macfarlane, P. W. and Veitch, T. D. (1989). *Comprehensive electrocardiology. Theory and practice in health and disease*. Pergamon Press UK.
- Michie, D., Spiegelhalter, D. J. and Taylor, C. C. (1994). *Machine learning, neural and statistical classification*. Ellis Horwood.
- Mirowski, M., Mower, M. M. and Reid, P. R. (1980). The automatic implantable defibrillator. *Amer. Heart Journal*, 100, pp. 1089-1092.
- MIT-BIH. (2002). MIT-BIH arrhythmia database. (<http://www.physionet.org/>)
- Osowski, S., Hoai, L. T. and Markiewicz, (2004). Support vector machine-based expert system for reliable heartbeat recognition. *IEEE Trans. Biomed. Eng.*, 51, pp. 582-9.
- Ozbay, Y., Ceylan, R. Karlik, B. (2006). A fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Comp. Biol. Med.*, 36, 376-388.
- Pardey, J. (2007). Detection of ventricular fibrillation by sequential hypothesis testing of binary sequences. *Computers Cardiol.*, pp. 573-6.
- Rojo-Álvarez, J. L., Arenal-Maíz, A. and Artés-Rodríguez, A. (2002). Support vector black-box interpretation in ventricular arrhythmia discrimination. *IEEE Eng. Med. Biol. Mag.*, 21, pp. 27-35.
- Rosado, A., Bataller, M., Vicente, J. M., Guerrero, J., Chorro, J. Francés, J. V. (2000). VF detection method based on a fast real-time algorithm. *Proc. WC Med. Phys. Biomed. Eng.*

- Rosado, A., Camps-Valls, G., Guerrero-Martínez, J., Francés-Villora, J., Muñoz-Marí, J. Serrano-López, A. (2002). Enhancing VF detection using data mining techniques for feature extraction. *Comp. Cardiol.*, pp. 209-212.
- Rosado, A., Guerrero, J., Bataller, M., and Chorro, F. J. (2001). A fast non-invasive ventricular fibrillation detection method using pseudo Wigner-Ville distribution. *Comp. Cardiol.* pp. 237-240.
- Rosado, A., Guerrero, J., Serrano, A. J., Soria, E., Martínez, M., and Camps, G. (1999). Ventricular fibrillation detection method using pseudo Wigner-Ville time-frequency representation. *Proc. Eur. Soc. Eng. Med.*
- Salcedo-Sanz, S., Camps-Valls, G., Pérez-Cruz, F., Sepúlveda-Sanchis, J. and Bousoño-Calzón, C. (2004). Enhancing genetic feature selection through restricted search and Walsh analysis. *IEEE Trans. System, Man Cybern. Part C*, 24, pp. 398-406.
- Thakor, N. V. (1984). From holter monitors to automatic defibrillators: developments in ambulatory arrhythmia monitoring. *IEEE Trans. Biomed. Eng.*, 31, pp. 770-778.
- Ubeyli, E. D. (2008). Usage of eigenvector methods in implementation of automated diagnostic systems for ECG beats. *Dig. Sig. Proc.*, 18, pp.33-48.
- Unterkofler, A. (2005). Reliability of old and new ventricular fibrillation detection algorithms for automated external defibrillators. *Biomed. Eng. Online*, 4(60).
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY.
- Wang, Y., Zhu, Y.-S., Thakor, N. and Xu, Y. (2001). A short-time multifractal approach for arrhythmia detection based on fuzzy neural network. *IEEE Trans. Biomed. Eng.*, 48(9), 989-995.
- Yakaitis, R. W., Ewy, G. A. and Otto, C. W. (1980). Influence of time and therapy on ventricular fibrillation in dogs. *Crit. Care Med.*, 8, pp. 157-163.
- Zhang, X., Zhu, Y. and Gu, H. 1996. Analysis of the nonlinear dynamical characteristics of ventricular fibrillation. In *Proc. IC Sig. Proc.*, pp. 272-275.

Chapter 19

Evolutionary Fuzzy Mixture Model based Decision Support System for an Anaesthetist

Ross Folland, Evor Hines and David Morgan

19.1 INTRODUCTION

Here we present the problem of providing a decision support tool for the anaesthetist, and ergo to model the knowledge of the anaesthetist in evaluating patients during pre-operative assessment. During the pre-operative assessment (*i.e.* prior to a surgical operation) the anaesthetist is responsible for assessing the patient's suitability for anaesthesia. The anaesthetist decides whether the general physical state of the patient is robust enough for the stresses, *etc.* which are a natural part of surgery. Such an assessment may include blood pressure, pulse oximetry, lung function and body weight measurements, *etc.* but most notably includes cardiac and respiratory competence tests. A complete assessment of the patient's cardiac and respiratory systems must be made as it is these systems in which the anaesthetist intervenes during surgery.

Here we investigate the potential application of intelligent systems to support the anaesthetist in pre-operative assessments. In particular we consider two of the principal aspects of these being namely:

- (1) to develop a technique for assessing the competence of the cardiac and respiratory systems of the patient via auscultation; and
- (2) to develop a technique for determining the patient's suitability for surgical anaesthesia via information fusion.

Section 2 introduces the intelligent systems techniques that are employed as part of this work whilst Sections 3 and 4 discuss proposed solutions to the above problem areas. We complete our discussion with an account of the development of a decision support aid (DSA) for the anaesthetist which combines the work undertaken in Sections 3 and 4.

19.2 BACKGROUND

Whilst there is no formal definition, it can be said that an intelligent system is one which is able to learn from and adapt to input stimuli (Bishop, 1995). In particular they can be considered to solve problems by applying knowledge gained from past experiences. Intelligent systems are varied and encompass neural networks (NNs), genetic algorithms (GAs) and linguistic approaches such as fuzzy systems (FSs). Here we briefly consider several NN architectures, namely the multilayer perceptron (MLP), radial basis function network (RBFN) and the constructive probabilistic neural network (CPNN). We conclude our discussion with Genetic Algorithms and the relatively more recent genetic programming (GP) paradigm.

19.2.1 Neural Network

Whilst a more in depth treatment of NNs is presented in previous chapters we briefly recount the underpinning principles behind these techniques. A NN is a collection of inter-connected artificial neurons. Generally, input data is presented to these neurons which as a collective perform a mathematical transform, generating output data as a result of these calculations. The artificial neuron - first proposed by McCulloch and Pitts in 1943 (McCulloch, 1943) - comprises three components: the m inputs $\mathbf{x}_m$ which are weighted with some arbitrary factor w_{ij} ($1 \leq j \leq m$, for neuron i); the processing element in which the weighted inputs are summed and translated through some activation function $a(f)$; and the output response y_i of the function. The output of the neuron is the transform of the weighted sum of inputs such that

$$y_i = a \left(\sum_{j=1}^m w_{ij} x_j - \theta_i \right), \quad (1)$$

where θ_i is an arbitrary offset. As can be seen from Eq.(1) the neuron has a number of adjustable parameters such as the weights w_{ij} and offset θ_i which affect how the neuron responds to input data. The manner in which these neurons are implemented within the NN architecture and the selection of neuronal activation functions gives rise to the different NN types which will be discussed presently.

19.2.2 The Multilayer Perceptron Architecture

Multilayer perceptrons (MLPs) are a particular type of NN favoured in classification problems as they are relatively simple to use, and have been shown to operate well on linear and non-linear problems (Bishop, 1995). The MLP architecture is comprised of layers of neurons inter-connected in such a way that the outputs of one layer are propagated to the subsequent layer. Figure 1 shows the architecture of the MLP and illustrates the relationships between neurons, layers and inter-connections.

