

NASA Conference Publication 10112, Vol. 1

# NORTH AMERICAN FUZZY LOGIC PROCESSING SOCIETY (NAFIPS '92)



*Proceedings of a Conference held in  
PUERTO VALLARTA, MEXICO*

*December 15-17, 1992*

(NASA-CP-10112-Vol-1) NORTH  
AMERICAN FUZZY LOGIC PROCESSING  
SOCIETY (NAFIPS 1992), VOLUME 1  
(NASA) 376 p

N93-29516  
--THRU--  
N93-29555  
Unclass

G3/59 0161800

---

**NASA**

---

*NASA Conference Publication 10112, Vol. 1*

# **NORTH AMERICAN FUZZY INFORMATION PROCESSING SOCIETY (NAFIPS '92)**

*James Villarreal, Compiler  
NASA Lyndon B. Johnson Space Center  
Houston, Texas*

Proceedings of a conference sponsored by the North American Fuzzy Information Processing Society in cooperation with the National Aeronautics and Space Administration, the Instituto Tecnológico de Morelia, the Indian Society for Fuzzy Mathematics and Information Processing, the Instituto Tecnológico de Estudios Superiores de Monterrey, the International Fuzzy Theory and Systems, and the Microelectronics and Computer Technology Corporation, and held at the Melia Hotel Pasco de la Marina Sur, Marina Vallarta, Puerto Vallarta, Mexico

December 15-17, 1992

National Aeronautics and Space Administration  
Lyndon B. Johnson Space Center  
Houston, Texas  
1992

NAFIPS '92, an international conference on fuzzy set theory and applications, is sponsored by NAFIPS, in cooperation with:

- National Aeronautics and Space Administration (NASA)
- Instituto Tecnológico de Morelia
- Indian Society for Fuzzy Mathematics and Information Processing (ISFUMIP)
- Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM)
- International Fuzzy Systems Association (IFSA)
- Japan Society for Fuzzy Theory and Systems
- Microelectronics and Computer Technology Corporation (MCC)

Fuzzy set theory has led to a large number of diverse applications. Recently, interesting applications have been developed which involve the integration of fuzzy systems with adaptive processes such as neural networks and genetic algorithms. NAFIPS '92 will be directed toward the advancement, commercialization, and engineering development of these technologies.

The conference will consist of both plenary sessions and contributory sessions. The plenary sessions will be addressed by leading experts. Topics to be discussed at this conference include the following:

- Biomedical and Biochemical Issues
- Business and Decision Making
- Commercial Products and Tools
- Computer Systems and Information Processing
- Control Systems
- Decision Analysis
- Foundations and Mathematical Issues
- Genetic Algorithms
- Hardware
- Image Processing and Vision
- Neural Networks
- Optimization
- Path Planning
- Pattern Recognition
- Robotics



## TUTORIALS

Tutorials by leading experts will be provided on December 14, 1992.

- 8:00 - 9:40            **Introduction to Fuzzy Sets and Approximate Reasoning**  
RONALD R. YAGER, Iona College, New Rochelle, NY, USA
- 9:50 - 11:30          **Fuzzy Intelligent Information Systems**  
M. ZEMANKOVA, NATIONAL Science Foundation, Washington, DC, USA
- 11:30 - 12:30        **Lunch**
- 12:30 - 2:10          **Fuzzy Logic In Expert System and Its Applications for IE/OR/MS**  
I.B. TURKSEN, University of Toronto, Toronto, ON, CANADA
- 2:20 - 4:00           **Fuzzy Control and Its Applications**  
M. SUGENO, Tokyo Institute of Technology, Yokohama, JAPAN
- 4:10 - 5:50           **Fuzzy Hardware Design and Its Applications**  
K. HIROTA, Hosei University, Tokyo, JAPAN

## CONFERENCE

Tuesday, December 15, 1992

- 8:00                    **Welcoming Remarks**
- 8:15-9:00              **Plenary Speech**  
PROFESSOR LOTFI ZADEH, University of California at Berkeley
- 9:00 - 12:00          **Parallel Sessions**

### Fuzzy Theory & Systems

**An Analysis of Possible Applications of Fuzzy Set Theory to the Credibility Theory**  
KRZYSZTOF OSTASZEWSKI, University of Louisville, Louisville, KY  
WALDEMAR KARWOWSKI, University of Louisville, Louisville, KY

**Estimations of Expectedness and Potential Surprise in Possibility Theory**  
HENRI PRADE, Universite Paul Sabatier, Toulouse Cedex, FRANCE  
RONALD R. YAGER, Iona College, New Rochelle, NY

**Comparison of Specificity and Information for Fuzzy Domains**  
ARTHUR RAMER, University of New South Wales, Kensington, AUSTRALIA

**The Axiomatic Definition of a Linguistic Scale Fuzziness Degree, Its Major Properties and Applications**  
ALEXANDER P. RYJOV, Soviet Association of Fuzzy Systems, Moscow, RUSSIA

**How to Select Combination Operators for Fuzzy Expert Systems Using CRI**  
I.B. TURKSEN, University of Toronto, Toronto, Ontario, CANADA  
Y. TIAN, University of Toronto, Toronto, Ontario, CANADA

**Approximate Reasoning Using Terminological Models**  
JOHN YEN, Texas A&M University, College Station, TX  
NITIN VAIDYA, Texas A&M University, College Station, TX

## Image Analysis

### **Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions**

RAGHU KRISHNAPURAM, University of Missouri, Columbia, MO  
JAMES M. KELLER, University of Missouri, Columbia, MO  
YIBING MA, University of Missouri, Columbia, MO

### **A Fuzzy Clustering Algorithm to Detect Planar and Quadric Shapes**

RAGHU KRISHNAPURAM, University of Missouri, Columbia, MO  
HICHEM FRIGUI, University of Missouri, Columbia, MO  
OLFA NASRAOUI, University of Missouri, Columbia, MO

### **A Fuzzy Measure Approach to Motion Frame Analysis for Scene Detection**

ALBERT B. LEIGH, McDonnell Douglas Space Systems, Houston, TX  
SANKAR K. PAL, Indian Statistical Institute, Calcutta, INDIA

### **Automatic Rule Generation for High-Level Vision**

FRANK CHUNG-HOON RHEE, University of Missouri, Columbia, MO  
RAGHU KRISHNAPURAM, University of Missouri, Columbia, MO

### **Encoding Spatial Images - A Fuzzy Set's Theory Approach**

LESZEK M. SZTANDERA, University of Toledo, Toledo, OH

### **Image Segmentation Using LVQ Clustering Networks**

ERIC CHEN-KUO TSAO, The University of West Florida, Pensacola, FL  
JAMES C. BEZDEK, The University of West Florida, Pensacola, FL  
NIKHIL R. PAL, The University of West Florida, Pensacola, FL

12:00 - 1:00            **Lunch**

1:00 - 3:30            **Parallel Sessions**

## Adaptive Learning

### **A Neuro-Fuzzy Architecture for Real-Time Applications**

P. A. RAMAMOORTHY, University of Cincinnati, Cincinnati, OH  
SONG HUANG, University of Cincinnati, Cincinnati, OH

### **A Composite Self Tuning Strategy for Fuzzy Control of Dynamic Systems**

C-Y SHIEH, University of Missouri, Columbia, MO  
SATISH S. NAIR, University of Missouri, Columbia, MO

### **A Self-Learning Rule Base for Command Following in Dynamical Systems**

WEI K. TSAI, University of California at Irvine, Irvine, CA  
HON-MUN LEE, University of California at Irvine, Irvine, CA  
ALEXANDER PARLOS, Texas A&M University, College Station, TX

### **Adaptive Defuzzification for Fuzzy Systems Modeling**

RONALD R. YAGER, Iona College, New Rochelle, NY  
DIMITAR P. FILEV, Iona College, New Rochelle, NY

### **Design Issues of a Reinforcement-Based Self-Learning Fuzzy Controller for Petrochemical Process Control**

JOHN YEN, Texas A&M University, College Station, TX  
HAQJIN WANG, Texas A&M University, College Station, TX  
WALTER C. DAUGHERITY, Texas A&M University, College Station, TX

## Applications

### **Learning Characteristics of a Space Time Neural Network as a Tether Skitrope Observer**

ROBERT N. LEA, NASA/Johnson Space Center, Houston, TX  
JAMES A. VILLARREAL, NASA/Johnson Space Center, Houston, TX  
JANI YASHVANT, Togai Infralogic Inc., Houston, TX  
CHARLES COPELAND, Loral Space Systems, Houston, TX

### **Clustering of Tethered Satellite System Simulation Data by an Adaptive Neuro-Fuzzy Algorithm**

SUNANDA MITRA, Texas Tech University, Lubbock, TX  
SURYA PEMMARAJU, Texas Tech University, Lubbock, TX

### **Character Recognition Using a Neural Network Model with Fuzzy Representation**

NASSRIN TAVAKOLI, University of North Carolina at Charlotte, Charlotte, NC  
DAVID SENIW, University of North Carolina at Charlotte,  
Charlotte, NC

### **Designing a Fuzzy Scheduler for Hard Real-Time Systems**

JOHN YEN, Texas A&M University, College Station, TX  
JONATHAN LEE, Texas A&M University, College Station, TX  
NATHAN PFLUGER, Texas A&M University, College Station, TX  
SWAMI NATARAJAN, Texas A&M University, College Station, TX

### **WARP: Weight Associative Rule Processor A Dedicated VLSI Fuzzy Logic Megacell**

ANDREA PAGNI, SGS-Thompson Microelectronics, Agrate Brianza (MI) ITALY  
R. POLUZZI, SGS-Thompson Microelectronics, Agrate Brianza (MI) ITALY  
G. G. RIZZOTTO, SGS-Thompson Microelectronics, Agrate Brianza (MI) ITALY

## **Wednesday, December 16, 1992**

8:00 - 8:45

**Plenary Speech**

Piero Bonissone, "Fuzzy Logic Control: From Development to Deployment (with an Application to Aircraft Engine Control)"

8:45 - 10:45

**Parallel Sessions**

## Decision Analysis

### **Evaluation of Fuzzy Inference Systems Using Fuzzy Least Squares**

JOSEPH M. BARONE, Loki Software, Inc., Liberty Corner, NJ

### **A Model for Amalgamation in Group Decision Making**

VINCENZO CUTELLO, Consorzio per la Ricerca sulla Microelettronica del Mezzogiorno, Catania, ITALY  
JAVIER MONTERO, Complutense University, Madrid, Spain

### **Fuzzy Forecasting and Decision Making in Short Dynamic Time Series**

EFIN JA. KARPOVSKY, Odessa Institute of National Economy, Odessa, UKRAINE

### **Decision Analysis With Approximate Probabilities**

THOMAS WHALEN, Georgia State University, Atlanta, GA

## Applications

### **Distributed Traffic Signal Control Using Fuzzy Logic**

STEPHEN CHIU, Rockwell International Science Center,  
Thousand Oaks, CA

### **Intelligent Virtual Reality in the Setting of Fuzzy Sets**

JOHN T. DOCKERY, George Mason University, Fairfax, VA  
DAVID LITTMAN, George Mason University, Fairfax, VA

### **Comparison of Crisp and Fuzzy Character Networks In Handwritten Word Recognition**

PAUL GADER, University of Missouri, Columbia, MO  
MAGDI MOHAMED, University of Missouri, Columbia, MO  
JUNG-HSIEN CHIANG, University of Missouri, Columbia, MO

### **Fuzzy Neural Network Methodology Applied to Medical Diagnosis**

MARIAN B. GORZALCZANY, Technical University of Kielce, Kielce, POLAND  
MARY DEUTSCH-MCLEISH, University of Guelph, Guelph, Ontario, CANADA

11:00 - 12:00      **Parallel Sessions**

## Decision Analysis

### **An Experimental Methodology for a Fuzzy Set Preference Model**

I.B. TURKSEN, University of Toronto, Toronto, ON, CANADA  
IAN A. WILLSON, University of Toronto, Toronto, ON, CANADA

### **A Fuzzy Set Preference Model for Market Share Analysis**

I.B. TURKSEN, University of Toronto, Toronto, ON, CANADA  
IAN A. WILLSON, University of Toronto, Toronto, ON, CANADA

## Knowledge Representation

### **Information Compression in the Context Model**

JORG GEBHARDT, Technical University of Braunschweig, Braunschweig, GERMANY  
RUDOLF KRUSE, Technical University of Braunschweig, Braunschweig, GERMANY  
DETLEF NAUCK, Technical University of Braunschweig, Braunschweig, GERMANY

### **Fuzzy Knowledge Base Construction Through Belief Networks Based on Lukasiewicz Logic**

FELIPE LARA-ROSANO, Universidad Nacional Autonoma de Mexico, Mexico DF, MEXICO

12:00 - 1:00      **Lunch**

1:00 - 3:30      **Parallel Sessions**

## Control Systems

### **Intelligent Fuzzy Controller for Event-Driven Real Time Systems**

JANOS GRANTNER, University of Minnesota, Minneapolis, MN  
MAREK PATYRA, University of Minnesota, Minneapolis, MN  
MARIAN S. STACHOWICZ, University of Minnesota, Minneapolis, MN

### **Fuzzy Coordinator In Control Problems**

A. RUEDA, University of Manitoba, Winnipeg, Manitoba, CANADA  
W. PEDRYCZ, University of Manitoba, Winnipeg, Manitoba, CANADA

**Tuning a Fuzzy Controller Using Quadratic Response Surfaces**

BRIAN SCHOTT, Georgia State University, Atlanta, GA  
THOMAS WHALEN, Georgia State University, Atlanta, GA

**The Cognitive Bases for the Design of a New Class of Fuzzy Logic Controllers: The Clearness Transformation Fuzzy Logic Controller**

LABIB SULTAN, York University, Toronto, Ontario, CANADA  
TALIB JANABI, Mentalogic Systems Inc., Markham,  
Ontario, CANADA

**A Fuzzy Control Design Case: The Fuzzy PLL**

H.N. TEODORESCU, Polytechnic Institute of Iasi, ROMANIA  
I. BOGDAN, Polytechnic Institute of Iasi, ROMANIA

**Adaptive Learning**

**Adding Dynamic Rules to Self-Organizing Fuzzy Systems**

CATALIN V. BUHUSI, Romanian Academy, Calea Copou, Iasi, ROMANIA

**Fuzzy Learning Under and About an Unfamiliar Fuzzy Teacher**

BELUR V. DASARATHY, Dynetics, Huntsville, AL

**Some Problems with the Design of Self-Learning Management Systems**

ZINY FLIKOP, NYNEX Science and Technology, Inc.,  
White Plains, NY

**A Neural Fuzzy Controller Learning by Fuzzy Error Propagation**

DETLEF NAUCK, Technical University of Braunschweig, Braunschweig, GERMANY  
RUDOLF KRUSE, Technical University of Braunschweig, Braunschweig, GERMANY

**Thursday December 17, 1992**

**8:00 - 10:00          Parallel Sessions**

**Decision Analysis**

**Determining Rules for Closing Customer Service Centers: A Public Utility Company's Fuzzy Decision**

ANDRE DEKORVIN, University of Houston - Downtown, Houston, TX  
MARGARET F. SHIPLEY, University of Houston - Downtown,  
Houston, TX  
ROBERT N. LEA, NASA/Johnson Space Center, Houston, TX

**Fuzzy Simulation in Concurrent Engineering**

A. KRASLAWSKI, Lappeenranta University of Technology, Lappeenranta, FINLAND  
L. NYSTROM, Lappeenranta University of Technology, Lappeenranta, FINLAND

**Inverse Problems: Fuzzy Representation of Uncertainty Generates a Regularization**

V. KREINOVICH, University of Texas at El Paso, El Paso, TX  
CHING-CHUANG CHANG, University of Texas at El Paso, El Paso, TX  
L. REZNIK, Victoria University of Technology, MMC Melbourne,  
VIC 3000, AUSTRALIA  
G. N. SOLOPCHENKO, St. Petersburg Technical University, St. Petersburg, RUSSIA

### **Quantification of Human Responses**

RALPH C. STEINLAGE, University of Dayton, Dayton, OH  
T. E. GANTNER, University of Dayton, Dayton, OH  
P. Y. W. LIM, Boise Cascade R&D, Portland, OR

## **Fuzzy Theory & Systems**

### **Non-Scalar Uncertainty**

SALVADOR GUTIERREZ-MARTINEZ, Instituto Tecnológico de Morelia, Morelia, MEXICO

### **Comparison Between the Performance of Two Classes of Fuzzy Controllers**

TALIB H. JANABI, Mentalogic Systems Inc., Markham, Ontario, CANADA  
L.H. SULTAN, York University, Toronto, Ontario, CANADA

### **Possibilistic Measurement and Set Statistics**

CLIFF JOSLYN, SUNY-Binghamton, Portland, ME

### **The Fusion of Information via Fuzzy Integration**

JIM KELLER, University of Missouri, Columbia, MO  
HOSSEIN TAHANI, University of Missouri, Columbia, MO

10:15 - 11:45      **Parallel Sessions**

## **Database Management**

### **On the Evaluation of Fuzzy Quantified Queries in a Database Management System**

PATRICK BOSCH, IRISA/ENSSAT, Lannion, Cedex, FRANCE  
OLIVIER PIVERT, IRISA/ENSSAT, Lannion, Cedex, FRANCE

### **A Fuzzy Case Based Reasoning Tool for Model Based Approach to Rocket Engine Health Monitoring**

SRINIVAS KROVIDY, University of Cincinnati, Cincinnati, OH  
ADAM NOLAN, University of Cincinnati, Cincinnati, OH  
YONG LIN HU, University of Cincinnati, Cincinnati, OH  
WILLIAM G. WEE, University of Cincinnati, Cincinnati, OH

### **A High Performance, Ad-Hoc Fuzzy Query Processing System for Relational Databases**

W.H. MANSFIELD, Bellcore, Cambridge, MA, USA  
ROBERT M. FLEISCHMAN, BBN, Cambridge, MA, USA

## **Genetic Algorithms/Optimization**

### **Genetic Algorithms in Adaptive Fuzzy Control**

C. LUCAS KARR, U. S. Department of Interior Bureau of Mines, Tuscaloosa, AL

### **A Genetic Algorithms Approach for Altering the Membership Functions in Fuzzy Logic Controllers**

HANA SHEHADEH, LinCom Corporation, Houston, TX  
ROBERT N. LEA, NASA/Johnson Space Center, Houston, TX

### **Fuzzy Multiple Linear Regression - A Computational Approach**

C.H. JUANG, Clemson University, Clemson, SC  
X.H. HUANG, Clemson University, Clemson, SC  
J.W. FLEMING, Clemson University, Clemson, SC

12:00 - 1:00 Lunch

1:00 - 4:30 Parallel Sessions

### Neural Networks

#### **Incorporation of Varying Types of Temporal Data In a Neural Network**

M. E. COHEN, California State University, Fresno, CA  
D.L. HUDSON, California State University, Fresno, CA

#### **Fuzzy Operators and Cyclic Behaviour In Formal Neural Networks**

E. LABOS, Semmelweis University Medical School, Budapest, HUNGARY  
A. V. HOLDEN, The University of Leeds, Leeds, UK  
J. LACZKO, Ludwig Maximilien University, Munchen, GERMANY  
A. S. LABOS, Semmelweis University Medical School, Budapest, HUNGARY

#### **Neural Networks: A Simulation Technique Under Uncertainty Conditions**

LUISA MCALLISTER, Moravian College, Bethlehem, PA

#### **Incomplete Fuzzy Data Processing Using Artificial Neural Network**

MAREK J. PATYRA, University of Minnesota, Duluth, MN

#### **Stochastic Architecture for Hopfield Neural Nets**

SANDY PAVEL, Polytechnical Institute of Iasi, Iasi, ROMANIA

#### **Hierarchical Model of Matching**

W. PEDRYCZ, University of Manitoba, Winnipeg, Manitoba, CANADA  
EUGENE ROVENTA, York University, Toronto, Ontario, CANADA

#### **A Conjugate Gradients/Trust Regions Algorithm for Training Multilayer Perceptrons for Nonlinear Mapping**

RAGHAVENDRA K. MADYASTHA, Rice University, Houston, TX  
BEHNAAM AAZHANG, Rice University, Houston, TX  
TROY F. HENSON, IBM Corporation, Houston, TX  
WENDY L. HUXHOLD, IBM Corporation, Houston, TX

### Fuzzy Theory & Systems

#### **On Probability-Possibility Transformations**

GEORGE KLIR, State University of New York, Binghamton, NY  
BEHZAD PARVIZ, California State University, Los Angeles, CA

#### **Inference In Fuzzy Rule with Conflicting Evidence**

LASZLO T. KOCZY, Technical University of Budapest, Budapest, HUNGARY

#### **Gaussian Membership Functions are Most Adequate In Representing Uncertainty In Measurements**

V. KREINOVICH, University of Texas at El Paso, El Paso, TX  
C. QUINTANA, University of Michigan at Ann Arbor, Ann Arbor, MI  
L. REZNIK, Victoria University of Technology, MMC Melbourne,  
VIC 3000, AUSTRALIA

#### **Applying the Metric Truth Approach to Fuzzified Automated Reasoning**

VESA A. NISKANEN, University of Helsinki, Helsinki, FINLAND

#### **Life Insurance Risk Assessment Using a Fuzzy Logic Expert System**

L. A. CARRENO, Togai InfraLogic, Houston, TX  
R. A. STEEL, Togai InfraLogic, Houston, TX

## CONTENTS

An Analysis of Possible Applications of Fuzzy Set Theory to the Actuarial Credibility Theory.....	1-1
Estimations of Expectedness and Potential Surprise in Possibility Theory .....	7-2
Comparison of Specificity and Information for Fuzzy Domains .....	14-3
The Axiomatic Definition of a Linguistic Scale Fuzziness Degree, Its Major Properties and Applications .....	21-4
How to Select Combination Operators for Fuzzy Expert Systems Using CRI .....	29-5
Approximate Reasoning Using Terminological Models .....	39-6
Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions .....	49-7
A Fuzzy Clustering Algorithm to Detect Planar and Quadric Shapes.....	59-8
A Fuzzy Measure Approach to Motion Frame Analysis for Scene Detection .....	69-9
Automatic Rule Generation for High-Level Vision .....	81-10
Encoding Spatial Images - A Fuzzy Set Theory Approach.....	89-11
Image Segmentation Using LVQ Clustering Networks .....	98-12
A Neuro-Fuzzy Architecture for Real-Time Applications .....	108-13
A Composite Self Tuning Strategy for Fuzzy Control of Dynamic Systems .....	118-14
A Self-Learning Rule Base for Command Following in Dynamical Systems.....	125-15
Adaptive Defuzzification for Fuzzy Systems Modeling .....	135-16
Design Issues of a Reinforcement-Based Self-Learning Fuzzy Controller for Petrochemical Process Control .....	143-17
Learning Characteristics of a Space-Time Neural Network as a Tether "Skiprope Observer" .....	154-18
Clustering of Tethered Satellite System Simulation Data by an Adaptive Neuro-Fuzzy Algorithm .....	166-19
Character Recognition Using a Neural Network Model with Fuzzy Representation.....	175-20
Designing a Fuzzy Scheduler for Hard Real-Time Systems .....	185-21
WARP: Weight Associative Rule Processor, A Dedicated VLSI Fuzzy Logic Megacell.....	195-22
Evaluation of Fuzzy Inference Systems Using Fuzzy Least Squares .....	205-23
A Model for Amalgamation in Group Decision Making .....	215-24
Fuzzy Forecasting and Decision Making in Short Dynamic Time Series .....	224-25
Decision Analysis With Aproximate Probabilities .....	229-26
Distributed Traffic Signal Control Using Fuzzy Logic.....	239-27
Intelligent Virtual Reality in the Setting of Fuzzy Sets .....	248-28
Comparison of Crisp and Fuzzy Character Networks in Handwritten Word Recognition.....	257-29
Fuzzy Neural Network Methodology Applied to Medical Diagnosis.....	266-30

An Experimental Methodology for a Fuzzy Set Preference Model .....	276	-31
A Fuzzy Set Preference Model for Market Share Analysis.....	286	-32
Information Compression in the Context Model .....	296	-33
Fuzzy Knowledge Base Construction Through Belief Networks Based on Lukasiewicz Logic .....	304	-34
Intelligent Fuzzy Controller for Event-Driven Real Time Systems.....	312	-35
Fuzzy Coordinator in Control Problems.....	322	-30
Tuning a Fuzzy Controller Using Quadratic Response Surfaces .....	330	-37
The Cognitive Bases for the Design of a New Class of Fuzzy Logic Controllers: The Clearness Transformation Fuzzy Logic Controller .....	340	-38
A Fuzzy Control Design Case: The Fuzzy PLL .....	350	-39
Adding Dynamic Rules to Self-Organizing Fuzzy Systems .....	360	
Fuzzy Learning Under and About an Unfamiliar Fuzzy Teacher.....	368	
Some Problems With the Design of Self-Learning Management Systems .....	378	
A Neural Fuzzy Controller Learning by Fuzzy Error Propagation .....	388	
Determining Rules for Closing Customer Service Centers: A Public Utility Company's Fuzzy Decision .....	398	
Fuzzy Simulation in Concurrent Engineering .....	408	
Inverse Problems: Fuzzy Representation of Uncertainty Generates a Regularization.....	418	
Quantification of Human Responses .....	427	
Non-Scalar Uncertainty .....	437	
Comparison Between the Performance of Two Classes of Fuzzy Controllers .....	449	
Possibilistic Measurement and Set Statistics .....	458	
The Fusion of Information Via Fuzzy Integration.....	468	
On the Evaluation of Fuzzy Quantified Queries in a Database Management System .....	478	
A Fuzzy Case Based Reasoning Tool for Model Based Approach to Rocket Engine Health Monitoring.....	488	
A High Performance, Ad-Hoc, Fuzzy Query Processing System for Relational Databases .....	496	
Genetic Algorithms in Adaptive Fuzzy Control .....	506	
A Genetic Algorithms Approach for Altering the Membership Functions in Fuzzy Logic Controllers .....	515	
Fuzzy Multiple Linear Regresssion - A Computational Approach .....	524	
Incorporation of Varying Types of Temporal Data in a Neural Network .....	535	
Fuzzy Operators and Cyclic Behavior in Formal Neuronal Networks.....	545	
Neural Networks: A Simulation Technique Under Uncertainty Conditions .....	555	
Incomplete Fuzzy Data Processing System Using Artificial Neural Network.....	565	

Stochastic Architecture for Hopfield Neural Nets .....	575
Hierarchical Model of Matching .....	581
A Conjugate Gradients/Trust Regions Algorithm for Training Multilayer Perceptrons for Nonlinear Mapping .....	589
On Probability-Possibility Transformations .....	598
Inference in Fuzzy Rule Bases with Conflicting Evidence .....	608
Gaussian Membership Functions Are Most Adequate in Representing Uncertainty in Measurements .....	618
Applying the Metric Truth Approach to Fuzzified Automated Reasoning .....	625
Life Insurance Risk Assessment Using a Fuzzy Logic Expert System.....	627
Author Index .....	A-1

# An Analysis of Possible Applications of Fuzzy Set Theory to the Actuarial Credibility Theory

Krzysztof Ostaszewski<sup>1</sup> and Waldemar Karwowski<sup>2</sup>

<sup>1</sup>Department of Mathematics

<sup>2</sup>Center for Industrial Ergonomics

Department of Industrial Engineering

University of Louisville

Louisville, Kentucky 40292, U.S.A.

Telephone: (502) 588-6829; (502) 588-7173

Fax: (502) 588-0150; (502) 588-7397

## ABSTRACT

In this work we review basic concepts of the actuarial credibility theory from the point of view of introducing applications of fuzzy set-theoretic method. We show how the concept of actuarial credibility can be modeled through the fuzzy set membership functions, and how fuzzy set methods, especially fuzzy pattern recognition, can provide an alternative tool for estimating credibility.

## INTRODUCTION

Credibility theory is one of the most fundamental tools of actuarial science applied to casualty and property insurance. Casualty and property insurance are characterized by high frequency of claims (even for the same individual or group), and significantly more variable patterns of both claim frequency and severity. On the other hand, the time until payment, or until a failure of a status, are of less importance, as claims arise so frequently.

## THE CONCEPT OF ACTUARIAL CREDIBILITY

The simplest description of credibility can be as the measure that an actuary believes should be attached to a given body of data about risks considered for insurance for rate-making purposes. To say that data is "*fully credible*" means that the data is sufficient for setting the premium rates based on it, while the data concerning loss experience is "*too small to be credible*" if we believe that the future experience may well be very different, and that we have more confidence in the knowledge prior to data collection.

For example, data concerning personal automobile liability insurance loss experience in the state of Kentucky is "*fully credible*" if it is adequate for rate levels in the state without reference to any previous data, or other states or countries experience. The standard mathematical models of credibility produces a number Z between 0 and 1 which is a measure of credibility assigned to the data, while 1 - Z is treated as a measure of credibility assigned to the alternative (e.g., previous data, or other states' experience, in the case of personal automobile liability insurance in Kentucky). We then have

$$C = ZR + (1-Z)H$$

\*The first author was partially supported by a University of Louisville research grant

where  $R$  is the mean loss calculated from the current observation, it is the prior mean, and  $C$  is the compromise estimate used for setting the net premium.