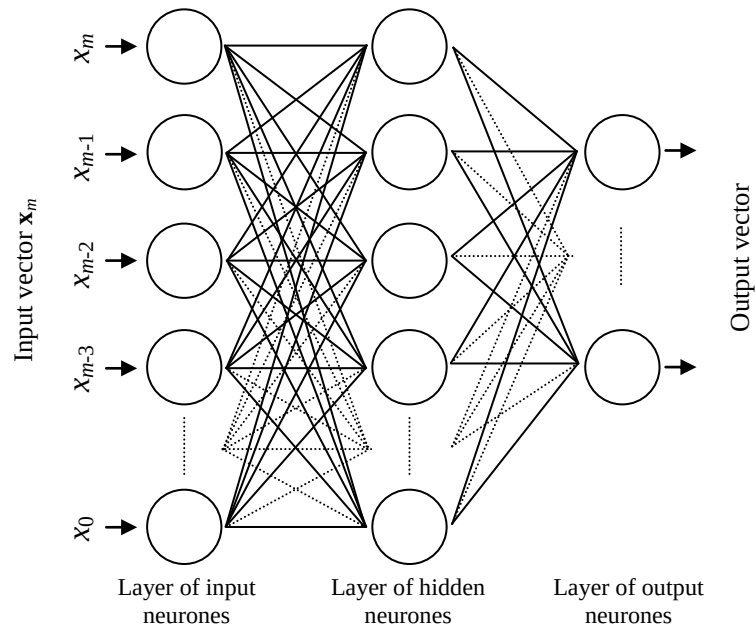


Figure 1: Illustration of a 3-layer MLP

In Figure 1 some data vector $\mathbf{x}_m$ is presented to the input layer. At this layer, the values of $\mathbf{x}_m$ are not weighted but are transformed through the neurons (depicted by circles in Figure 1). The values are propagated to the hidden layer neurons through the inter-connecting synapses (denoted in Figure 1 by the unbroken and dashed lines). Here they are weighted and transformed by the neurons as given by Eq.(1). This process is replicated as the transformed outputs are propagated to the subsequent layer(s). It can be seen from Figure 1 that the output vector of the MLP is some transform of the input vector $\mathbf{x}_m$. It follows that the MLP's response to an input vector $\mathbf{x}_m$ is fundamentally governed by the parameters of the individual neurons. Rumelhart *et al.* elegantly said of the MLP, “*In this case the information coming into the input units [neurons] is recoded into an internal representation and the outputs are generated by the internal representation rather than the original pattern*” (Rumelhart, 1986).

19.2.3 The Radial Basis Function Network architecture

Similar to the MLP, the radial basis function network (RBFN) - suggested by Moody *et al.* (Moody, 1989) - is a NN consisting of artificial neurons. Recalling Figure 1, the RBFN has a similar architecture to the standard MLP, exhibiting fully interconnected layers. Whilst the output layer neurons follow the McCulloch and Pitts model, the hidden layer employs a different type of neuron: the Radial Basis (RB) neuron.

An RB neuron's activation function is an m -dimensional homoscedastic Gaussian kernel φ centred on some prototype vector $\boldsymbol{\mu}$. Considering an m -dimensional input pattern $\mathbf{x}$, the response of the activation function is a measure of the similarity between the input pattern and prototype vector. The activation function $\varphi_n(\mathbf{x})$ is expressed as

$$\varphi_n(\mathbf{x}) = \exp \left\{ -\frac{\|\mathbf{x} - \boldsymbol{\mu}_n\|^2}{2\sigma_n^2} \right\}, \quad \mathbf{x} \in \mathbb{R}^m, \quad n = 1, \dots, N, \quad (2)$$

where the variance of the n^{th} Gaussian function is σ_n^2 . The Gaussian functions are typically normalised such that

$$g_n(\mathbf{x}) = \frac{\varphi_n(\mathbf{x})}{\sum_{n=1}^N \varphi_n(\mathbf{x})}. \quad (3)$$

It is evident from Eq.(2) that the response of a Gaussian activation function is partially governed by the Euclidean distance between the input pattern and the prototype vector with the smoothness of the function being defined by σ_n .

The output layer of the RBFN consists of neurons with linear activation functions. The output neurons sum the weighted outputs of the hidden-layer RB neurons and transform the result through the activation function so the output o_i becomes

$$o_i = a_i \left(\sum_{k=1}^N w_{ik} g_k(\mathbf{x}) - \theta_i \right), \quad i = 1, \dots, L, \quad (4)$$

where o_i is the i^{th} final-layer neuron output, L is the number of neurons in the output layer, N is the number of neurons in the hidden layer and θ_i is the i^{th} neuron offset.

19.2.4 The Constructive Probabilistic Neural Network architecture

We start our discussion with the probabilistic neural network (PNN) (Specht, 1990; Specht, 1992). Recalling Figure 1, the PNN is similar to the 3-layer MLP in that it is comprised of an input layer, hidden layer and output layer. The PNN possesses a fully inter-connected input and hidden layer, ensuring that all pattern vectors presented to the network are available to all hidden neurons. The hidden neurons utilise an exponential activation function in their neuronal model and there exist as many neurons as there are patterns presented to the network – similar to an un-optimised RBFN. The second synaptic layer of inter-connections exists to aggregate the responses of the hidden neurons into class neurons and is thus only partially connected. The existence of these partial inter-connects indicate that the true class assignments of the training data are known during the training phase, *i.e.* that the PNN employs a supervised learning mechanism. The final layer represents the probability of class assignment for each class (Berthold, 1998). NNs operate by estimating the statistical distribution of the training data and use these estimations to define posterior probability densities $P(x|C_k)$ (*i.e.* the probability of class k given the datum x) for classification of unknown data. One of the most straightforward approaches to this statistical density estimation is to represent the probability density $P(x)$ in a functional form. The PNN represents these densities as multivariate normal distributions which can be expressed as

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{m/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}. \quad (5)$$

The constructive probabilistic neural network (CPNN) is essentially a PNN that is grown by the sequential addition of neurons in the hidden and output layers. The neurons are added in response to patterns presented in the training dataset. Prior to adding a neuron, an assessment is firstly made as to whether existing neurons can perform the same function. If they can, then they are adjusted to encompass the new training pattern, otherwise a new neuron is added. This method of constructing the CPNN often results in highly compact and efficient architectures requiring low computational overheads. One method for constructing a PNN is the dynamic decay adjustment (DDA) algorithm.

The DDA algorithm (Berthold, 1998) is a constructive algorithm in that it iteratively adds neurons to the network and adjusts the parameters of the neuronal activation functions in Eq.(5). Parameters are adjusted by the algorithm to prevent conflict between neurons *i.e.* when two or more neurons linked to different classes respond to an input pattern resulting in an ambiguous classification. Adjustment to the parameters include modification of the amplitude of the function and the covariance matrix Σ . Berthold et al. define two terms

- (1) θ^+ the *minimum correct-class probability*. Given an input pattern $\mathbf{x}$ linked to class k presented to the network, at least one neuron linked to the same class must respond with an activation greater than θ^+ ; and
- (2) θ^- the *maximum incorrect-class probability*. Given an input pattern $\mathbf{x}$ linked to class k presented to the network, any neurons linked to any other class are permitted to respond provided that their activations are no greater than θ^- .

Presented now is an annotated formal description of the algorithm as given by (Berthold, 1998) followed by a qualitative description. For clarity $\phi_i^k(\mathbf{x})$ is the activation function for neuron i linked to class k as given in Eq.(5).

- (1) Present a training pattern-class pair $(\mathbf{x}, k)$ to the network.
- (2) If $\exists \phi_i^k(\mathbf{x}) : \phi_i^k(\mathbf{x}) \geq \theta^+$ then increase the amplitude A_i^k of function $\phi_i^k(\mathbf{x})$ by 1 and proceed to step 4.
- (3) Introduce a new neuron with activation function $\phi_{i+1}^k(\mathbf{x})$, amplitude $A_{i+1}^k = 1$

and mean vector $\mu_{i+1}^k = \mathbf{x}$. Calculate the standard deviation using

$$\sigma_{i+1}^k = \min_{l \neq k, 1 \leq j \leq q_l} \left\{ \sqrt{-\frac{\|\mu_j^l - \mu_{i+1}^k\|^2}{\ln \theta^+}} \right\}, \quad (6)$$

where $l \neq k, 1 \leq j \leq q_l$ is equivalent to all neurons connected to all classes other than class k (correct class), and q_l is the number of neurons in class l .