## DETERMINATION OF CREDIBILITY

Mathematical models of actuarial credibility assume generally that losses are generated randomly by the distribution of a variable of the form

$$Y = X_1 + X_2 + \dots + X_N$$

where  $N$  is the random claim frequency, while each  $X_i$ , a random variable as well, corresponds to the individual claim severity. If  $N$  is assumed to have the Poisson distribution, the variables  $X_i$  are independent identically distributed, and we adopt the approach of interval estimation, the credibility  $Z$  can be estimated as

$$Z = \sqrt{\frac{N}{n_F}}$$

where  $N$  is the observed number of losses, and

$$n_F = \frac{y^2}{k^2} \left( 1 + \left( \frac{\sigma}{m} \right)^2 \right)$$

Here,  $k$  is the fluctuation limit away from the mean of total claims,  $y$  is the prescribed confidence interval boundary for the standard normal distribution, and  $\sigma/m$  is the coefficient of variation of the individual claim severity distribution. An alternative method (Herzog, 1992) is to evaluate the posterior total claim size distribution using the classical Bayesian approach. The third standard method is the Bühlman's (1967) credibility estimate

$$Z = \frac{n}{n + K}$$

where  $n$  is the number of exposure units in the experience and  $K$  is the ratio of the expected value of process variance to the variance of hypothetical means.

## DETERMINATION OF CREDIBILITY WITH FUZZY PATTERN RECOGNITION

Ostaszewski (1992) gives an extensive discussion of applicability of fuzzy set theoretic methods in actuarial science. He points out that pattern recognition methods can be applied directly to classification of risks, thus creating an alternative rate-making approach. If

$$\underline{x}_1 = \left( x_1^{(1)}, x_2^{(1)}, \dots, x_p^{(1)} \right)$$

$$\underline{x}_2 = \left( x_1^{(2)}, x_2^{(2)}, \dots, x_p^{(2)} \right)$$

...

$$\underline{x}_n = \left( x_1^{(n)}, x_2^{(n)}, \dots, x_p^{(n)} \right)$$

is the data set representing the historical loss experience, and

$$\underline{y} = (y_1, \dots, y_p)$$

represents data concerning the recent experience (vector coordinates represent risk characteristics and loss features), one can use a clustering algorithm (see Ostaszewski, 1992, for an example of such direct application and further references) to assign  $\underline{y}$  to fuzzy clusters in data. If  $\mu$  is the maximum membership degree of  $\underline{y}$  in a cluster, the number  $Z = 1 - \mu$  could be used as the credibility measure of the experience provided by  $\underline{y}$ , while  $\mu$  gives the membership degree for the historical experience indicated by the cluster.

Using our previous automobile rate-making example, consider an insurer with historical experience in the states of Ohio, Pennsylvania and California, extending her business to Kentucky. The insurer can cluster new data from Kentucky into patterns from other states, and arrive at a credibility reading of her loss experience in Kentucky versus the historical net premiums from Ohio, Pennsylvania and California (or subsets of this three-element set, if clustering so indicates).

Assume, hypothetically, that the mean claims and the standard deviations of claims for Ohio, Pennsylvania, and California are:

$$\text{Ohio: } \mu_1 = 100, \sigma_1 = 25;$$

$$\text{Pennsylvania: } \mu_2 = 125, \sigma_2 = 30;$$

$$\text{California: } \mu_3 = 175, \sigma_3 = 50.$$

Let Kentucky experience be  $\mu_4 = 200, \sigma_4 = 40$ . Assuming equal probability for each of the three historical states, and using Bühlman's (1967) actuarial credibility formula we get:

$$K = \frac{\text{Expected value of process variance}}{\text{Variance of hypothetical mean}} =$$

$$\begin{aligned}
&= \frac{\frac{1}{3}(25)^2 + \frac{1}{3}(30)^2 + \frac{1}{3}(50)^2}{\frac{1}{3}\left(\left(100 - \frac{400}{3}\right)^2 + \left(125 - \frac{400}{3}\right)^2 + \left(175 - \frac{400}{3}\right)^2\right)} \\
&\approx 1.38
\end{aligned}$$

and

$$Z = \frac{n}{n + K} = \frac{3}{3 + 1.38} = 0.6849$$

We have, therefore:

$$\begin{aligned}
C &= ZR + (1 - Z)H = \\
&0.6849 \cdot 200 + (1 - 0.6849) \frac{400}{3} \approx 179.
\end{aligned}$$

On the other hand, if we consider just the means and standard deviations as features, and treat the data from the four states as four feature vectors:

$$\underline{x}_1 = \begin{bmatrix} 100 \\ 25 \end{bmatrix}, \underline{x}_2 = \begin{bmatrix} 125 \\ 30 \end{bmatrix}, \underline{x}_3 = \begin{bmatrix} 175 \\ 50 \end{bmatrix}, \underline{x}_4 = \begin{bmatrix} 200 \\ 40 \end{bmatrix}$$

Then we can use clustering methods to analyze them. We will use the classical Bezdek's (1981) clustering algorithm specified by a matrix

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix},$$

parameter  $m = 2$ , initial partition

$$\bar{U}^{(0)} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the stopping parameter  $\epsilon = 0.3$ .

The first step cluster centers are

$$v_1^{(0)} = \begin{bmatrix} 133.33 \\ 31.67 \end{bmatrix}, v_2^{(0)} = \begin{bmatrix} 200 \\ 40 \end{bmatrix}.$$

This results in a new partition

$$\tilde{U}^{(1)} = \begin{bmatrix} 0.8956 & 0.9870 & 0.251 & 0 \\ 0.1044 & 0.0130 & 0.7479 & 1 \end{bmatrix}$$

Using the standard matrix norm we get

$$\|\tilde{U}^{(0)} - \tilde{U}^{(1)}\| \approx 1.068 > 0.3.$$

The second step cluster centers are

$$v_1^{(1)} = \begin{bmatrix} 115.828 \\ 28.511 \end{bmatrix}, v_2^{(1)} = \begin{bmatrix} 190.393 \\ 45.457 \end{bmatrix}$$

The second step partition is:

$$\tilde{U}^{(2)} = \begin{bmatrix} 0.9697 & 0.9811 & 0.0695 & 0.0169 \\ 0.0303 & 0.0189 & 0.9305 & 0.9831 \end{bmatrix}$$

and  $\|\tilde{U}^{(2)} - \tilde{U}^{(1)}\| \approx 0.28 < 0.3$ , resulting in stopping.

At this point, we see that a cluster of Pennsylvania and Ohio rates differs significantly from the cluster of California and Kentucky rates. Due to such difference, one can use the membership of 0.9831 for Kentucky in its cluster as a new credibility rating  $Z$ , resulting in

$$C = 0.9831(200) + 0.0169\left(\frac{400}{3}\right) \approx 199.$$

Alternatively, one can propose to give the membership 0.9831 the meaning of credibility of the mean of Kentucky and California cluster, thus producing a new mean:

$$C = 0.9831\left(\frac{200 + 175}{2}\right) + 0.0169\left(\frac{100 + 125}{2}\right) \approx 186.$$

We believe this procedure, being a natural extension of the meaning of cluster membership and a modification of classical credibility, to be a potentially significant new development in our understanding of actuarial credibility.

## CONCLUSIONS

Our paper provides a relatively simple idea for extending the fuzzy clustering methods to credibility theory models. Further empirical investigations are needed in order to determine which clustering algorithms are most appropriate for the purpose of credibility measurement.

## REFERENCES

Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, Plenum Press, 1981.

Bühlman, H., Experience rating and credibility, *The ASTIN Bulletin*, 4, 199-207, 1967.

Herzog, T. N., *An Introduction to Bayesian Credibility Theory and Related Topics*, Casualty Actuarial Society, in Part 4 Study Notes, 1992.

Hossack, I. B., Pollard, J. H. and Zehnirith, B., *Introductory Statistics with Applications in General Insurance*, Cambridge University Press, 1983.

Ostaszewski, K., *An Investigation into Possible Applications of Fuzzy Set Methods in Actuarial Science*, Society of Actuaries, 1992, (to appear).

# ESTIMATIONS OF EXPECTEDNESS AND POTENTIAL SURPRISE IN POSSIBILITY THEORY

Henri Prade  
I.R.I.T.-C.N.R.S.  
Université Paul Sabatier  
118 route de Narbonne  
31062 Toulouse Cedex, France

Ronald R. Yager  
Machine Intelligence Institute  
Iona College  
New Rochelle, N.Y. 10801, U.S.A.

**Abstract :** This note investigates how various ideas of "expectedness" can be captured in the framework of possibility theory. Particularly, we are interested in trying to introduce estimates of the kind of lack of surprise expressed by people when saying "I would not be surprised that..." before an event takes place, or by saying "I knew it" after its realization. In possibility theory, a possibility distribution is supposed to model the relative levels of possibility of mutually exclusive alternatives in a set, or equivalently, the alternatives are assumed to be rank-ordered according to their level of possibility to take place. Four basic set-functions associated with a possibility distribution, including standard possibility and necessity measures, are discussed from the point of view of what they estimate when applied to potential events. Extensions of these estimates based on the notions of Q-projection or OWA operators are proposed when only significant parts of the possibility distribution are retained in the evaluation. The case of partially-known possibility distributions is also considered. Some potential applications are outlined.

## 1 - Introduction

In case of incomplete knowledge, people facing a query like "is A (going to be) true ?", may answer it with a great variety of ways such that "I don't know", "it is not impossible", "it is quite possible", "I would not be surprised that A is true", "I am quite certain that A is true", etc., according to the actual state of knowledge and the query. Possibility theory (Zadeh, 1978) offers a framework for modelling uncertain or vague information by means of a possibility distribution. Such a distribution assesses the level of possibility of each possible value of a considered (single-valued) variable  $x$ , i.e. the elements of the domain of the variable  $x$  are rank-ordered according to their relative possibility on the scale  $[0,1]$ . Then a possibility measure  $\Pi$  is associated with the distribution, and  $\Pi(A)$  estimates the consistency of the available knowledge with the statement "A is true" (short for "x is in A is true"). A dual measure of necessity  $N$  estimates the certainty of A as the impossibility of "non A", namely  $N(A) = \text{Impos}(\bar{A}) = 1 - \Pi(\bar{A})$ . Then  $N(\text{non } A)$ , the certainty that A is false, can be interpreted as a degree of surprise  $S(A) = N(\text{non } A) = \text{Impos}(A)$  that A is true. This corresponds exactly to the view developed by the English economist Shackle (1961) who worked out a non-probabilistic model of expectation, before the introduction of possibility theory. However this notion of surprise where  $\Pi(A) = 1 - S(A)$  does not seem to correspond exactly to the intended meaning of a sentence such that "I would not be surprised that A is true", which rather expresses that "A true" is more than just possible (even with a high degree), and is not far to be somewhat certain; what is stated is a very strong kind of possibility.

In this note we investigate what estimates can be defined from a possibility distribution-based knowledge representation, in order to evaluate, in various ways, how much an event, such that "x is in A", is expected to be true. The next section introduces four basic set functions defined from a possibility distribution, which are then extended using the notions of OWA operators, or of Q-projection, and also in the case of partially-defined possibility distributions.

## 2 - The Four Basic Set Functions in Possibility Theory

Let  $U$  be the domain of a single-valued variable  $x$ . In this note,  $U$  is supposed to be finite for simplicity. A possibility distribution  $\pi_x$  on  $U$  is a function from  $U$  to  $[0,1]$  which constrains the possible values of  $x$  according to the available information ;  $\pi_x(u) = 0$  means that  $x = u$  is definitely impossible while  $\pi_x(u) = 1$  means that absolutely nothing prevents that  $x = u$ . A possibility distribution  $\pi_x$  is said to be *normalized* iff  $\exists u_0 \in U$ ,  $\pi_x(u_0) = 1$ , i.e. at least one value of  $x$  in  $U$  is completely possible, which is natural if  $U$  is an exhaustive domain

for  $x$ .  $\pi_x$  can be viewed as a simple way of encoding a preference relation among the possible values of the variable  $x$ ; the smaller  $\pi_x(u)$ , the more unexpected  $x = u$  (or the less feasible  $x = u$ ). It is assumed in the following that  $\pi_x$  is normalized.

Given a possibility distribution  $\pi_x$  and an event  $A$ , four basic estimates can be imagined which are in agreement with the ordinal nature of  $\pi_x$ ; namely

- the *possibility measure* (Zadeh, 1978)

$$\Pi_x(A) = \max_{u \in A} \pi_x(u); \quad (1)$$

- the *guaranteed possibility* (Dubois and Prade, 1992)

$$\Delta_x(A) = \min_{u \in A} \pi_x(u); \quad (2)$$

and the similar evaluations for "non  $A$ ", denoted  $\bar{A}$ , whose complements to 1 are taken in order to define meaningful quantities for  $A$  ( $\pi_x$  should be normalized), namely

- the *necessity measure*

$$N_x(A) = \min_{u \in A} (1 - \pi_x(u)) = 1 - \Pi_x(\bar{A}); \quad (3)$$

- the *unguaranteed necessity*

$$\nabla_x(A) = \max_{u \in A} (1 - \pi_x(u)) = 1 - \Delta_x(\bar{A}). \quad (4)$$

$\Pi_x(A)$  estimates to what extent *there exists* a value  $u$  in  $A$  which is possible for  $x$ , i.e. the consistency of the proposition "x is in  $A$ " with what is not unexpected according to the available information.

$\Delta_x(A)$  estimates to what extent *all* the values in  $A$  are actually possible for  $x$  according to what is known; any value in  $A$  is at least possible for  $x$  at the degree  $\Delta_x(A)$ ; so  $\Delta_x(A)$  expresses a *guaranteed* possibility since it is a minimum level over  $A$ .

$N_x(A)$  estimates to what extent *all* the values in  $\bar{A}$  are impossible for  $x$ , or equivalently to what extent the value of  $x$  is necessarily in  $A$ ; any value in  $\bar{A}$  is at most possible for  $x$  at the degree  $1 - N_x(A)$ .

$\nabla_x(A)$  estimates to what extent *there exists* a value  $u$  in  $\bar{A}$  which is impossible for  $x$ . It is a measure of *unguaranteed* necessity in favor of  $A$  since we check the impossibility for  $x$  of only one value in  $\bar{A}$ , and not the impossibility of all.

Clearly

$$\Delta_x(A) \leq \Pi_x(A) \quad (5)$$

$$N_x(A) \leq \nabla_x(A). \quad (6)$$

Provided that  $\pi_x$  is normalized, and that  $\exists u \in U, \pi_x(u) = 0$  (at the technical level, it is always possible to add an extra-element to  $U$ , if necessary, in order to satisfy this requirement), we have the stronger inequality (Dubois and Prade, 1992)

$$\max(N_x(A), \Delta_x(A)) \leq \min(\Pi_x(A), \nabla_x(A)). \quad (7)$$

Thus  $\Delta_x$  corresponds to a very strong possibility and  $\nabla_x$  to a very weak necessity. Noticeably,  $N_x$  and  $\Delta_x$  are completely unrelated, as well as  $\Pi_x$  and  $\nabla_x$ . When estimating the tendency of  $A$  to contain the true value of  $x$ , we have indeed two complementary points of view, the extent to which values in  $A$  are effectively possible, and the extent to which values out of  $A$  are impossible. These two complementary evaluations may contribute to estimate our lack of surprise to have  $A$  true.

The four measures enjoy the following characteristic properties (the subscript  $x$  is omitted in the following)

$$\Pi(A \cup B) = \max(\Pi(A), \Pi(B)); \quad (8)$$

$$\Delta(A \cup B) = \min(\Delta(A), \Delta(B)); \quad (9)$$

$$N(A \cap B) = \min(N(A), N(B)); \quad (10)$$

$$\nabla(A \cap B) = \max(\nabla(A), \nabla(B)). \quad (11)$$

Thus  $\Pi$  and  $N$  are monotonically increasing with respect to set inclusion, while  $\Delta$  and  $\nabla$  are decreasing.

The interval  $[\Delta(A), \Pi(A)]$  characterizes the amplitude of the variation of the levels of possibility among the values in  $A$ , the interval  $[N(A), \nabla(A)] = 1 - [\Delta(\bar{A}), \Pi(\bar{A})]$  the amplitude of the variations in  $\bar{A}$ . We can then symbolically write  $[N(A), \nabla(A)] = [N, \nabla](A)$  and  $[\Delta(A), \Pi(A)] = [\Delta, \Pi](A)$ ; then we have

$$[N, \nabla](A) = 1 - [\Delta, \Pi](\bar{A}) \quad (12)$$

and (8)-(9)-(10)-(11) become

$$\begin{aligned} [\Delta, \Pi](A \cup B) &= mM([\Delta, \Pi](A), [\Delta, \Pi](B)) & (13) \\ [N, \nabla](A \cap B) &= mM([N, \nabla](A), [N, \nabla](B)) & (14) \end{aligned}$$

with  $mM([a,b],[c,d]) = [\min(a,c), \max(b,d)]$ , and then  $1 - mM([a,b],[c,d]) = mM(1 - [a,b], 1 - [c,d])$ . Note that  $mM([a,b],[c,d]) = \text{convex\_hull}([a,b] \cup [c,d])$ .

We now discuss what measures of expectedness and surprise are, and we introduce generalizations of the set functions  $\Pi, \Delta, N, \nabla$  based on the notions of OWA operators, or of Q-projection, in order to build intermediary estimates which may be used as estimates of how much an event  $A$  is expected to be true.

### 3 - Measures of Expectedness and Surprise

In this section we shall introduce some formal mechanism for capturing the concepts of "expectedness" and "surprise" associated with a set, based upon the assumption of some possibility distribution.

Assume we are concerned about John's height. Then a possibility distribution would be induced by the knowledge that *John is "tall"*. In this situation if it was found that John's height is six-feet seven inches one would *not be surprised* and would even have *expected* an answer like that. We shall in the following suggest some formal methods for capturing a measure of these concepts.

Assume we have a variable  $x$  which induces a possibility distribution  $\pi_x$  on  $U$ . Let  $A$  be a crisp subset of  $U$ . We shall let  $\text{Exp}(A)$  measure the degree of *expectedness* of  $A$  based upon  $\pi$ . We shall define this measure as the truth of the proposition

"most of the elements not in  $A$  are not possible".

We can more formally express this partial inclusion of  $\bar{A}$  into the fuzzy set of values of  $U$  which are rather impossible, as

$$\text{Exp}(A) = \text{most}_{u \in \bar{A}} [1 - \pi(u)]$$

where 'most<sub>u</sub>' refers to the proportion of elements in  $\bar{A}$  whose degree of possibility should be low. In this section and in the next one, we shall propose two slightly different ways of precisely defining this formal expression, either using OWA operators or Q-projections. Let us first consider a special extreme case of 'most': "all". In this case

$$\text{Exp}(A) = \min_{u \in \bar{A}} [1 - \pi(u)].$$

Thus this extreme definition becomes what we previously called the necessity measure. Thus the extreme of expectedness is necessity.

In order to evaluate expectedness in the general case, we can use the concept of OWA operators introduced by Yager (1988), i.e.

$$\text{Exp}(A) = \text{OWA}_{u \in \bar{A}} [1 - \pi(u)].$$

Let  $\bar{A} = \{u_1, \dots, u_r\}$ . Let  $a_i = 1 - \pi(u_i)$ . Let  $\{\omega_1, \dots, \omega_r\}$  be a set of weights such that

- 1)  $\forall i, \omega_i \in [0,1]$  ;
- 2)  $\sum_i \omega_i = 1$  ;

then  $\text{OWA}(a_1, \dots, a_r) = \sum_i \omega_i \cdot b_i$ , where  $b_i$  is the  $i$ th largest of the  $a_j$ . Two extreme cases of weights are worth noting. Taken  $\omega_r = 1$  (and then all others are zero), we get :

$$\text{OWA}(a_1, \dots, a_r) = b_r = \min_i a_i = \min_{u \in \bar{A}} [1 - \pi(u)],$$

i.e. the necessity measure of A. When  $\omega_1 = 1$  (and then all others are zero), we get

$$\text{OWA}(a_1, \dots, a_r) = b_1 = \max_i a_i = \max_{u \in \bar{A}} [1 - \pi(u)],$$

which is what we previously called the *unguaranteed necessity*. When  $\omega_i = 1/k$  for  $i = 1, k$  with  $k \leq r$ , we compute the average of the  $k$  largest levels of impossibility  $a_i = 1 - \pi(u_i)$ . Following Yager (1988)'s discussion, we can express "most" by an appropriate selection of weights.

We need now introduce a formal definition for "surprise of A" given a possibility distribution. We denote  $\text{Sur}(A)$  as the measure of surprise and define it as the truth of the proposition

"most of the elements in A are not possible".

We can more formally express this as

$$\text{Sur}(A) = \text{most}_{u \in A} [1 - \pi(u)].$$

As we can see we have  $\text{Sur}(A) = \text{Exp}(\bar{A})$ , which expresses that A is surprising if non-A is expected. However we do *not* have  $\text{Sur}(A) = 1 - \text{Exp}(A)$  in the same time (i.e. "A is surprising" is different from "A is unexpected" in our model). Clearly these two understandings of  $\text{Sur}(A)$  would be equivalent in a probabilistic model. Again considering the special case where "most" is replaced by "all" we get

$$\text{Sur}(A) = \min_{u \in A} [1 - \pi(u)].$$

This special case can be further simplified so that

$$\text{Sur}(A) = 1 - \max_{u \in A} \pi(u) = 1 - \Pi(A)$$

which corresponds to Shackle (1961)'s definition.

Considering the more general case of surprise (with "most" in place of "all"), we can use OWA operators to implement the formal expression by appropriate selection of the weights. At the extreme when  $\omega_s = 1$ , with  $A = \{u_{r+1}, \dots, u_s\}$ , we get  $\text{Sur}(A) = \min_u (1 - \pi(u)) = 1 - \Pi(A)$ , while when  $\omega_{r+1} = 1$

$$\text{Sur}(A) = \max_u (1 - \pi(u)) = 1 - \min_{u \in A} \pi(u) = 1 - \Delta(A).$$

We can further observe that if one considers the *negation* of "most" as "at\_least\_a\_few", then

$$\text{Sur}(A) = 1 - \text{at\_least\_a\_few}_{u \in A} [\pi(u)]$$

where *at\_least\_a\_few* corresponds to an ordered weighted average OWA' related to the one defining  $\text{Sur}(A) = \text{OWA}_{u \in A} [1 - \pi(u)]$  in the following way.  $\text{Sur}(A) = \text{OWA}(1 - \pi(u_{r+1}), \dots, 1 - \pi(u_s)) = \sum_i \omega_i \cdot b'_i$  where  $b'_i$  is the  $i$ th largest of the  $1 - \pi(u_j)$ . Then  $\text{Sur}(A) = \sum_i \omega_i - \sum_i \omega_i (1 - b'_i) = 1 - \text{OWA}'(\pi(u_{r+1}), \dots, \pi(u_s)) = 1 - \sum_i \omega_i \cdot c_i$  where  $c_i$  is the  $i$ th *smallest* of the  $\pi(u_j)$ .

**Remark : Extension to belief structures**

We shall here briefly suggest the extension of the preceding ideas to the case in which our basic knowledge is a belief structure of the type introduced by Shafer (1976). Assume we have a belief structure consisting of the focal elements  $B_1, \dots, B_n$  with weights  $m(B_i)$  (and  $\sum_i m(B_i) = 1$ ). We can define the degree of *amazement* associated with the subset A as

$$\text{Amaze}(A) = \sum_i \text{Sur}(A | B_i) \cdot m(B_i)$$

where

$$\text{Sur}(A | B_i) = \text{most}_{u \in A} [1 - \mu_{B_i}(u)]$$

and  $\mu_{B_i}$  is the characteristic function of  $B_i$ . We can define the degree of *anticipation* associated with  $A$  as

$$\text{Anticipate}(A) = \sum_i \text{Exp}(A | B_i) \cdot m(B_i)$$

where

$$\text{Exp}(A | B_i) = \text{most}_{u \in \bar{A}} [1 - \mu_{B_i}(u)].$$

#### 4 - Generalizations Based on Q-Projection

As established in the preceding sections, we have shown that  $\Pi, \Delta$  (resp.  $\cdot, \nabla$ ) are closely related to the concepts of surprise and expectedness. These concepts actually being related to the extremes of these measures. Crucial to the determination of the measures of surprise and expectedness are evaluations based upon quantifiers such as "most" lying being the extremes "for all" and "there exists" (corresponding to min and max operations). In the previous section we have suggested the use of OWA operators to implement these soft quantifiers. In this section we shall suggest an alternative approach to the kinds of evaluations necessary. This approach is based upon the notion of Q-projection (Yager, 1985). We only consider the case of non-fuzzy quantifiers where  $Q$  is a quantifier of the type "at least  $r/k$ " for simplicity. We define each Q-projection in terms of a median operator, which has some notation advantage for expressing Q-projection. In the following we shall first define the concept of Q-possibility of  $A$ . In the ordinary measure of possibility we have  $Q = \text{"at least one"}$  (thus  $A$  is possible if just one element in  $A$  is possible), while for Q-possibility measure of the type discussed here, we have  $Q = \text{"at least } r/k\text{"}$ , where  $k$  is the cardinality of the set  $A$ . In this more general setting we are saying that  $A$  is Q-possible if at least  $r/k$  of the elements in  $A$  are possible. We further note that if we define "most" by the appropriate selection of some value  $r$  as explained below, we have

$$\text{Sur}(A) = 1 - Q\text{-possibility}(A).$$

Let  $A = \{u_1, \dots, u_k\}$  be the finite subset of  $U$  on which we want to estimate to what extent a given number (or a given proportion) of values of  $A$  are possible. This number or proportion can be translated into a  $k$ -tuple of the form  $Q = (1, \dots, 1, 0, \dots, 0)$  where  $k = |A|$ , and where the number of '1' in the tuple representing  $Q$  is  $r$ . Then the Q-possibility of  $A$ , denoted by  $Q(A)$ , is defined by

$$Q(A) = \text{median}(\{\pi(u_1), \dots, \pi(u_k)\} \cup \bar{Q} \cup \{1\}) \quad (15)$$

where  $\bar{Q}$  denotes the complement of  $Q$ . Indeed,  $Q(A)$  is obtained as the median of a set of  $2k + 1$  elements made of  $k - r + 1$  elements equal to '1', of the  $k$  values  $\pi(u_1), \dots, \pi(u_k)$ , and of  $r$  values equal to 0. Thus,  $Q(A)$  is equal to the  $(k + 1)$ th value when the  $2k + 1$  elements are ranked in decreasing order, i.e. the  $r$ th value in the set  $\{\pi(u_1), \dots, \pi(u_k)\}$ , once these degrees are decreasingly ordered. Clearly  $Q = (1, 0, \dots, 0)$  (with  $(k - 1)$  '0') gives back  $Q(A) = \Pi(A)$ , while  $Q = (1, 1, \dots, 1)$  (with  $k$  '1') yields  $Q(A) = \Delta(A)$ . Clearly, in any case

$$Q(A) \in [\Delta, \Pi](A). \quad (16)$$

It can be shown (see Prade (1990) for instance) that the Q-possibility of  $A$  is nothing but the possibility measure that the number of *possible* elements (according to  $\pi$ ) is at least  $r$ , computed from the possibility distribution representing the more or less possible values of the cardinality of the fuzzy subset of  $A$  made of the elements which are rather *possible*.

By duality, quantities of the form  $1 - Q'(\bar{A})$  can be introduced. We have

$$\begin{aligned} 1 - Q'(\bar{A}) &= 1 - \text{median}(\{\pi(u'_1), \dots, \pi(u'_k)\} \cup \bar{Q}' \cup \{1\}) \\ &= \text{median}(\{1 - \pi(u'_1), \dots, 1 - \pi(u'_k)\} \cup Q' \cup \{0\}) \end{aligned} \quad (17)$$

where  $\bar{A} = \{u'_1, u'_2, \dots, u'_k\}$ ,  $k' = |\bar{A}|$ , and  $Q'$  is a  $k'$ -tuple of '1' and '0'. When  $Q' = (1, 0, \dots, 0)$ , we recover  $N(A) = 1 - Q'(\bar{A})$ , and when  $Q' = (1, 1, \dots, 1)$ , we get  $\nabla(A) = 1 - Q'(\bar{A})$ . When "most" of the values in  $A$  are highly possible, or when only few values outside  $A$  are possible (i.e. equivalently, "most" of the values in  $\bar{A}$  are impossible), which can be estimated using respectively  $Q(A)$  (with  $r$  "close" to  $k$ ), and  $1 - Q'(\bar{A})$  where  $Q'$  models "few" (the number of '1' in  $Q'$  is small), we may consider that this is the kind of situation where we would expect

that A is true. Unfortunately, Q does not enjoy a decomposability property with respect to the union of subsets in the general case, as  $\Pi$  and  $\Delta$  do.

We may conclude that A *should* be true either by checking, on a completely known possibility distribution, that (at least) most values outside A are impossible for instance, or from the computation of approximations of  $[\Delta, \Pi](A)$  and  $[N, \nabla](A)$  on the basis of a partially-known possibility distribution, as explained below.