(4) Adjust the standard deviation of all conflicting neurons using

$$\forall l \neq k, 1 \leq j \leq q_l : \sigma_j^l = \min \left\{ \sigma_j^l, \sqrt{-\frac{\|\mathbf{x} - \mu_j^l\|^2}{\ln \theta^-}} \right\}, \quad (7)$$

and then proceed to Step 1.

In qualitative terms, when a pattern-class pair $(\mathbf{x}, k)$ is presented to the network during training, if any neuron i linked to class k responds with an activation $\geq \theta^+$ then the neuron's activation function amplitude is increased by 1. All neurons linked to classes $l \neq k$ have their standard deviations σ reduced using Eq.(7). The reduction is such that their activation responses to pattern $\mathbf{x}$ will be less than θ^- . If no neuron of class k responds to the pattern $\mathbf{x}$ then a new neuron is added to the architecture, linked to class k and centred around $\mathbf{x}$ i.e. $\mu_{i+1}^k = \mathbf{x}$. The new neuron is added to the architecture to account for future occurrences of the pattern.

19.2.5 Genetic Algorithms & Genetic Programming

Here we discuss two of the principal evolutionary techniques, these being namely the genetic algorithm (GA) and genetic programming (GP) paradigms.

A GA can be defined as a search procedure based on the mechanics of natural selection and genetics (Goldberg, 2002). To better discuss GAs and their application we consider the following points by (Koza, 1992)

- (1) There exists an environment possessing some entity which has the ability to reproduce;
- (2) The environment contains a population of such self-reproducing entities;
- (3) There is some variety between the self-reproducing entities; and

- (4) This variety affects the entity's ability to survive in the environment.

Given a natural or artificial system adhering to these four points it becomes possible to observe an evolutionary process by which successive generations of the self-reproducing population change in response to their environment and peers (usually to the betterment of the population). Theorised by Darwin (Darwin, 1859) the process of evolution remained until when Holland (Holland, 1962; Holland, 1975) demonstrated how it could be applied to artificial systems. One such modality is the GA which provides a framework for codifying the problem domain and evolving solutions via a mixture of breeding functions and random mutation operations. Whilst a more comprehensive account is presented in previous chapters, for a detailed exposition of the theory of GAs the reader is referred to (Goldberg, 2002; Holland, 1975).

The theory of GAs as developed by Holland and subsequent researchers was based primarily on the underlying principle that the GA operates on chromosome parameter sets of fixed size. Koza argues that this principle limits the effectiveness of GAs on numerous fronts and - given that one generally lacks sufficient information concerning the optimisation problem - it is ill advised to impose a fixed-size parameter set on the GA as this can have the undesired effect of constraining the optimisation process (Koza, 1992). In response to this perceived lack of dynamic variability in GAs the GP paradigm has been lent considerable credence.

In the GP paradigm a genetic program consists of functions and terminals specific to the problem under consideration. Functions can be arithmetic operations, mathematical functions, logical operations, *etc.*; whilst terminals can be variables, parameters, internal system states, *etc.* The GP approach, as will be shown later, adopts the same evolutionary principle as the GA technique and for a more comprehensive discussion on the mechanics of the GP paradigm the reader is referred to (Koza, 1992).

19.3 CLASSIFYING CARDIAC & RESPIRATORY ABNORMALITY USING INTELLIGENT SYSTEMS

Here we provide an account of how a NN ensemble was developed to identify cardio-respiratory abnormalities based on auscultated sounds. A précis of the data pre-processing methods and subsequent NN analysis is presented alongside

a technique for aggregating the responses of each NN in the ensemble to form a more robust classification scheme.

19.3.1 Data Processing Considerations

The audible sounds heard during auscultation are the consequence of either normal or abnormal cardio-respiratory function with abnormal sounds being frequently the result of some pathology or physiological defect. The different cardiac and respiratory sounds that the NNs were trained to recognise are presented in Table 1 and Table 2 respectively. The regions of auscultation are also indicated in both cases.

Table 1: Cardiac abnormalities the NNs were trained to recognize.

Cardiac abnormality	Region of auscultation
Normal heart sound	Mitral region
Hypertrophic obstructive cardiomyopathy	Aortic region
Non-rheumatic mitral regurgitation	Mitral region
Mitral valve prolapse	Mitral region
Bicuspid aortic valve	Aortic region
Aortic stenosis	Aortic region
Severe calciphic aortic stenosis	Aortic region
Aortic regurgitation	Aortic region
Mixed aortic valve disease	Aortic region
Mitral stenosis	Mitral region
Mixed rheumatic mitral valve disease	Mitral region
Atrial septal defect	Aortic region
Ventricular septal defect	Ventricular region
Dilated cardiomyopathy	Ventricular region
Aortic incompetence	Aortic region
Papillary muscle dysfunction	Ventricular region
Left ventricular impairment	Ventricular region

Two separate NN ensembles were trained: one which classified cardiac auscultated sounds into the classes given in Table 1; and one which classified respiratory auscultated sounds into the classes given in Table 2. Each ensemble consisted of a MLP, RBFN and CPNN. Prior to NN training the acquired samples were pre-processed with the aim of retaining information whilst reducing the amount of underlying data.

Real-world data is frequently unsuitable for direct analysis because the information it contains is inefficiently represented (Bishop, 1995). For example, given a dataset pertaining to some process we wish to model, each datum within the set is N dimensional (*i.e.* N variables that describe the data).

Table 2: Abnormalities the NNs were trained to recognise.

Respiratory abnormality	Region of auscultation
Normal tracheal-bronchial sounds	Tracheal region
Normal broncho-vesicular sounds	Bronchial region
Normal vesicular sounds	Bronchial region
Diminished sounds	Bronchial region
Tubular breath sounds	Bronchial region
Fine inspiratory crackles	Bronchial region
Medium, coarse inspiratory and expiratory crackles	Bronchial region
Fine, late inspiratory crackles	Bronchial region
Mild expiratory wheeze	Bronchial region
Medium inspiratory crackles with moderate and severe expiratory wheeze	Bronchial region
Pleural friction rub	Bronchial region
Inspiratory and expiratory stridor	Tracheal region
Bone crepitus	Bronchial region
Subcutaneous emphysema	Bronchial region

It is often the case that a subset of these dimensions is irrelevant as it contains no apparently useful information (*e.g.* the variables may be correlated with each other). Dimensionality reduction is the process by which one identifies and removes redundant dimensions from the dataset. By removing apparently unimportant dimensions we retain the informative features of the dataset which may be used in a subsequent analysis. The importance of feature extraction is exemplified by (Hartman, 1990).

The auscultated sounds were sampled at a rate of 44kHz for approximately 30 seconds. The following processes were performed separately for the cardiac and respiratory sounds

- (1) Each sound was sampled and vectorized to approximately 1.32×10^6 elements (*i.e.* ~ 30 seconds $\times$ 44 kHz);
- (2) A matrix was created from the samples giving $\mathbf{X} = \{x_{ij}\}$ where x_{ij} was the j^{th} element of sample i ; and

- (3) The frequency spectra of the acquired sounds were computed using the Fast Fourier Transform (FFT) which afforded a frequency domain representation of the sounds. After the addition of noisy replicates the final size of the cardiac dataset was 1700 samples, and the respiratory dataset 1400 samples.

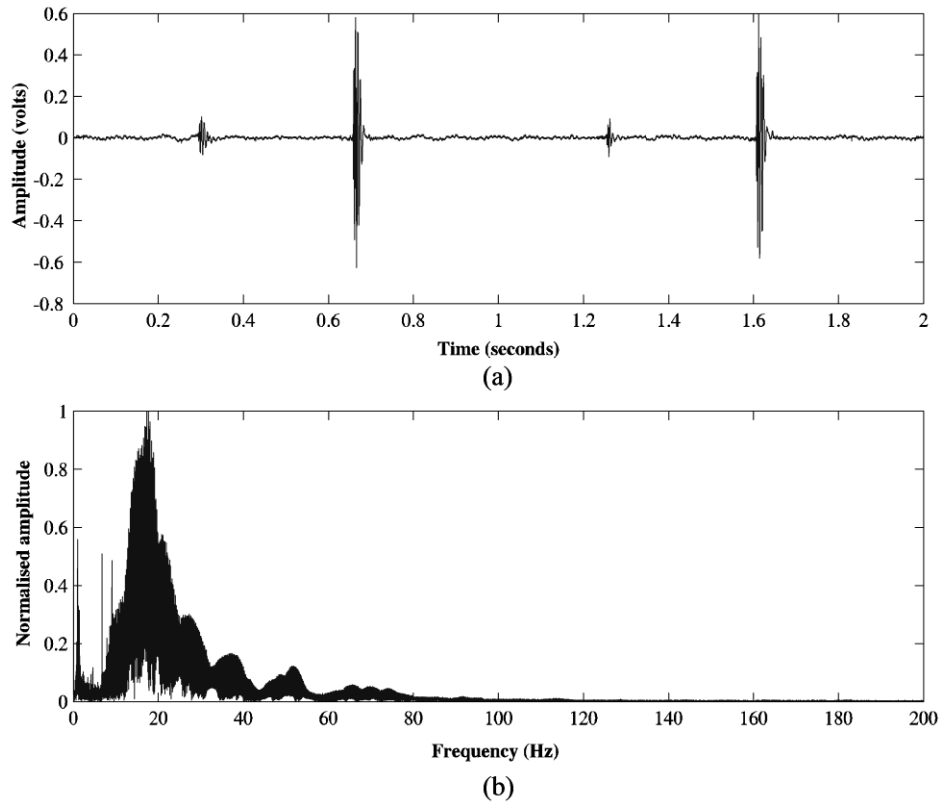


Figure 2: Example of (a) 2 normal cardiac sound cycles; and (b) the associated frequency spectrum.

Direct analysis of these spectra by NNs was not feasible due to the FFTs being highly dimensional so a method of encoding the FFTs whilst reducing the dimensionality of the dataset was required. The method employed was the Levinson-Durbin autoregression algorithm for spectral estimation (Luo, 1997). The method is a computationally light-weight approach to parametric signal estimation, constructing a P -order data model that describes the frequency

spectra. Given a data model $\hat{s}_x(f)$ which is a P -order estimate of signal s , information from s can be encoded into the estimation model parameters. A model order of 15 parameters was selected based on an analysis of residual error between the original and estimated signals. The Levinson-Durbin algorithm was applied to all of the frequency spectra for both the cardiac and respiratory sounds. This resulted in two matrices, one containing cardiac FFT estimation parameters and the other containing respiratory FFT estimation parameters, both containing 15 columns which were subsequently linearly normalised such that $\{a_k \mid k = 1, 2, \dots, 15\} \rightarrow [0, 1]$.

19.3.2 System Implementation

The MLP was applied to discriminate between the different classes of cardiac and respiratory sounds. The 3-layer (*i.e.* input, hidden and output layer) MLP was employed with the topology being estimated using the geometric mean approach suggested by Widrow *et al.* (Widrow, 1990).

The MLP was trained with 1000 cardiac samples and tested with the remaining 700, from which a results matrix was formed (of dimensions 700×17). Given that each MLP response y took some value $y \rightarrow [1, 0]$ a threshold θ was used to convert these results to a binary 1-in-N equivalent. The optimal θ was found to be 0.30, yielding an MLP classification accuracy of 99.20% with sensitivity of 88.24% and specificity of 99.88%.

The process used to train an MLP in identifying different classes of respiratory abnormalities was similar to that used in the previous cardiac sounds experiment. We recall that the respiratory dataset contained 1400 samples each of 15 variables. The resulting MLP topology was a 15-15-14 architecture which was trained for 2500 epochs on a dataset of size 1000×15 and subsequently tested with a dataset of size 400×15 . The optimal threshold θ was found to be 0.10 resulting in a classification accuracy of 67.11% (77.00% sensitivity, 66.35% specificity).

We now consider the training of a RBFN to classify the cardiac and respiratory datasets in an attempt to compare with – and improve upon – the MLP-based classifier. The RBFN's architectural considerations were such that an appropriate number of hidden radial basis neurons should be selected prior to training. K-means clustering analysis was performed on the cardiac dataset in order to determine the optimal number of clusters that could partition the data

and hence the optimal number of neurons in the RBFN hidden layer. 27 clusters adequately covered the input feature space and consequently a 15-27-17 RBFN was created and trained on the cardiac 1000-sample training dataset with the resultant SSE being 92.30. Given the 700-sample novel test dataset, the RBFN attained a classification accuracy of 99.88% (sensitivity 99.06%, specificity 99.93%). The optimal threshold cut-off θ yielding these results was found to be at 0.45.

Training and subsequent testing of the RBFN to discriminate between the different respiratory sounds followed a similar procedure. The K-means clustering technique identified that 24 clusters provided the optimal coverage of feature space and hence the optimal number of RBFN hidden layer neurons. A 15-24-14 RBFN was created and trained on 1000 samples of the available respiratory dataset. The RBFN was subsequently tested on the remaining 400 samples and attained a classification accuracy of 95.79% (100% sensitivity, 95.46% specificity) with the optimal θ being 0.02. We now consider the application of the CPNN to the discrimination between the cardio-respiratory abnormalities.

Prior to creating, training and testing a CPNN with the cardiac data the threshold levels θ^+ and θ^- were defined. These two values fundamentally govern the resulting sensitivity and specificity of the CPNN (Berthold, 1998). Berthold *et al.* observed that the respective values of 0.4 and 0.2 provided excellent results in practical situations.

The CPNN was trained using 1000 of the cardiac data samples, with a fixed input layer size of 15 neurons, and fixed output layer size of 17 neurons. The number of training epochs was set to 5 and the algorithm grew the architecture entirely within the first training pass. Adjustments to the hidden neurons' density functions were made during each training epoch. After training the CPNN hidden layer contained 286 neurons and when tested with the remaining 400 test patterns, attained a classification accuracy of 89.17% (89.17% sensitivity, 99.36% specificity). A CPNN was also created and trained to discriminate between the different abnormalities presented in the respiratory dataset. Similar to the previous CPNN implementation, the classifier was trained for 5 epochs on 1000 samples from the respiratory dataset. When tested with the remaining 400 samples the CPNN attained a classification accuracy of 97.80% (97.80% sensitivity, 89.60% specificity).

19.3.3 Evidence Combination using Dempster-Shafer Theory

The Dempster-Shafer Theory (DST) of evidence combination offers a framework for reasoning between imprecise, ambiguous and conflicting probabilistic assertions (*i.e.* classifications). We start with a brief overview of DST whilst a complete account can be found in (Shafer, 1976). DST extends traditional Bayesian classification to incorporate belief and plausibility terms. In order to define these concepts and examine how they can assist in combined classification we consider the following. Let each possible classification $\mathbf{x} \in C_k$ be one of a number of mutually exclusive propositions A_i where $i = 1, \dots, k$. The classification from a NN in an ensemble can be considered to be a basic belief assignment m . The value $m(A)$ can be considered to be the mass of belief that is committed to $\mathbf{x} \in A_k$ (and to none of its subsets) (Denceux, 1997). Consequently for all subsets (*i.e.* classifications) $B \subseteq A$ the belief measure $Bel(A)$ can be expressed as

$$Bel(A) = \sum_{B \subseteq A} m(B). \quad (8)$$

The quantity $Bel(A)$ is the extent to which one believes in A either directly attributable to $m(A)$ or indirectly attributable to $m(B)$ for all $B \subseteq A$. In extension to this, plausibility is defined as the body of evidence committed to $\bar{A}$ or its subsets *i.e.* the extent of doubt of $\mathbf{x} \in A_k$. Plausibility is therefore expressed as

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B) = 1 - Bel(\bar{A}). \quad (9)$$

Belief and plausibility measures afford the user with a framework for assessing the strength of classifications from ANNs. An important aspect of DST is its ability to combine classifications (*i.e.* evidence) provided by multiple sources (*e.g.* ANNs in an ensemble). Given two basic belief assignments m_1 and m_2 from two independent sources, Dempster's rule of evidence combination follows the form

$$m_E(A) = \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)}, \quad \forall A \neq \emptyset, \quad (10)$$

where $m_1(B) > 0$ and $m_2(C) > 0$ for some non-disjoint subsets B and C of universe of discourse Θ and $m_E(A)$ is the orthogonal sum of m_1 and m_2 . As can

be seen from Eq.(10), Dempster's rule of evidence combination permits the aggregation of independent NN classifications at the output-neuron level. It facilitates the creation of a combined NN classifier that accounts for conflicting and ambiguous classifications.