## 5 - Estimations Based on Partially-Known Possibility Distributions

By a partially known possibility distribution, we mean that for each element u of U, the degree of possibility  $\pi_x(u)$  is only known to belong to an interval  $[\pi_x^-(u), \pi_x^+(u)]$ . The upper bound  $\pi_x^+$  is normalized on U since  $\pi_x$  is supposed to be normalized, while  $\pi_x^-$  is not necessarily normalized.

Then, the following bounds can be computed

$$\Pi^+(A) = \max_{u \in A} \pi_x^+(u) \geq \Pi(A) \geq \max_{u \in A} \pi_x^-(u) = \Pi^-(A) \quad (18)$$

$$N^+(A) = \min_{u \notin A} (1 - \pi_x^+(u)) \leq N(A) \leq \min_{u \notin A} (1 - \pi_x^-(u)) = N^-(A) \quad (19)$$

$$\Delta^+(A) = \min_{u \in A} \pi_x^+(u) \geq \Delta(A) \geq \min_{u \in A} \pi_x^-(u) = \Delta^-(A) \quad (20)$$

$$\nabla^+(A) = \max_{u \notin A} (1 - \pi_x^+(u)) \leq \nabla(A) \leq \max_{u \notin A} (1 - \pi_x^-(u)) = \nabla^-(A). \quad (21)$$

In other words we have inner and outer approximations of  $[\Delta, \Pi]$  and  $[N, \nabla]$ , namely

$$\forall A, [\Delta^+, \Pi^-](A) \subseteq [\Delta, \Pi](A) \subseteq [\Delta^-, \Pi^+](A) \quad (22)$$

$$\forall A, [N^-, \nabla^+](A) \subseteq [N, \nabla](A) \subseteq [N^+, \nabla^-](A) \quad (23)$$

However  $[\Delta^+, \Pi^-](A)$  may be empty if it happens that  $\Delta^+(A) > \Pi^-(A)$ , as well as  $[N^-, \nabla^+](A)$  if  $N^-(A) > \nabla^+(A)$ . A particular case which is worth considering is when  $\exists V \subseteq U, \forall u \in V, \pi_x^-(u) = \pi_x^+(u) = \pi_x(u)$  and  $\forall u \in U - V, \pi_x^-(u) = 0, \pi_x^+(u) = 1$ , i.e.  $\pi_x$  is perfectly known on a part of U and completely unknown elsewhere. Then the lower bound of  $N(A)$

$$N^-(A) = \min_{u \in \bar{A} \cap V} (1 - \pi_x(u)) = N(A \cup \bar{V}) \geq N(A) \quad (24)$$

(while  $N^+(A) = 0$  as soon as  $V \not\supseteq \bar{A}$ ) is a good candidate for estimating a beginning of certainty in favor of A. Indeed,  $N^-(A) = N(A \cup \bar{V})$  corresponds to the certainty in favor of a set less specific than A, but which contains A. Note that  $N(A) \geq N(A \cap V) = \min(N(A \cup \bar{V}), N(V))$  where  $N(V)$  is totally unknown, since  $\pi_x$  is only supposed to be known on V. Then  $N^-(A) = N(A \cup \bar{V})$  is a good approximation of the certainty of A with respect to the available information. Moreover if, together with  $N^-(A) > 0$ ,  $\Delta^+(A) = \min_{u \in A \cap V} \pi_x(u) = \Delta(A \cap V) \leq \Pi^-(A) = \max_{u \in A \cap V} \pi_x(u) = \Pi(A \cap V) \leq \Pi(A)$  is large enough, we would not be surprised that A turns to be true.

## 6 - Potential Applications

Although this note is basically oriented towards the formalization of the concepts of expectedness and surprise in the framework of possibility theory, let us briefly outline some potential applications.

A first use we may think of is the representation of decision rules of the kind "if A is expected then do..." which is a soft and more realistic version of the rule "if A is certain then do..."

Another use might be in information systems where we want to rank the items according to what extent they can be expected to satisfy the request. This might be of interest particularly if the set of items which more or less certainly satisfy the request is empty and the set of items which satisfy it only possibly is too large.

Clearly, it is not only important to be able to represent incomplete, uncertain, vague states of knowledge, but also to understand what a model offers for modelling expectation. This may be important in knowledge-based systems for instance for representing the state of knowledge of the user and deciding what explanation has to be provided to him/her (usually what is expected has not to be explained and what is surprising has to be explained).

Another issue that our future work in this area will focus on is possibility distribution generation based upon surprise and expectedness qualification. Consider a proposition like "I expect (at degree  $\alpha$ ) that John will be late". This proposition can be seen to induce a possibility distributions  $\pi$ , over John's arrival time. In particular we see that this requires the solution of an equation of the type  $\alpha = \text{Exp}[A / \pi]$ . In a similar fashion propositions like "I would be surprised (at the degree  $\alpha$ ) if John is early" or "I would not be surprised (at the degree  $\beta$ ) if John is late" can be seen to induce possibility distributions. The ability to generate possibility distributions from propositions of the above type would provide an interesting tool in knowledge representation.

## 7 - Concluding Remarks

In this note we have investigated all the estimates which can be attached to a non-fuzzy event A, when the available knowledge is modelled by a possibility distribution (even if this distribution is partially specified). The role of four basic measures has been emphasized, two of them define an interval related to the estimation of the idea of possibility, while the two others define another interval related to the idea of necessity or certainty. The characteristic properties of these intervals have been laid bare. Other quantities, which generalize the previous ones in various ways, have been introduced. The appropriateness of these different degrees for estimating how much an event can be expected to be true, how much its occurrence is not surprising, has been discussed. All these measures could be extended to fuzzy events A.

## References

- Dubois D., Prade H. (1992) Possibility theory as a basis for preference propagation in automated reasoning. Proc. of the 1st IEEE Inter. Conf. on Fuzzy Systems (FUZZ-IEEE'92), San Diego, CA, March 8-12, 821-832.
- Prade H. (1990) A two-layer fuzzy pattern matching procedure for the evaluation of conditions involving vague quantifiers. J. of Intelligent and Robotic Systems, 3, 93-101.
- Shackle G.L.S. (1961) Decision, Order and Time in Human Affairs. (2nd edition) Cambridge University Press.
- Shafer G. (1976) A Mathematical Theory of Evidence. Princeton University Press, Princeton, N.J..
- Yager R.R. (1985) Q-projections of possibility distributions. IEEE Trans. on Systems, Man and Cybernetics, 15, 775-777.
- Yager R.R. (1988) On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. on Systems, Man and Cybernetics, 18, 183-190.
- Zadeh L.A. (1978) Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems, 1, 3-28.

S3-67

161803

p. 5

## Comparison of Specificity and Information for Fuzzy Domains

Arthur Ramer

Computer Science, University of New South Wales  
PO Box 1, Kensington, NSW 2033, Australia

May 1, 1992

### 1 Overview

Given a *universe of discourse*  $X$ —a domain of possible outcomes—an experiment may consist of selecting one of its elements, subject to operation of chance, or of observing the elements, subject to imprecision.

A priori uncertainty about the actual result of the experiment may be quantified, representing either the likelihood of the choice of  $x \in X$  or the degree to which any such  $x \in X$  would be suitable as a description of the outcome. The former case corresponds to probability distribution, while the latter gives a possibility assignment on  $X$ .

Study of such assignments and their properties comes under the purview of *possibility theory* [1]. It, like probability theory, assigns values in between 0 and 1 to express likelihoods of outcomes. Here, however, similarity ends. Possibility theory uses *maximum* and *minimum* functions to combine uncertainty, where probability theory uses plus and times operations. This leads to a very dissimilar theory in its analytical framework, even though they share several semantic concepts.

One of them consists of expressing quantitatively the *uncertainty* associated with a given distribution [2, 3]. Its value corresponds to the gain of *information* that would result from conducting an experiment and ascertaining its actual result. This gain becomes simultaneously a decrease in *uncertainty* about the outcome of an experiment.

The other concept we consider in depth is one of *specificity*. Although it has been introduced previously in a few different forms, a closer analysis shows that they share main epistemic features. We follow here the presentation of Ramer and Yager [10].

Fuzzy set  $(X, \mathbf{p})$  can be considered as a form of a likelihood function, with the elements of  $X$  where  $\mathbf{p}$  reaches its maximum playing privileged role. When selecting  $x : \mathbf{p}(x_0) = \max \mathbf{p}(x)$  is important to ask how definite has been such decision, and whether another element would offer a close choice.

In this interpretation, *specificity* becomes an attribute of the complete set of possibilities, the attribute assuming either numeric or linguistic values. Here we develop a comprehensive model of such specificity, expressed as a *numerical function* of a possibility assignment.

## 2 Introduction

This paper demonstrates how an integrated theory can be built on the foundation of possibility theory. Information and uncertainty were considered in 'fuzzy' literature since 1982. Our departing point is the model proposed by Klir [4, 5] for the discrete case. It was elaborated axiomatically by Ramer [9], who also introduced the continuous model [7].

Specificity as a numerical function was considered mostly within Dempster-Shafer evidence theory. An explicit definition was given first by Yager [11], who has also introduced it in the context of possibility theory [12]. Axiomatic approach and the continuous model have been developed very recently by Ramer and Yager [10]. They also establish a close analytical correspondence between specificity and information.

In literature to date, specificity and uncertainty are defined only for the discrete finite domains, with a sole exception of [10]. Our presentation removes these limitations. We define specificity measures for *arbitrary measurable* domains. When discrete, they can be finite or infinite or, in general have  $\mu(X) < \infty$  or  $\mu(X) = \infty$ . prespecified pattern. By abuse of the language we refer to this model as a *continuous* one.

We adopt the convention of avoiding, whenever possible, subscripts and indices. We do not specify explicitly basis of logarithms, as its change would simply amount to a multiplying all expressions by the same constant. Following tradition, binary logarithms— $\log_2$ —are assumed for the discrete distributions, and natural— $\ln$ — for the continuous cases. We use  $(\tilde{p})$  for the decreasing rearrangement of the sequence  $(p_i)$ . For finite sequences, rearrangements are permutations of their elements. For infinite sequences and functions we construct rearrangements using *cuts*. To define  $\tilde{f}$ , given  $f$  on  $X$ , we want all their  $\alpha$ -cuts to be of the same measure. We put

$$P(y) = \mu(\{x : f(x) \geq y\}),$$

$$\tilde{f}(x) = P^{-1}(x).$$

Now for the discrete rearrangements we associate with the sequence  $(p) = (p_1, \dots, p_n, \dots)$  a step function  $f : x \mapsto p_{[x]}$ , where  $[x]$  denotes the greatest integer no less than  $x$ . Then the descending rearrangement  $\tilde{f}$  corresponds to  $(\tilde{p})$ .

### 3 Information and uncertainty

We use the model of possibility theory introduced by Zadeh [13]. We view mapping  $\mathbf{p}$  as assigning a degree of assurance or certainty that an element of  $X$  is the outcome of an experiment. A priori we know only the distribution  $\mathbf{p}$ ; to determine  $x \in X$  means to remove uncertainty about the result, thus entailing a gain of information. We would be particularly interested in quantifying that gain of information, which would also express the uncertainty inherent in the complete distribution  $\mathbf{p}$ .

Following established principles of information theory [3], we stipulate that such information function satisfies certain standard properties. For  $\mathbf{p}_1$  on  $X$  and  $\mathbf{p}_2$  on  $Y$  we define a *noninteracting, joint* distribution  $\mathbf{p}_1 \otimes \mathbf{p}_2$  on  $X \times Y$  as

$$\mathbf{p} \otimes \mathbf{p}_2 : (x, y) \mapsto \min(\mathbf{p}_1(x), \mathbf{p}_2(y)).$$

If  $\mathbf{p}$  was already defined on a product domain  $X \times Y$ , we construct its *projections* (marginal distributions) using *maximum* operation

$$\mathbf{p}' : x \mapsto \max_y \mathbf{p}(x, y), \quad \mathbf{p}'' : y \mapsto \max_x \mathbf{p}(x, y).$$

There is often a need to consider a given assignment  $\mathbf{p}$  as defined on on a larger domain, without, however, making any essential change to the possibility values it represents. We do so by defining  $\mathbf{p}^Y$  for  $Y \supset X$ , as agreeing with  $\mathbf{p}$  on the elements of  $X$ , and 0 otherwise. Lastly, the elements of the domain of discourse could be permuted; if  $s : X \rightarrow X$  is one-to-one, we define

$$s(\mathbf{p}) : x \mapsto \mathbf{p}(s(x)).$$

We now postulate [5]

<i>additivity</i>	$I(\mathbf{p}_1 \otimes \mathbf{p}_2) = I(\mathbf{p}_1) + I(\mathbf{p}_2)$
<i>subadditivity</i>	$I(\mathbf{p}) \leq I(\mathbf{p}') + I(\mathbf{p}'')$
<i>symmetry</i>	$I(s(\mathbf{p})) = I(\mathbf{p})$
<i>expansibility</i>	$I(\mathbf{p}^Y) = I(\mathbf{p})$

It turns out that these properties essentially characterize the admissible information functions [6, 9]. Subject to the normalization of parameters, for

the discrete case of  $X = \{x_1, \dots, x_n\}$

$$U(\mathbf{p}) = \sum (\tilde{p}_i - \tilde{p}_{i+1}) \log i$$

which can be also written using *finite differences* notation

$$U(\mathbf{p}) = \sum \tilde{p}_i \nabla \log i.$$

We observe that the distribution which carries the highest uncertainty value consists of assigning possibility 1 to all the events in  $X$ . It states that, a priori, every event is fully possible. This distribution, carrying no prior information, can be considered the *most uninformed* one.

We shall now extend previous definitions to arbitrary measurable domains [7]. To avoid technical complications, we consider only a typical case of the unit interval.

As a first step, the discrete formula  $U(x) = \sum p_i \nabla \log i$  suggests forming  $\int_0^1 \tilde{f}(x) d \ln x, = \int_0^1 \frac{\tilde{f}(x)}{x} dx$  as a candidate expression for the value of information. Unfortunately,  $\tilde{f}(x)$  is equal to 1 at 0, and the integral above diverges. A solution can be found through a technique (used also in probability) of information distance between a given distribution and the most 'uninformed' one—where U-uncertainty attains its maximum. Our final formula becomes

$$I(f) = \int_0^1 \frac{1 - \tilde{f}(x)}{x} dx.$$

This integral is well defined and avoids the annoying singularity at 0. It can be used for a very wide class of functions, including all polynomials.

## 4 Principles of specificity

The discussion will be conducted in terms of a discrete countable distribution ( $p_i$ ), with finite distributions viewed as the initial segments. Our objective is to capture formally the informal intuition about specificity. The main premise is the principle of *juxtaposition*:

$Sp(\mathbf{p})$  expresses the preference for a certain maximal  $p_0$  over any and all the remaining  $p_i$ .