Using Dempster's rule of evidence combination presented in Eq.(10) the combined classifications for both datasets were established. Table 3 and Table 4 present individual NN classifications and their combined results for both cardiac and respiratory datasets respectively.

Table 3: Comparison of accuracy, sensitivity and specificity of the MLP, RBFN, CPNN and DST approaches to the cardiac sounds classification.

Classifier	Accuracy	Sensitivity	Specificity
MLP	99.20%	88.24%	99.88%
RBFN	99.88%	99.06%	99.93%
CPNN	89.17%	89.17%	99.36%
Combined via DST	99.77%	98.14%	99.88%

Table 4: Comparison of accuracy, sensitivity and specificity of the MLP, RBFN, CPNN, and DST approaches to the respiratory sounds classification.

Classifier	Accuracy	Sensitivity	Specificity
MLP	67.11%	77.00%	66.35%
RBFN	95.79%	100.0%	95.46%
CPNN	97.80%	97.80%	89.60%
Combined via DST	100.0%	100.0%	100.0%

Tables 3 and 4 demonstrate that given a plurality of NN classifiers, an aggregated approach can improve classification accuracy. Whilst the accuracy of the combined classification mechanism for analysing cardiac auscultated sounds appeared lower than the RBFN, the difference was negligible (0.11%). Table 4 shows that the combined classification scheme attained 100% classification of the test dataset, with 100% sensitivity and specificity.

In Figure 4 we observe the accuracies of the MLP, RBFN and DST techniques. It is interesting to note that for low (< 0.3) values of θ the DST system behaves like the RBFN classifier in terms of its accuracy, whereas above this θ value it tends towards the relatively poorer abilities of the MLP.

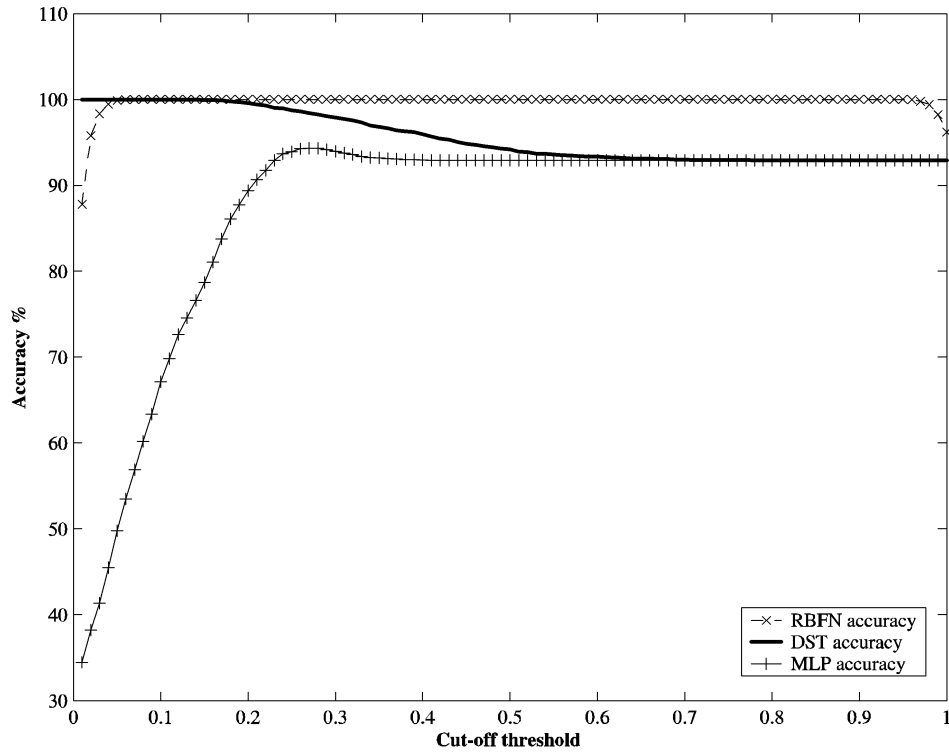


Figure 4: Classification accuracies of the RBFN, DST and MLP classifiers on the respiratory dataset.

At $\theta > 0.3$ the MLP sensitivity approached 2% so a trade-off between classification accuracy and sensitivity was found at $\theta = 0.10$. Given the results in Tables 3 and 4 it can be observed that the combined classification mechanism does not inherit the relatively poor sensitivities attained by the weakest classifier in the ensemble.

19.4 CLINICAL DECISION SUPPORT WITH EVOLUTIONARY CLASSIFIERS

In forming the DSA this section draws upon the work presented in the previous section. Here we apply the GP paradigm to the task of constructing an Evolutionary Fuzzy Mixture Model (EFMM) that can successfully model the anaesthetist's decision-making strategy. The purpose of the EFMM is to fuse the

classifications performed by the DST-based ensembles of Section. 3 with other pertinent patient data in forming a complimentary decision strategy.

19.4.1 Data Processing Considerations

The DSA is furnished with input from the cardiac and respiratory classifiers. Recalling from Section 3 that the outputs of the cardio-respiratory ANNs were 1-in- N encoded and normalised such that the DST method could be applied to aggregate responses in forming more accurate classifications. The outputs from the DST-based system were similarly normalised thereby representing the classifications as probabilistic assertions as to the cardio-respiratory abnormality. Each DSA input from both the cardiac and respiratory classifiers was bound between 0 and 1 representing classification probabilities. A further DSA input was the patient's SpO_2 measured via pulse oximetry. The final DSA inputs were a plurality of biometric values briefly comprising blood pressure, heart rate, lung function and clinical history indicators. The variables considered as part of this work are presented in the summary Table 5. The encoding of the patient's clinical history and medication as presented in Table 5 merits particular attention. Relevant patient information was grouped into five categories: haemodynamic, cardiac (incorporating circulatory), respiratory, biohazard and medication. Table 5 further expands each category into the relevant conditions which are noted in the patient's clinical history. Each item (*e.g.* cardiovascular disease history) was binary-encoded with 1 and 0 representing the presence or absence of the item respectively. These values were bit-masked such that the category value was set to 1 if any of its 'child' values were 1.

Data was collected from 194 patients and was classified by a consultant anaesthetist into three categories: 'suitable for general anaesthesia', 'possibly suitable for general anaesthesia (refer patient)', and 'unsuitable for general anaesthesia'.

Table 5: Values provided to the DSA during normal operation.

Metric	Unit	Minimum value	Maximum value
Cardiac sound classification	N/A	0	1
Respiratory sound classification	N/A	0	1
Blood Pressure (systolic)	mmHg	40	220
Blood Pressure (diastolic)	mmHg	30	150
Heart Rate	Bpm	25	130

Peak Expiratory Flow Rate	L/min	50	500
Forced Expiratory Volume (1s)	L	0.1	10
SpO ₂	%	1	100
Patient age	Years	18	110
Patient gender	N/A	0 (female)	1 (male)
Warfarin, Inhalers, HRT	N/A	Binary encoded:	1: true / positive presence or history 0: false / negative presence or history
Steroids, Aspirin, Diuretics	N/A		
Female Contraceptive Pill	N/A		
Hepatitis B, Hepatitis C	N/A	Binary encoded:	1: true / positive presence or history 0: false / negative presence or history
MRSA, Active TB, HIV	N/A		
Afro-Caribbean origin	N/A	Binary encoded:	1: true / positive presence or history 0: false / negative presence or history
Excessive alcohol intake	N/A		
Infection	N/A		
Blood disorders / anaemia	N/A		
Diabetic	N/A		
Kidney / urinary problems	N/A		
Bleeding disorders	N/A		
Thyroid disease	N/A		
Malignancy	N/A		
Open wound	N/A		
Smoker (> 40 per day)	N/A	Binary encoded:	1: true / positive presence or history 0: false / negative presence or history
Cardiovascular disease history	N/A		
Varicose veins	N/A		
Diabetic	N/A		
Smoker (> 40 per day)	N/A	Binary encoded:	1: true / positive presence or history 0: false / negative presence or history
Pneumonia / chronic bronchitis	N/A		
Asthma (well controlled)	N/A		
Asthma (hospitalised)	N/A		
Shortness of breath at rest	N/A		
Respiratory disease history	N/A		
Productive cough	N/A		

19.4.2 The Evolutionary Fuzzy Mixture Model

Whilst NNs can generalise solutions to domain classification problems, they cannot easily produce a linguistic interpretation of their decisions. The response of a conventional NN is the weighted sums of successive linear and non-linear translations usually devoid of any inductive reasoning that could explain its

decision-making. Because of this NNs are infrequently employed in medical applications which require classifications based on this inductive reasoning. Fuzzy classifiers on the other hand possess the ability to linguistically demonstrate the chain of reasoning which results in their classifications. Rule-based expert systems can also be used for this type of decision-making, however fuzzy systems are frequently used due to their ability to handle imprecision in their reasoning scheme (Jang, 1997). For this compelling reason the DSA employed a form of fuzzy classification in its decision-making strategy.

For the purpose of this section we consider a fuzzy rule to be a linguistic statement of variable x 's membership of class C to degree $\mu_C(x)$ (the fuzzy membership function). In many real-world scenarios data does not generally follow set distributions (*e.g.* Gaussian, *etc.*) and when analysed with models employing fuzzy membership functions the mechanism is committed to increasing rule-base complexity to account for these distributions, sometimes to the detriment of classification accuracy and sensitivity. The EFMM attempts to ameliorate this by providing a framework for the mixing of fuzzy membership functions. To illustrate this principle, consider the set of membership functions $\psi = \{\mu_{tr}, \mu_g, \mu_b, \mu_t\}$ where these functions are trapezoidal, Gaussian, bell and triangular respectively. From these functions one can identify a set of common parameters: the function's centre point c and spread value σ . The fuzzy membership functions are mixed by simple addition of the weighted functions of ψ such that

$$\mu(x) = w_{tr}\mu_{tr}(x) + w_g\mu_g(x) + w_b\mu_b(x) + w_t\mu_t(x), \quad (11)$$

where w_{tr} , w_g , w_b and w_t are weight terms which sum to unity. The weights themselves are defined by a series of triangular membership functions which are governed by a mixture coefficient m ($0 \leq m \leq 100$). The membership function $\mu(x)$ in Eq.(11) is a linear mixture of at most two functions in ψ . Between the range $0 \leq m < 33$ the membership function is a mixture of triangular and Gaussian components (at $m=33$ the function is purely Gaussian). Similarly between the range $33 < m < 66$ the membership is a mixture of Gaussian and bell-shaped components (at $m=66$ the function is purely bell-shaped), and so forth. The consequent terms employed by the EFMM were namely 'suitable for general anaesthesia', 'possibly suitable for general anaesthesia (refer patient)', and 'unsuitable for general anaesthesia'.

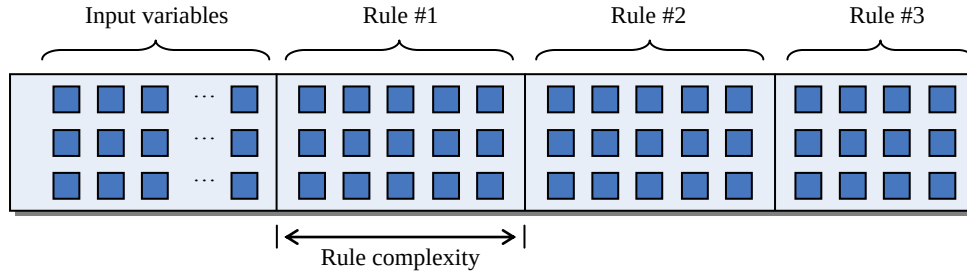


Figure 5: Structure of the EFMM chromosome.

With reference to Figure 5 an EFMM can be coded into a chromosome of variable length. The first segment of the chromosome relates to the input variable mapping using fuzzy membership functions, *i.e.* for each input variable there was created a fuzzy membership function centred around c with spread σ and being a mixture of m . Each fuzzy membership function linguistically expressed whether its associated variable was medium and further membership functions were implied for the variable being low and high. Example membership functions are illustrated in Figure 5.

The remainder of the EFMM chromosome, as illustrated in Figure 5, is comprised of 3 fuzzy rules, each corresponding to an output classification. Each rule is of varying complexity and each term in the rule comprises an operator (either AND $\tilde{\times}$, OR $\tilde{+}$, AND NOT $\neg\tilde{\times}$ and OR NOT $\neg\tilde{+}$), a fuzzy variable (given in the first segment of the chromosome), and a linguistic expression (either ‘is low’, ‘is medium’ or ‘is high’).

As can be observed the above model provided no mechanism for adapting its fuzzy rulebase to accord with the anaesthetist’s decision making strategy. We therefore employed a genetic technique to evolve the model to adapt to – and hence learn – the anaesthetist’s strategy. Evolution of the EFMM is derived from the fact that new models can be bred from two existing models. Breeding is a two-stage process

- (1) Breed the input variable mappings: for each variable membership function, either mutate the function, create a new function, or breed the function from its parents *i.e.* take m , c , and σ from either parent.
- (2) Breed the terms in each rule: for each rule term either mutate the term, create a new term, delete a term, or breed the term from its parents *i.e.* take the

operator, variable and linguistic term from either parent.

Using the method of evolution above a population of characteristically unique EFMMs can be generated and change over successive generations. Two distinct methods of evolution are presented in subsequent sections alongside an account of their efficacy.

19.4.3 Model Learning with Genetic Algorithms

Of the available dataset discussed in Sec. 4.1, 150 patient samples were made available to the EFMM for learning whilst 44 samples were retained for testing purposes. The system was trained for 100 generations on a dataset which had been previously classified by a consultant anaesthetist. The model was evolved using a GA with a population size of 500 chromosomes and the maximum allowable rule complexity of 44 antecedents per rule *i.e.* permitting one antecedent term per variable.

The model converged to a solution after 5 generations, with the fittest chromosome MSE being 0.26. The worst performing chromosome MSE appeared to stabilise at approximately 1.75. Given the training dataset of 150 samples, the GA EFMM correctly classified 87.56% with a sensitivity of 81.33% and specificity of 90.67%. The EFMM was tested with the 44 remaining samples and correctly classified 90.91% with a sensitivity of 86.36% and specificity 93.18%.

19.4.4 Model Learning with Genetic Programming

The GP-based DSA incorporating the EFMM kernel was executed with an initial population size of 50 programs. This was performed as a preliminary test to assess the performance of the system. The poorest performing 20% of the population was arbitrarily terminated between generations, thereby freeing space for new programs.