Now let us consider how, having selected  $p_0 = \max(p)$ , its *informal specificity* is estimated. We look first for the next largest  $p_i$  and estimate how its presence diminishes the specificity. The process is then iterated in the order of decreasing values of  $p_i$ , every next value lowering the estimated specificity. We can picture it as a sequential process, its input the decreasing

rearrangement ( $\tilde{p}_i$ ). We may also surmise that, for a given  $i$ , the drop in specificity caused by  $\tilde{p}_i$  will not depend on the earlier inputs  $\tilde{p}_1, \dots, \tilde{p}_{i-1}$ . This assumption of *independent influence* is consistent with the *juxtaposition* interpretation of specificity.

Let us consider the effect of a uniform modification of ( $p$ ). For a scaling  $\alpha \mathbf{p} = (\alpha p_1, \dots, \alpha p_n, \dots)$ ,  $0 \leq \alpha \leq 1$ , we may assume that the relative specificities remain unchanged, while with a shift of values  $\mathbf{p} - \beta = (p_1 - \beta, \dots, p_n - \beta, \dots)$  no change should occur.

Last item considered will be the effect of offering yet another choice, identical in value to several choices already provided. The common perception of *specificity* is that the change due to such  $n$ -th choice will be ever less as  $n$  increases—a *diminishing return*. For its *relative* effect, we can postulate taking away the same proportion of the specificity still available. After all, we consider yet another *identical* choice; only we consider it at stage  $n$  and not sooner.

We can extract an analytical representation from the rules elaborated above. The result is a linear formula

$$Sp(\mathbf{p}) = \tilde{p}_1 - \sum_{i \geq 2} w_i \tilde{p}_i$$

with  $\sum_{i \geq 2} w_i = 1$ . From here we can conclude that  $\lim_{i \rightarrow \infty} w_i = 0$ , and  $1 > w_2 > w_3 > \dots$ , in agreement with the 'diminishing returns'.

We shall consider the linear form of  $Sp(\mathbf{p})$  as *general* specificity function. It is general enough to fit most applications and, if  $w_i$  are supplied, it offers a comparison scale among the distributions.

Coefficients  $w_i$  can be established precisely if we assume the rule of constant influence of equal choices. After more calculations

$$Sp(\mathbf{p}) = \tilde{p}_1 - \sum_{i \geq 2} (\omega^{i-1} - \omega^i) \tilde{p}_i$$

for some  $\omega$ ,  $0 < \omega < 1$ , producing a *definite* form of specificity. Choosing  $\omega = \frac{1}{2}$  (in spirit of binary logarithms) gives  $Sp(\mathbf{p}) = \tilde{p}_1 - \sum_{i \geq 2} \frac{\tilde{p}_i}{2^{i-1}}$ . In the above formulas the role of  $\tilde{p}_1$  is manifestly different from that of  $\tilde{p}_i$ ,  $i \geq 2$ . A more symmetric expression can be obtained defining  $W_i = 1 - w_2 - \dots - w_i$ , resulting in a *general expression*

$$Sp(\mathbf{p}) = \sum_{i \geq 1} W_i (\tilde{p}_i - \tilde{p}_{i+1})$$

and the *definite one*

$$Sp(\mathbf{p}) = \sum_{i \geq 1} \omega^{i-1} (\tilde{p}_i - \tilde{p}_{i-1}).$$

## 5 Specificity as information

Design of a specificity function can be also approached from the perspective of Dempster-Shafer theory. It is a very general framework for capturing numerically notions of evidence in support of assertions about the domain of discourse. The model we use applies to a finite domain of discourse  $X$ , where evidence  $m_i$  is assigned to the selected subsets  $A_i \subset X$ . We require that  $\sum m_i = 1$  and that the empty set is not included. For such structures several measures of *nonspecificity* have been proposed, among which

$$N(\mathbf{m}) = \sum m_i \log |A_i|$$

is usually preferred, being both additivity and subadditive

This model can be applied to fuzzy sets and possibility distributions. It results in a familiar

$$U(\mathbf{p}) = \sum (\bar{p}_i - \bar{p}_{i+1}) \log i.$$

We are interested in a *specificity* function, and an appropriate expression would be a complement of  $U(\mathbf{p})$  wrt the most nonspecific distribution  $1^{(n)} = (1, \dots, 1)$

$$I(\mathbf{p}) = U(1^{(n)}) - U(\mathbf{p}) = \log n - \sum (\bar{p}_i - \bar{p}_{i+1}) \log i.$$

For the continuous model we propose a two-part structure, depending on the measure of the domain of discourse.

If  $\mu(X)$  is finite we rearrange it to form  $\tilde{f}(x)$  on  $[0, \mu(X)]$ . Then we propose as the basic measure

$$I(f) = \int_0^{\mu(X)} \frac{1 - \tilde{f}(x)}{x} dx.$$

For  $X$  of infinite measure we propose using

$$Sp(f) = k \int_0^{\infty} \tilde{f}(x) e^{-kx} dx$$

or, in general

$$Sp(f) = \int_0^{\infty} \tilde{f}(x) W'(x) dx$$

for  $W(x)$ —a monotonically decreasing function satisfying

$$W(0) = 1, \quad \lim_{x \rightarrow \infty} W(x) = 0.$$

It can be derived from the *general discrete* form by a process similar to that which led to  $I(f)$ .

## References

- [1] DUBOIS, D. and PRADE, H., 1988, *Possibility theory*, Plenum Press, New York .
- [2] DUBOIS, D. and PRADE, H., 1987, Properties of measures of information in evidence and possibility theories, *Fuzzy Sets Syst.*, 24(2).
- [3] GUIASU, S., 1977, *Information Theory and Applications*, McGraw Hill, New York .
- [4] HIGASHI, M. and KLIR, G., 1983, On the notion of distance representing information closeness, *Int. J. Gen. Syst.*, 9(1).
- [5] HIGASHI, M. and KLIR, G., 1982, Measures of uncertainty and information based on possibility distributions *Int. J. Gen. Syst.*, 8(3).
- [6] KLIR, G. and MARIANO, M., 1987, On the uniqueness of possibilistic measure of uncertainty and information, *Fuzzy Sets Syst.*, 24(2).
- [7] RAMER, A., 1989, Concepts of fuzzy information measures on continuous domains, *Int. J. Gen. Syst.*, 17(2-3).
- [8] RAMER, A. Uniqueness of information measure in the theory of evidence, *Fuzzy Sets and Syst.* 24(2).
- [9] RAMER, A. and LANDER, L., 1987, Classification of possibilistic uncertainty and information functions, *Fuzzy Sets Syst.*, 24(2).
- [10] RAMER, A. and YAGER, R., 1992, Analysis of specificity of fuzzy sets, *FUZZ-IEEE'92*, San Diego, CA.
- [11] YAGER, R., 1983, Entropy and specificity in a mathematical theory of evidence, *Int. J. Gen. Syst.* 9.
- [12] YAGER, R., 1980, Aspects of possibilistic uncertainty, *Int. J. Man-Machine Studies*, 12.
- [13] ZADEH, L., 1978, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets Syst.*, 3(1).

THE AXIOMATIC DEFINITION OF A LINGUISTIC SCALE  
FUZZINESS DEGREE, ITS MAJOR PROPERTIES  
AND APPLICATIONS

161804  
p. 8

Alexander P. Ryjov

Abstract. Model of human estimate of real objects as measuring procedure in fuzzy linguistic scales (FLS) is being considered in the report. The definition of FLS fuzziness degree and its major properties is given in the report. Definitions of information losses and noise while user works with data base (or knowledge base), containing linguistic description of objects are being introduced and described, and proven, that this value gives linear connection with degree of fuzziness.

Key words: estimate of real object, fuzzy linguistic scales, degree of fuzziness, quality of information search.

INTRODUCTIONS

Model of human estimate of real objects as measuring procedure in fuzzy linguistic scales (FLS) /1/ is being considered in the report. While describing objects some human being can't use any measuring devices, he makes it in terms of some sensible properties, and he has some doubts while giving some value to a property.

If there are a lot of property's values the trouble of choice is that there are some of them, which are "just equally" suitable for the object description. And if there are little of values the trouble is that all of them are "just equally" unsuitable to describe some object.

General study object of this works is a set of scale's value of a linguistic scale /1/. Example of scale's value for linguistic scale "Height" is given an Fig.1.

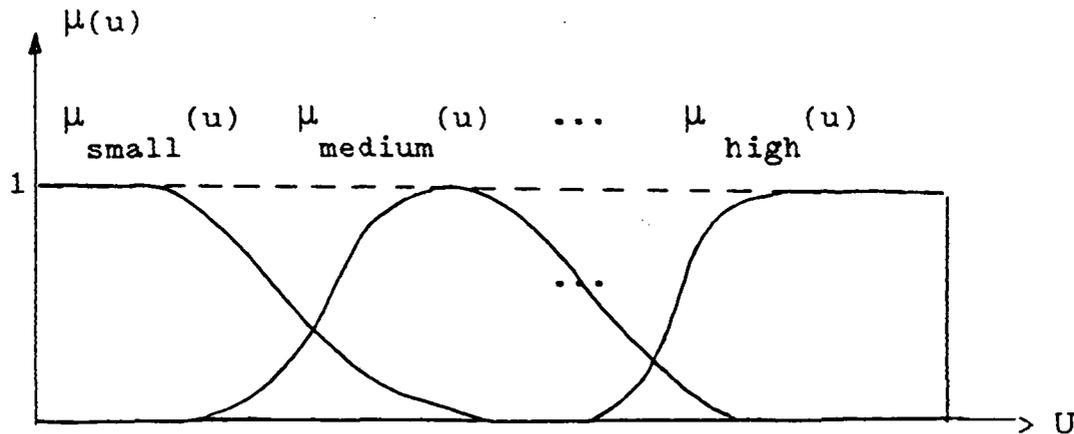


Fig.1

Such structures can be also interpreted as a set of different alternatives in problem solving and decision-making /2,3,4/ or a descriptions of classes in fuzzy classification and clustering /5,6/ or a representation of term-sets of linguistic values /7/ and etc. However, first interpretation (in the same way /8/) is the most preferable for application in information systems.

## 1. FLS FUZZINESS DEGREE: DEFINITION, EXAMPLE AND PROPERTY

The definition of FLS fuzziness degree is given in the paper under some matter-of-fact restrictions on membership function form, and the set of such functions, which create the FLS.

Let's assume, that membership functions for FLS  $l_t$  (where  $t$  - number of scale values) are defined on some segment  $U \in \mathbb{R}^1$  and meets following requirements:

- 1) normal /9/:  $\forall j (1 \leq j \leq t) \exists U_j^1 \neq \emptyset$ ,  
where  $U_j^1 = \{u \in U: \mu_j(u) = 1\}$ ,  $U_j^1$  are segment;
- 2) increasing from the left  $U_j^1$  and decreasing from the right  $U_j^1$ .

The requirements are quite natural for membership functions of notions gathered in some FLS's scale values set. Actually, the first means that there's at least one object for each scale value, which is typical or ideal for the notion; and the second may be interpreted as requirement of gradual changing of the notion limits.

Characteristic functions we'll be mentioned in the article. Let's assume, that :

- 3) those functions can have not more than two break points of second sort.

Let's assume that  $L$  is the set of functions satisfying requirements 1)- 3). The set  $L$  is a subset of a set of functions integral able on some measurable set of functions  $L_2$ , and therefore, a measure can be introduced on  $L$ . For example :

$$d(f, g) = \int_U |f(u) - g(u)| du, \quad f \in L, \quad g \in L.$$

Let's introduce some restrictions on a set of functions from  $L$ , which are creating a set value of FLS  $l_t$ . And let's assume that a set of such functions suit following requirements:

- 4) completeness:  $\forall u \in U \exists j (1 \leq j \leq t): \mu_j(u) \neq 0$

$$5) \text{ orthogonally: } \forall u \in U \quad \sum_{j=1}^t \mu_j(u) = 1$$

These restrictions are quite natural too. Assuming that 4) isn't true then a set  $U' = \{u \in U: \forall j (1 \leq j \leq t) \mu_j(u) = 0\}$  may be harmlessly deleted from  $U$ , therefore a set  $U \setminus U'$  may be considered instead of universum. That means that there's no scale values associated with any point from  $U'$  set, and scale has improper definition.

Restriction 5) was described in /2/. Scales built under 5) are not only useful for theoretical analysis, but they must be the most spread in use, because the restrictions mean that:

- used notions (scale values) are quite differ from each other;
- they do not describe the same objects.

Let's call a set of FLS with scale values under 4) and 5)  $G(L)$  - scales.

We can introduce a measure on  $G(L)$  too.

Lemma 1. Let's assume that  
 $\{\hat{\mu}_1(u), \hat{\mu}_2(u), \dots, \hat{\mu}_t(u)\}$  - a set of scale values  $\hat{l}_t$ ;  
 $\{\mu_1(u), \mu_2(u), \dots, \mu_t(u)\}$  - a set of scale values  $l_t$ ;  
 $d(f, g)$  - a measure in  $L$ .

Then  $\rho(l_t, \hat{l}_t) = \sum_{i=1}^t d(\mu_i, \hat{\mu}_i)$  - is a measure in  $G(L)$ .

To formulate axioms we should define a scale, which is based on some FLS and is "unfuzzy", meaning that the scale's value is a set of characteristic functions, produced with membership functions of FLS.

Thus, assuming that  $l_t \in G(L)$ , is a FLS defined on  $U$  and consisting of membership functions  $\mu_1(u), \dots, \mu_t(u)$ . Let's construct some "unfuzzy" set value  $\hat{l}_t$ .  $\hat{l}_t$  - is a set of characteristic functions  $h_1(u), \dots, h_t(u)$ , where

$$h_i(u) = \begin{cases} 1, & \text{if } \max_{1 \leq j \leq t} \mu_j(u) = \mu_i(u) \\ 0, & \text{otherwise} \end{cases}$$

Call  $\hat{l}_t$  - the nearest "unfuzzy" scale, based on FLS  $l_t \in G(L)$ .

Let's assume that fuzziness degree of FLS, whose scale values are defined upon universum  $U$ , is the value of functional  $\xi(l_t)$ , defined on the membership function scale values set and satisfying following axioms:

- A1.  $0 \leq \xi(l_t) \leq 1 \quad \forall l_t \in G(L)$ .  
 A2.  $\xi(l_t) = 0 \iff \forall u \in U \exists i (1 \leq i \leq t): \mu_i(u) = 1, \mu_j(u) = 0 \quad \forall j \neq i$ .  
 A3.  $\xi(l_t) = 1 \iff \forall u \in U \exists i_1, i_2 (1 \leq i_1, i_2 \leq t):$   

$$\mu_{i_1}(u) = \mu_{i_2}(u) = \max_{1 \leq j \leq t} \mu_j(u)$$

A4. Let's assume that FLS  $l_t$  and  $l'_t$  are defined on universes  $U$  and  $U'$  correspondingly;  $t$  and  $t'$  can be equal and not equal and not equal to each other.

$$\xi(l_t) \leq \xi(l'_t), \text{ if } \rho(l_t, \hat{l}_t) \leq \rho(l'_t, \hat{l}'_t),$$

where  $\rho(\cdot, \cdot)$  - some metric in  $G(L)$ .