Furthermore a probabilistic measure was applied to program crossover such that the programs which had the highest fitness is given highest probability of breeding, in accordance with the genetic paradigm formalised by Holland. The probability of mutation and reproduction were both 10%. Figure 6 shows the error curves for each population of 20 generations. It illustrates that as the

population changed from generation to generation, the average program MSE became smaller. The experiment was conducted again with a larger population size (1000 programs) and over 100 generations. The early stopping criterion was that at least one program had a $MSE \leq 0.25$. The fittest program classified the training dataset (*i.e.* successfully predicted the surgical anaesthesia suitability group) with an accuracy of 89.86%, sensitivity of 88.37% and specificity of 92.11%.

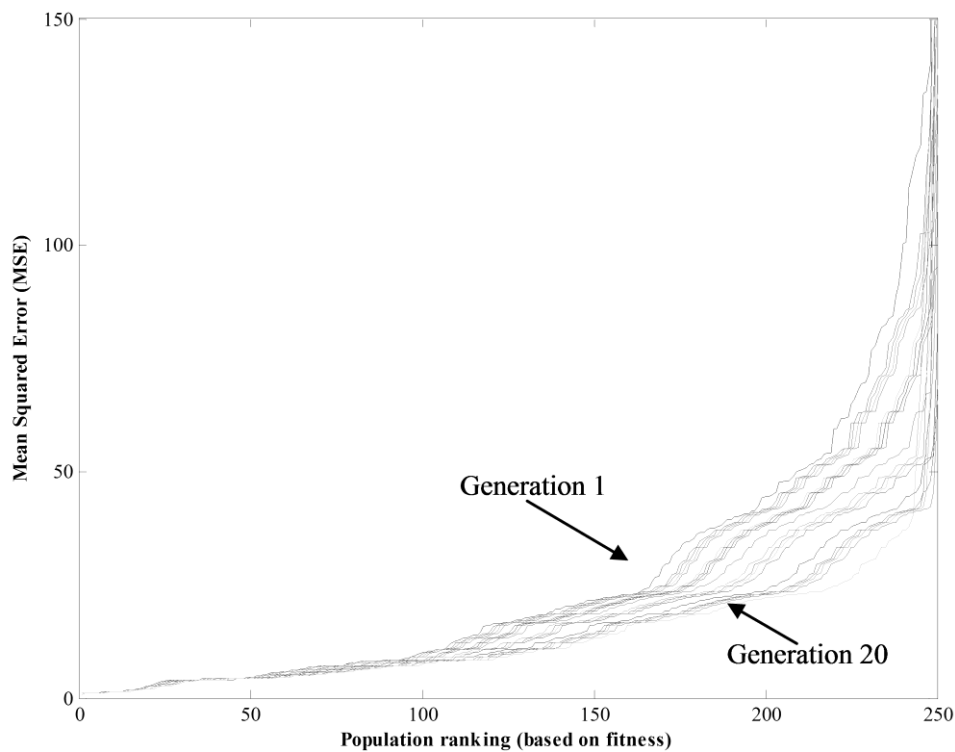


Figure 6: Error curves for the best and worst performing EFMM programs over 20 generations.

When presented with the test dataset, the fittest program correctly classified 91.79% (88.21% sensitivity, 95.18% specificity).

When comparing the performances of the GA and GP approaches, two main criteria must be considered: system accuracy and architectural size. The former relates to the ability of the system to correctly predict the surgical anaesthesia suitability group in accordance with the anaesthetist's decision. The second pertains to the efficiency of the decision *i.e.* how many antecedent rule-base

terms the system required in order to make the decision. Presented below is the rule-base generated by the GA EFMM after pruning of counterposing terms. We consider the following rule-base with reference to Table 6.

Rule #1: (Suitable for surgical anaesthesia)
IF (X(4) IS NOT high) THEN (Y IS class_1)

Rule #2: (Possibly suitable)
IF (X(11) IS high) AND (X(5) IS low) AND (X(3) IS NOT high) AND (X(9) IS high) OR (X(10) IS low) AND (X(11) IS high) AND
(X(3) IS high) OR (X(5) IS low) AND (X(12) IS NOT low) AND
(X(2) IS NOT medium) OR (X(4) IS high) THEN (Y IS class_2)

Rule #3: (Unsuitable for surgical anaesthesia)
IF (X(2) IS low) AND (X(7) IS high) AND (X(8) IS NOT high) THEN (Y IS class_3)

It can be clearly observed from the above rule-base that the complexity of each rule is such that it can be considered trivial. To demonstrate this point we contrast this rule-base to that generated by the GP EFMM mechanism, presented below.

Rule #1: (Suitable for surgical anaesthesia)
IF (X(6) IS high) AND (X(9) IS NOT high) AND (X(30) IS medium) AND (X(4) IS low) OR
(X(4) IS NOT high) AND (X(41) IS NOT high) OR (X(38) IS low) AND (X(5) IS NOT high) OR
(X(8) IS low) AND (X(5) IS low) OR (X(41) IS NOT high) AND (X(3) IS low) AND
(X(13) IS low) AND (X(7) IS NOT medium) AND (X(7) IS high) AND (X(10) IS low) AND
(X(17) NOT high) AND (X(1) IS NOT medium) OR (X(7) IS medium) AND (X(9) IS low)
THEN (Y IS class_1)

Rule #2: (Possibly suitable)
IF (X(40) IS low) AND (X(5) IS high) AND (X(6) IS high) OR (X(5) IS NOT high) AND
(X(8) IS NOT high) AND (X(17) IS NOT medium) AND (X(24) IS low) OR
(X(22) IS NOT medium) AND (X(4) IS high) OR (X(4) IS NOT medium) AND
(X(1) IS NOT high) OR (X(13) IS low) OR (X(5) IS high) OR (X(9) IS high) AND
(X(3) IS medium) AND (X(10) IS low) OR (X(19) IS NOT low) AND (X(4) IS low)
THEN (Y IS class_2)

Rule #3: (Unsuitable for surgical anaesthesia)
IF (X(5) IS low) AND (X(13) IS high) AND (X(8) IS low) AND (X(11) IS high) AND
(X(6) IS low) OR (X(4) IS NOT medium) AND (X(17) IS high) AND (X(12) IS NOT low) AND
(X(3) IS high) AND (X(7) IS NOT low) AND (X(6) IS low) OR (X(28) IS high) OR
(X(37) IS high) AND (X(2) IS high) AND (X(1) IS medium)
THEN (Y IS class_3)

Table 6: Variable descriptions for the evolved rule-base

Variable	Variable description
x(1)	Gender
x(2)	Age
x(3)	Systolic blood pressure
x(4)	Diastolic blood pressure
x(5)	Heart rate
x(6)	SpO2
x(7)	Peak expiratory flow rate
x(8)	Forced Expiratory volume (1 second)
x(9)	Medication indicator
x(10)	Biohazard indicator
x(11)	Haemodynamic indicator
x(12)	Cardiovascular indicator
x(13)	Respiratory indicator
x(14) – x(30)	Heart sound classification (1-17)
x(31) – x(44)	Lung sound classification (1-14)

The number of antecedent terms per class for both EFMM implementations is given in Table 7.

Table 7: Antecedent terms generated for each FIS classification.

Classification	Number of antecedent terms:	
	GA EFMM	GP EFMM
Suitable	1	20
Possibly Unsuitable	11	18
Unsuitable	3	15

As is evident from the rule-base evolved by the GA EFMM summarised in Table 7, there were insufficient terms by which the system could make an accurate appraisal; yet the system attained an accuracy of 87.56%. This is suggestive of the GA EFMM using the ‘suitable for anaesthesia’ category as a default classification: a potentially dangerous mode of operation. In comparison, the GP-based approach resulted in a more complex rule-base which accounted for more input variables. This would suggest that the system offered competition between the rules, rather than the default-state case of the GA-based system.

19.5 CONCLUSIONS

The objective of this work was to attempt to capture and model the expert knowledge of the anaesthetist and hence develop a decision support aid that would assist the anaesthetist in determining a patient's suitability for surgical anaesthesia. To address this objective we decomposed the anaesthetist's perceived decision-making strategy into two sub-problems, these being namely:

- (1) how does the physician diagnose cardio-respiratory disease processes from the associated heart and lung sounds?
- (2) how does the anaesthetist combine the diagnosis with additional patient information in determining their suitability for surgical anaesthesia?