Axiom A1 defines domain of values for functional  $\xi(l_t)$ , or fuzziness measuring borders.

Axioms A2 and A3 describes the scales where  $\xi(l_t)$  assumes minimal and maximal values, or maximal "unfuzzy" and maximal "fuzzy" scales correspondent.

Axiom A4 defines the fuzziness degree comparison rule for each pare scales. It may be expressed in such a way: the nearer given FLS to its nearest unfuzzy scale, the less it's fuzziness degree.

Let's give an answer for question of existence a functional satisfying those axioms.

Theorem 1. Assume that  $l_t \in G(L)$ . Then functional

$$\xi(l_t) = \frac{1}{|U|} \int_U f(\mu_{i_1^*}(u) - \mu_{i_2^*}(u)) du,$$

$$\text{here } \mu_{i_1^*}(u) = \max_{1 \leq j \leq t} \mu_j(u), \quad \mu_{i_2^*}(u) = \max_{\substack{1 \leq j \leq t \\ j \neq i_1^*}} \mu_j(u),$$

$f$  satisfies following requirement:

F1:  $f(0) = 1, f(1) = 0$ ;

F2:  $f$  decreases,

is fuzziness degree  $l_t$ , i.e. satisfies A1 - A4.

It's easy to prove, that the only linear function satisfying F1, F2 is a function  $f(x) = 1 - x$ .

A subset of polynomials of degree 2, satisfying F1, F2, can be described. Those are expressions of the following type:

$$f_a(x) = ax^2 - (1 + a)x + 1.$$

Subset of functions of other types (logarithmic, trigonometric etc.) satisfying conditions F1, F2 may be defined in a similar way. Let's use those functions in formula for  $\xi(l_t)$ , and get some functionals, satisfying A1 - A4, i.e. it is a fuzziness degree.

FLS fuzziness degree properties for linear  $f$  are being described in the report. In this case

$$\xi(l_t) = \frac{1}{|U|} \int_U (1 - (\mu_{i_1^*}(u) - \mu_{i_2^*}(u))) du, \quad (*)$$

here  $\mu_{i_1^*}(u) = \max_{1 \leq j \leq t} \mu_j(u)$ ,  $\mu_{i_2^*}(u) = \max_{\substack{1 \leq j \leq t \\ j \neq i_1^*}} \mu_j(u)$ ,

This fuzziness degree measurement functional was introduced at the first time in /10/ for the task of optimal quality properties values set choice in human-machine systems.

Let's define the following subset of function set  $L$ :

$\bar{L}$  - a set of functions from  $L$ , which are part-linear and linear on  $\tilde{U}$

$$\tilde{U} = \{u \in U: \forall j (1 \leq j \leq t) \ 0 < \mu_j(u) < 1\};$$

$L$  - a set of functions from  $L$ , which are part-linear on  $U$  (including  $\tilde{U}$ ).

Theorem 2. Let  $l_t \in G(\bar{L})$ . Then  $\xi(l_t) = \frac{d}{2|U|}$ , where

$$d = |U^*| = |\{u \in U: \forall j (1 \leq j \leq t) \ \mu_j(u) \neq 1\}|$$

Theorem 3. Let  $l_t \in G(L)$ . Then  $\xi(l_t) = C \frac{d}{|U|}$ , where

$C < 1$ ,  $C = \text{Const.}$

The fuzziness degree of a fuzzy set induced by  $\xi(l_t)$  is defined as fuzziness degree of a trivial FLS, determined with a fuzzy set  $\mu(u)$ :

$$\xi(\mu) = \frac{1}{|U|} \int_U (1 - |2\mu(u) - 1|) du$$

It's easy proved, that  $\xi(\mu)$  satisfies all the axioms for the set's fuzziness degree /11/. It may show that the introduced in the report more general notion  $\xi(l_t)$  had been correctly defined.

It's easy shown, that the functional may be considered as an average human doubts degree while describing some real object (situations) /4,12/.

## 2. APPLICATION FLS FUZZINESS DEGREE TO INFORMATION SEARCH

The results were published at the first time in /11/.

Definitions of information loses and noise while user works with data base, containing linguistic description of objects are being introduced and described in the report. While interacting with the system user formulates his query and gets an answer according to the search request. And if he knew real (not linguistic) values of object characteristics, he, possibly, would defeat some of displayed objects (noise) and he would add some others from data base (loses). Information noise and losses appear because of fuzziness of scale elements.

Because of volume restrictions and taking into account the illustrative character of the chapter we stop at the main results. In the next work we are going to describe the problems of formalization of fuzzy database information retrieval quality rations in complete.

Theorem 4. Assume that  $l_t \in G(\bar{L})$ ,  $\xi(l_t)$  - degree of fuzziness of  $l_t$ ;  $\Pi_x(U)$ ,  $H_x(U)$  - average information loses and noise, appearing during information search with search attribute value set  $X$ , equal to  $l_t$  -scale values set;  $U$  - universum  $l_t$ ;  $N(u)$  - number of objects, whose definitions are in a database and which having a real characteristic value equal to  $u$ , - is a constant. Furthermore, assume that all of property values are equally preferable for user, meaning that request probabilities for all the property values are equal. Then

$$\Pi_x(U) = H_x(U) = \frac{2N}{3t} \xi(l_t), \quad N = \text{Const.}$$

Theorem 5. Assume that  $l_t \in G(L)$ ,  $N(u) = N = \text{Const}$  and request probabilities for all the property values are equal. Then

$$\Pi_x(U) = H_x(U) = \frac{c}{t} \xi(\mu),$$

where  $c$  - a constant, which depends on  $N$  only.

Thus,  $\alpha\%$  - fuzziness degree decrease leads to the same decrease of average information loses and noise if the number of property values is constant. Simultaneous fuzziness degree decrease of properties values number lead to even more substantial decrease of information loses and noise.

The following method of property values set choosing for fuzzy databases, can be evaluated from the given results:

1. To generate all possible sets of property values.
2. To represent each of with FLS scale values set.
3. To evaluate the degree of fuzziness for each of the property values sets according to (\*).

4. Chose the set of property value set, which has the minimal ratio of fuzziness degree and number of elements. Your choice will provide the minimal information losses and noise of information retrieval using the property.

#### CONCLUSIONS

Some method to calculate the fuzziness degree of the combination of fuzzy sets (defined upon the same universum) has been given in the article. The axioms for such measure of uncertainty have been formulated, its interpretation has been given. The theorem of existence has been proven and some properties of fuzziness degree have been described.

The problems of using of the results in information applications (fuzzy retrieval systems) have been discussed. It is described that the fuzziness degree has linear dependence with the indicator of retrieval quality. Taking into account the result the method of choosing the optimal values has been suggested. Using the method some user may describe objects to achieve better results of finding information in fuzzy data bases. Under these circumstances a person - a source of information - would suffer minimal difficulties (uncertainties) to describe real objects.

The results may be used also in some tasks to construct knowledge bases, decision-making tasks under fuzzy conditions and pattern recognition.

#### References.

1. Decision-making models on the base of linguistic variable / Ed. by Borisov A.N. and etc. - Riga: Zinatne, 1982 (in Russian).
2. Fuzzy sets in models of control and artificial intelligence / Ed. by Pospelov D.A. - Moscow: Nauka, 1986 (in Russian).
3. Capocelli R., De Luca A. Fuzzy sets and decision theory // Information and Control. - 1973. - v.23. - p. 43 - 50.
4. Ryjov A. About a method of fuzzy object and situation optimal description in decision-making systems // International symposium on fuzzy approach to reasoning and decision-making. Abstracts. - Technical university Liptovsky Nikolach, 1990. - p. 81.
5. Bezdek J.S. Numerical taxonomy with fuzzy sets // Journal of Mathematical Biology. - 1974. - v.1. - p. 57 - 71.
6. De Luca A., Termini S. Measures of ambiguity in the analysis of complex systems // Lectures Notes on Computer Sciences. - 1977. - v. 53. - p. 58 - 73.
7. Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning, Part 1,2,3, // Inform.-Sci.8,199-249;8,301-357;9,43-80(1975).

8. Ryjov A.P. The degree of fuzziness of linguistic scale and its properties // Fuzzy sets in informatics: Abstracts of Moscow International Conference. - Moscow, September 20 - 23. - Moscow, 1988. - p. 57.

9. Kaufmann A. Introduction to the the theory of fuzzy subset, v.1 - N.Y.: Academic Press, 1975. - 643 p.

10. Ryjov A. About degree of fuzziness of fuzzy characteristics // VII All-Union conference on problem of theoretical cybernetics. September 18, 1985 - September 20, 1985. Abstracts. Part I - Irkutsk, 1985. - p. 53-55 (in Russian).

11. Ryjov A. The degree of fuzziness of fuzzy descriptions // Mathematical cybernetics and its application to biology / Ed. by Krushinsky L.V., Yablonsky S.V., Lupanov O.B. - Moscow, Moscow University Publishing, 1987. - p.60-77 (in Russian).

12. Ryjov A., Averkin A. The Axiomatic definition of a linguistic scale fuzziness degree and its general properties // II All-Union conference "Artificial intelligence - 90". Abstracts. Part I. - Minsk, 1990. - p.162-165 (in Russian).

HOW TO SELECT COMBINATION OPERATORS FOR  
FUZZY EXPERT SYSTEMS USING CRI

161805  
P-10

I.B. Turksen and Y. Tian

Dept. of Industrial Engineering  
University of Toronto  
Toronto, Canada  
M5S 1A4

**Abstract:** A method to select combination operators for fuzzy expert systems using Compositional Rule of Inference(CRI) is proposed from the consideration of basic requirement for fuzzy reasoning. First, fuzzy inference processes based on CRI are classified into three categories in terms of their inference results, i.e., the Expansion Type Inference, the Reduction Type Inference, and Other Type Inferences. Further, implication operators under Sup-T composition are classified as the Expansion Type Operator, the Reduction Type Operator, and the Other Type Operators. Finally combination of rules or their consequences is investigated for inference processes based on CRI. It is suggested that for inference processes using Sup-T composition in the context of CRI, the combination operator be "min" if the implication operator  $a \rightarrow b = F(a, b)$  is an Expansion Type and is an inversely proportional function of  $a$ , i.e., if  $a_1 \geq a_2$ , then  $F(a_1, b) \leq F(a_2, b)$ , and the combination operator be "max" if the implication operator  $F(a, b)$  is a Reduction Type and is a proportional function of  $a$ , i.e., if  $a_1 \geq a_2$ , then  $F(a_1, b) \geq F(a_2, b)$ .

**Keywords:** Compositional Rule of Inference, Inference Processes, Expansion, Reduction, Implication, Composition, Combination.

1. INTRODUCTION

Suppose there are  $\Omega$  fuzzy rules in the rule base of a fuzzy expert system as follows:

$$\begin{aligned}
 & \text{IF } X \text{ is } A_1 \text{ THEN } Y \text{ is } B_1 \\
 & \text{IF } X \text{ is } A_2 \text{ THEN } Y \text{ is } B_2 \\
 & \dots \\
 & \text{IF } X \text{ is } A_\omega \text{ THEN } Y \text{ is } B_\omega \\
 & \dots \\
 & \text{IF } X \text{ is } A_\Omega \text{ THEN } Y \text{ is } B_\Omega
 \end{aligned} \tag{1.1}$$

where  $A_\omega$  and  $B_\omega$ ,  $\omega = 1, 2, \dots, \Omega$ , are fuzzy sets defined in the universe of discourses  $V$  and  $W$ , respectively.

For a given system observation, in order to obtain a meaningful inference result based on Zadeh's Compositional Rule of Inference(CRI)[25], there are two basic approaches. The first one is called **FIRST INFER - THEN AGGREGATE** approach, "FITA" for short. In this first approach, for a given system observation  $A'$ , we first perform inference using CRI on each of the rules in the rule base, and then combine all these intermediate results as follows:

$$B' = \bigcup_{\omega=1}^{\Omega} B_{\omega}' \quad (1.2)$$

where  $B_{\omega}'$  is the inference result based on rule  $\omega$ , i.e.,  $B_{\omega}' = A' \circ R_{\omega}$ , where  $R_{\omega} = A_{\omega} \rightarrow B_{\omega}$  is the fuzzy implication relation for rule  $\omega$  and  $\circ$  represents composition within the context of CRI, for example, Sup-min composition, and  $\bigcup$  is a combination operator, i.e.,  $\bigcup \in \{S, T\}$ , in particular,  $\bigcup \in \{\vee, \wedge\}$ .

The second one is called **FIRST AGGREGATE - THEN INFER** approach, "FATI" for short. In this second approach, we first aggregate all the rules by forming an overall fuzzy relation  $R$  which is the combination of all the fuzzy implication relations as follows:

$$R = \bigcup_{\omega=1}^{\Omega} R_{\omega} \quad (1.3)$$

where  $R_{\omega} = A_{\omega} \rightarrow B_{\omega}$ , is the fuzzy implication relation for rule  $\omega$ ,  $\bigcup$  is a combination operator as specified above.

Then an inference is performed for a given observation  $A'$  as follows:

$$B'' = A' \circ R \quad (1.4)$$

where  $\circ$  represents composition within the context of CRI.

Therefore, it is clear that an inference process based on CRI includes several stages. More specifically, it includes **implication, composition, and combination** for FITA, and **implication, combination, and composition** for FATI. In the context of CRI, the comparison and selection of implication and composition operators have been widely studied for one rule case. For example, in [2], [10], and [22], applicability of implication operators is studied under Sup-min composition based on experiments for certain given problems. In [5], it is shown that implication is determined by composition operator, and that Gödel implication is a good implication under Sup-min composition in CRI[6]. In [9] and [23], implication operators are classified into three categories, i.e., S-implication, R-implication, and neither, and their properties are investigated based on some criteria which a Modus Ponens generation function[14] should satisfy. In [15, 16, 17], Interval-Valued Fuzzy Sets are used to represent fuzzy implications and reasoning results. Based on the bounds analysis of fuzzy reasoning, a linkage between CRI and AAR[21] is

established[17].

Inference with multiple rules are investigated by some researchers[1, 2, 3, 10]. In [1], combination operators are suggested for different implications from the consideration of interpretation of ELSE in "IF THEN ELSE" rule. In [3], combination is studied in the domain of fuzzy relational equations. In [2] and [10], both "max" and "min" operators are used in the combination for all implication operators in their experiments.