From the work presented in Sec. 3 it can be concluded that an automated means of cardio-respiratory disease process classification is possible using NNs both individually and collectively. In particular we conclude that the frequency spectra of heart and lung sounds provide sufficient information with which to form these classifications.

The final conclusions relate to the function of the DSA presented in Sec. 4; to its design; and to its performance in the context of these objectives. In the context of this work, the main function of the DSA was to encapsulate the knowledge of the anaesthetist, and do so in such a way that an understanding of the decision-making strategy could be imparted in the form of a rule-base. It is concluded that the evolutionary approach to rule-base generation is attractive, due to it being computationally efficient. This approach can confer the additional advantage of not generating overly-complex rules, thereby facilitating an easier understanding of the decisions it makes.

In conclusion, it is possible to encapsulate the knowledge of the anaesthetist in an expert decision support system which determines whether a patient is suitable for surgical anaesthesia. We consider this a demonstration that intelligent systems, such as NNs, fuzzy systems and evolutionary techniques, can be applied to the healthcare domain in eliciting knowledge of the anaesthetist in the context of pre-operative anaesthesia assessment.

References

- Berthold, M. J. and Diamond, J. (1998), Constructive training of probabilistic neural networks, *Neurocomputing*, 19(1-3), pp. 167-183.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*, Oxford University Press, Oxford.
- Darwin, C. R. (1859), On the origin of species by means of natural selection: or the preservation of favoured races in the struggle for life, John Murray, London.
- Denceux, T. (1997), Analysis of evidence-theoretic decision rules for pattern classification, *Pattern Recognition*, 30(7), pp. 1095-1107.
- Goldberg, D. E. (2002), The design of innovation: lessons from and for competent genetic algorithms, Kluwer Academic Publishers, MA.
- Hartman, E. J., Keeler, J. J. and Kowalski, J. M. (1990), Layered neural networks with Gaussian hidden units as universal approximators, *Neural Computation*, 2(2), pp. 210-215.
- Holland, J. H. (1962), Outline for a logical theory of adaptive systems, *Journal of the Association for Computing Machinery*, 9(3), pp. 297-314.
- Holland, J. H. (1975), Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence, University of Michigan Press, Ann Arbor, MI.
- Jang, J.-S. R., Sun, C.-T. and Mizutani, E. (1997), *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*, Prentice Hall Press, Upper Saddle River, NJ.
- Koza, J. R. (1992), Genetic programming: on the programming of computers by natural selection, MIT Press, Cambridge, MA.
- Luo, F.-L. and Unbehauen, R. (1997), *Applied neural networks for signal processing*, Cambridge University Press, Cambridge.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5, pp. 115-133.
- Moody, J. and Darken, C. (1989), Fast learning in networks of locally-tuned processing units, *Neural Computing*, 1(2), pp. 281-294.
- Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986), Learning internal representations by error propagation, *Parallel distributed computing: explorations in the microstructure of cognition*, (Eds. Rumelhart, D. E. and McClelland, J. L.), MIT Press, Cambridge, MA, pp. 318-362.
- Shafer, G. (1976), *A mathematical theory of evidence*, Princeton University Press, Princeton.
- Specht, D. F. (1990), Probabilistic neural networks, *Neural Networks*, 3(1), pp. 109-118.
- Specht, D. F. (1992), Enhancements to probabilistic neural networks, *Proceedings of the IEEE International Joint Conference on Neural Networks*, IEEE, New York, 1992.
- Widrow, B. and Lehr, M. (1990), 30 years of adaptive neural networks: perceptron, madaline and backpropagation, *Proceedings of the IEEE*, 78, pp. 1415-1442.

Index

- acetic acid, 25
- ANSI, 8, 12, 14
- antenna array, 399
- antenna, 199
- AODV, 1, 3, 12, 14
- artificial neural network, 7
- ATLAS, 223, 224, 226, 229, 234
- bootstrap resampling, 514
- Call blocking problem, 312
- cellular radio network, 287, 288
- change detection, 207
- Channel allocation, 301
- channel equalization, 352
- clustering, 351
- Cohen Kappa, 478
- Cohen, 476, 478, 480, 487, 494, 505
- Cohen's Kappa Coefficient, 476, 478
- Cohen's kappa, 487
- Constructive Probabilistic Neural Network, 540
- cost function, 402
- data mining techniques, 513
- data-learning methods, 400
- Davies-Bouldin, 235
- DB, 235, 236, 240
- dedicated path protection, 330
- Dempster-Shafer Theory, 548
- DWDM, 319
- Earth Observation, 199
- electronic nose, 3, 109
- ethanol, 26
- Evolutionary Fuzzy Mixture Model, 550
- failure mechanism, 250
- Fast Fourier Transform, 545
- FCM, 226, 233, 236, 239, 240, 241, 472, 473, 474, 478, 487, 492, 493, 494, 497, 499, 500
- feature selection, 508
- feeding values, 412
- fiber-reinforced plastic, 250
- fuzzy clustering, 470, 472, 498, 474, 488, 505, 507
- Fuzzy logic, 9, 294
- fuzzy membership function, 553
- gas chromatography, 24
- gas sensor, 53
- genetic algorithm, 10, 289, 324, 541
- Genetic Program, 315, 541
- GNMM, 5, 11
- graph Laplacian, 206
- insensitive loss function, 428
- Kappa, 476, 478, 480, 487, 491, 494, 496, 497, 503
- KM, 471, 472, 474, 478, 487, 489, 490, 491, 494, 495, 496, 497
- K-Means, 471, 505

- Kohonen, 461, 463, 506
- lampante, 26
- laplacian SVM, 205
- Levinson-Durbin algorithm, 546
- Link stability, 7, 8
- MANET, 1, 2, 6
- MIMO systems, 354
- MLP, 227, 228, 233, 241, 242, 243, 246
- MOS sensors, 26
- Multilayer Perceptron, 537
- multi-temporal image classification, 207
- NPV, 244
- NSLBP, 459, 460, 461, 482, 495, 497, 504
- OAV, 26
- odour threshold, 22
- one-class support vector machine, 204
- one-class SVM, 201
- optimization, 54
- OSTBC systems, 365
- pattern recognition, 54, 250
- PCA, 226, 227, 238, 239, 240, 243, 244, 246
- plane waves, 425
- PPV, 244
- Principal Component Analysis, 226
- Probabilistic Neural Network, 539
- protection, 322
- quadratic programming, 206
- Radial Basis Function Network, 116, 538
- radiation pattern, 412
- real properties, 415
- regression risk, 428
- REGRESSION, 401
- remote sensing, 199
- restoration, 321
- RWA, 321
- Self Organized Maps, 227
- semi-definite, 203
- semi-supervised learning, 201
- sensitivity, 244, 247
- sensorgram, 38
- sensory defects, 24
- shared path protection, 324
- SILS, 8, 9, 11, 12
- smart antennas, 423
- SOM, 227, 233, 239, 240, 241, 242
- SOM, 460, 461, 462, 463, 464, 465, 466, 467, 468, 470, 474, 478, 479, 480, 481, 483, 484, 485, 486, 487, 488, 489, 490, 492, 493, 496, 497, 498, 503, 506, 507
- spatial correlation matrix, 427
- specificity, 244, 247
- spectrum, 199
- squared correlation coefficient, 118
- support vector classifier, 203
- support vector domain description, 204
- support vector machine
- support vector machines, 109, 200, 400, , 438, 512
- support vector regression, 424
- Survivability, 321
- swarm intelligence, 1, 4, 6, 8
- synthesis, 400
- synthetic aperture radar, 199
- tapped delay, 118

TDNN, 228, 243
temperature modulation, 54
template, 407
time-varying channels, 353
training set, 429
two-level clustering, 470, 478
unsupervised training, 355
Vapnik Chervonenkis dimension,
109
Ventricular fibrillation, 505