In this paper, issues of combination of rules or their consequences in fuzzy expert systems using CRI are investigated. A method is proposed for the selection of combination operators from the consideration of the basic requirement for fuzzy reasoning, i.e., if we have a system observation which is the same as the left hand side of a rule in the rule base, then the reasoning result should be the same as the right hand side of the rule. As a result of our analysis, we suggest that for an inference process using Sup-T composition in the context of CRI, "min" be used for combination if the implication  $a \rightarrow b = F(a, b)$  is an Expansion Type and is an inversely proportional function of a, i.e., if  $a_1 \geq a_2$ , then  $F(a_1, b) \leq F(a_2, b)$ , and "max" be used for combination if the implication  $F(a, b)$  is a Reduction Type and is a proportional function of a, i.e., if  $a_1 \geq a_2$ , then  $F(a_1, b) \geq F(a_2, b)$ .

This paper is organized as follows. In Section 2, Compositional Rule of Inference is reviewed, and inference processes are classified into three categories, i.e., Expansion Type Inference, Reduction Type Inference, and Other Types. Further, implication operators under Sup-T composition are classified as Expansion Type, Reduction Type, and Other Types. Finally, in Section 3, two general classes of implication operators are identified to be appropriate for "max" and "min" combinations. Conclusions are stated in the last section. We use either  $\mu_{A \rightarrow B}(a, b)$ , or  $a \rightarrow b$ , or  $F(a, b)$ , or  $R(\rightarrow)$ , or just  $r$  to represent the implication operator in CRI for the convenience of discussion where it is applicable.

## 2. CLASSIFICATION OF INFERENCE PROCESSES

In this section fuzzy inference based on CRI is reviewed. Inference processes based on CRI change the membership function grades of the right hand sides of the corresponding rules either by reducing or by increasing the membership grades. Here we consider reasoning with one rule using CRI.

CRI is also called Generalized Modus Ponens (GMP). With a single rule and a system observation, an inference result can be deduced as follows:

Rule:	IF X is A THEN Y is (should be) B
Observation:	X is A'
Consequence:	Y is (should be) $A' \circ (A \rightarrow B)$

where  $A, A' \subset V$  and  $B \subset W$  are fuzzy sets defined in the universe of discourses  $V$  and  $W$ ,

respectively,  $(A \rightarrow B)$  denotes the implication relation,  $R(\rightarrow)$ , which is a fuzzy set of Cartesian product universe  $V \times W$ , and  $\circ$  denotes the composition between  $A'$  and  $(A \rightarrow B)$ .

The most notable is Zadeh's Sup-min composition in CRI[25], which has the form(in the membership domain) as follows:

$$\mu_{B'}(y_j) = \bigvee_i \mu_{A'}(x_i) \wedge \mu_{A \rightarrow B}(x_i, y_j), \quad i = 1, 2 \dots I, j = 1, 2 \dots J, \quad (2.1)$$

where  $B'$  is the inference result which is a fuzzy set defined in the universe of discourse  $W$ ,  $\mu_{B'}(y_j)$  is the membership value of  $j$ th element of  $B'$ ,  $\mu_{A'}(x_i)$  is the membership value of the  $i$ th element of  $A'$ , and  $\mu_{A \rightarrow B}(x_i, y_j)$  is the membership value of the  $ij$ th element of the implication relation  $R(\rightarrow)$ .

## 2.1 Expansion vs. Reduction Inferences

In this subsection, we present our classification of the inference processes based on their inferred results. More specifically, we classify the inference processes into three categories, i.e., Expansion Type Inference, Reduction Type Inference, and Other Types. Following this point of view, we propose the selection of a proper combination operator such as "max" and "min" as will be discussed in detail later.

**Definition 1.** For a given rule:  $A \rightarrow B$ , and a system observation:  $A'$ , where  $A, A' \subset V$  and  $B \subset W$  are fuzzy sets defined in the universe of discourses  $V$  and  $W$ , respectively, suppose the deduced consequence through an inference process is denoted as  $B'$ , if for any  $A'$ , we always have:

$$B \subseteq B', \quad (2.2)$$

then the inference process is called the "Expansion Type Inference". Suppose, on the other hand, the deduced consequence is denoted as  $B^*$ , if for any  $A'$ , we always have:

$$B^* \subseteq B, \quad (2.3)$$

then the inference process is called the "Reduction Type Inference". Further, if the deduced consequence is at some times  $B \subseteq B'$ , and at some times  $B^* \subseteq B$ , then the inference process is called the "Other Type Inferences".

After Zadeh's Sup-min composition in CRI was proposed, Sup-T composition has been studied by many researchers[e.g., 6, 12, 15]. In [2, 10], the behaviours of many implication operators are studied using Sup-min composition in the context of CRI for certain specific problems. In this paper, it is assumed that Sup-T is used for composition in CRI in order to cover the general cases, and that all fuzzy sets are normalized.

Without proof here, we have the following theorem for the classification of inference processes.

**Theorem 1.** For Sup-T composition in the context of CRI, if the implication  $a \rightarrow b = F(a, b) \geq b$  for all  $a \in [0, 1]$ , then the inference process is "Expansion Type Inference". If the implication  $a \rightarrow b = F(a, b) \leq b$  for all  $a \in [0, 1]$ , then the inference is "Reduction Type Inference". If the implication  $a \rightarrow b = F(a, b) \geq b$  for some  $a \in [0, 1]$ , but  $F(a, b) \leq b$  for some other  $a \in [0, 1]$ , then the inference is "Other Type Inference".

According to Theorem 1, for a given implication operator, we can determine whether an inference process is an Expansion or a Reduction Type Inference under Sup-T composition. Thus, if we use Sup-T composition, those implication operators can be classified into three categories: the Expansion Type implication, Reduction Type implication, and Other Types. If Sup-T composition is used, then it is easy to show some implication operators proposed in the literature are Expansion Type implications, e.g.,  $\min(1, 1-a+b)$ ; some are Reduction Type ones, e.g.,  $\min(a,b)$ ; and some are Other Type implications, e.g.,  $\max(1-a, \min(a,b))$ .

### 3. PROPER COMBINATION OPERATOR

Unless we have an exact match between a system observation and the antecedent of a rule, we need more than one rule to deduce a meaningful result by combining the intermediate results based on each of the rules. In this section, we first discuss the basic requirement for an inference process. We then propose a method to select combination operators for both Expansion and Reduction inference processes from the consideration of the basic requirement for fuzzy reasoning.

#### 3.1 Basic Requirement for Fuzzy Reasoning

The basic requirement for fuzzy reasoning with one rule is that: given a rule  $A \rightarrow B$ , if the system observation is  $A' = A$ , then the deduced result should be  $B$ . Some researchers have studied this property[e.g., 4, 5, 6, 13, 14]. For example, in [5], for a given composition  $m$ , an implication operator  $I$  is derived such that  $A \circ_m (A \rightarrow B) = B$ . It is shown [5] that for Sup-T composition, denoted as  $\circ_{s,t}$ , and R-implication where the same t-norm operator as in the Sup-T is used, denoted as  $\rightarrow_R$ , we have  $A \circ_{s,t} (A \rightarrow_R B) = B$ . For example, in CRI, if Sup-min composition is used, Gödel implications have this property[6]. In [14], for a given implication function  $I$ , a modus ponens function  $m$  is derived, such that  $A \circ_m (A \rightarrow B) = B$ .

As mentioned previously, we need more than one rule to perform inference unless we have an exact match between the system observation and the left hand side of a rule. Suppose there are  $\Omega$  rules in the rule base. For each of the rules, we have a reasoning result which we need to combine to obtain an overall inference result. We propose that a fuzzy inference process, with multiple rules, should satisfy the basic requirement for fuzzy reasoning stated as follows.

**Criterion 1.** The basic requirement for fuzzy reasoning, with multiple rules, is that given  $\Omega$

rules: IF X is  $A_\omega$  THEN Y is  $B_\omega$ ,  $\omega = 1, 2, \dots, \Omega$ , if observation is  $A' = A_\omega$ , then reasoning result  $B' = B_\omega$ .

This criterion is important to the reliability of an expert system. More specifically, this criterion requires that when given a system observation which is one of the left hand sides of the rules, a fuzzy expert system will return the same conclusion as in the rule.

With the presentation of multiple rules, we have to deal with the combination problem as mentioned previously. In [1], combination operators are suggested for different implications from the consideration of interpretation of ELSE in "IF THEN ELSE" rule. In [3], the problem is studied in the domain of fuzzy relational equations. In [2] and [10], both "max" and "min" operators are used in the combination for all implication operators in their experiments. In what follows, from the consideration of the requirement for fuzzy reasoning processes stated above, we propose a method for the selection of combination operators for both the Expansion Type Inference and the Reduction Type Inference processes.

### 3.2 Combination: min vs. max

For a given system observation, we can perform inference by CRI with two approaches as indicated in Section 1, i.e., "FITA" and "FATI" approaches. The question is "what must be the proper combination operator for (1.2) and (1.3)?", i.e., "must  $\cup$  be max or min"? As discussed in Section 2, if Sup-T composition is used, then the category of an inference process can be determined by the implication operator, i.e., if  $a \rightarrow b = F(a,b) \geq b$ , then the process is an Expansion Type Inference, and if  $a \rightarrow b = F(a,b) \leq b$ , then the inference process is a Reduction Type Inference. Therefore, in this sense, (1.2) and (1.3) are consistent in terms of reasoning results.

#### 3.2.1 Expansion Inference Process

In an Expansion Inference process, with Definition 1 in Section 2.1, for any system observation  $A'$ , we always have:

$$B \subseteq B'.$$

For an expansion inference process based on CRI, we have Necessary condition 1 as follows.

**Necessary condition 1.** Suppose there are  $\Omega$  rules in the rule base of a fuzzy expert system. For a system observation and an inference process using Sup-T composition in the context of CRI, if implication  $a \rightarrow b = F(a, b) \geq b$  for all  $a \in [0,1]$ , and is an inversely proportional function of  $a$ , i.e., if  $a_1 \geq a_2$ , then  $F(a_1, b) \leq F(a_2, b)$ , then "min" is needed for the combination.

The proof of Necessary condition 1 is based on the following idea: for a very low level of similarity(matching)[e.g., 26] between the observation and the left hand side of a rule and in the limit including the case of no match at all, i.e., no overlap, the membership function grade of the inferred result based on that rule has a value equal to(approaching) 1.0 in the limit at each

support point, i.e., this rule creates "unknown". Hence the use of this rule is useless and in this case it does not infer any useful information. Thus, considering the "Criterion 1" and getting a meaningful result for any system observation, we must use "min" for the combination, which will eliminate this useless information.

### 3.2.2 Reduction Inference Process

In a Reduction Type Inference process, with Definition 1 in Section 2.1, for any observation  $A'$ , we always have:

$$B^* \subseteq B.$$

For a reduction inference process based on CRI, we have Necessary condition 2 as follows.

**Necessary condition 2.** Suppose there are  $\Omega$  rules in the rule base of a fuzzy expert system. For a system observation and an inference process using Sup-T composition in the context of CRI, if implication  $a \rightarrow b = F(a, b) \leq b$  for  $a \in [0,1]$ , and is a proportional function of  $a$ , i.e., if  $a_1 \geq a_2$ , then  $F(a_1, b) \geq F(a_2, b)$ , then "max" is needed for the combination.

The proof of Necessary condition 2 is based on the following idea: for a very low level of similarity(matching) between the observation and the left hand side of a rule and in the limit including the case of no match at all, i.e., no overlap, the membership function grade of the inferred result has a value equal to 0 in the limit at each support point. That is, the use of this rule generates "meaningless". Considering the "Criterion 1" of the fuzzy inference and getting a meaningful result for any system observation, we must use "max" for the combination.

Necessary conditions 1 and 2 establish the choice of a combination operator for both Expansion and Reduction inference processes. In other words, after we select the implication and composition operator in CRI, then we could determine the combination operator in accordance with Necessary conditions 1 and 2.

## 4. CONCLUSIONS

In this paper, we analyzed fuzzy inference method of CRI in terms of inference results. Inference processes are classified into three categories, i.e., the Expansion Type Inference, Reduction Type Inference, and other types, which can be determined based on the implication operator under Sup-T composition in CRI. Based on the basic requirement of fuzzy reasoning stated as Criterion 1, we suggest that for an inference process using Sup-T composition in the context of CRI, "min" be used for the combination if the implication  $F(a, b)$  is an Expansion Type and is an inversely proportional function of  $a$ , and "max" be used for combination if the implication is a Reduction Type and is a proportional function of  $a$ . Therefore, we have general conclusions for both Expansion and Reduction inference processes based on the reasoning results no matter which inference process is used. This proposed principle is also consistent with the existing results in the literature[e.g., 1, 2, 3, 10]. Our method can be used as a guidance to select operators in the

design of fuzzy expert systems and fuzzy controllers. More specifically, for an inference process using Sup-T composition, we first identify the class of an implication operator as discussed in Section 2; then select the combination operator according to Necessary condition 1 or 2. For example, in [2] and [10], the pair of operators:  $\mu_{R12}$  and  $\mu_{R12^*}$ , is not necessary since they are Reduction Type but are not directly proportional(not non-decreasing) functions of a; the pairs of operators:  $\mu_{R14}$  and  $\mu_{R14^*}$ ,  $\mu_{R23}$  and  $\mu_{R23^*}$ ,  $\mu_{R30}$  and  $\mu_{R30^*}$ , and  $\mu_{R32}$  and  $\mu_{R32^*}$ , are not necessary since they are Expansion Type but are not inversely proportional(not non-increasing) functions of a; because  $\mu_{R3}$ ,  $\mu_{R4}$ ,  $\mu_{R5}$ ,  $\mu_{R6}$ ,  $\mu_{R22}$ ,  $\mu_{R27}$ , and  $\mu_{R29}$  are the Expansion Type, therefore "min" must be used for the combination for each of these processes. In other words,  $\mu_{R3^*}$ ,  $\mu_{R4^*}$ ,  $\mu_{R5^*}$ ,  $\mu_{R6^*}$ ,  $\mu_{R22^*}$ ,  $\mu_{R27^*}$ , and  $\mu_{R29^*}$  are "appropriate" candidates. And because  $\mu_{R8^*}$ ,  $\mu_{R25^*}$ , and  $\mu_{R31^*}$  are the Reduction Type, "max" must be used for the combination for each of these processes. In other words,  $\mu_{R8}$ ,  $\mu_{R25}$ , and  $\mu_{R31}$  are "appropriate" candidates. Since Necessary conditions 1 and 2 establish the selection of combination operators for the Expansion and the Reduction Type inferences, we suggest that appropriate combination operators be selected in the design of fuzzy expert systems.

It should be noted that in order to satisfy Criterion 1, the membership functions of the linguistic terms of a rule in the rule base of an expert system must satisfy some constraints or conditions[19].

In this paper we always make reference to CRI in one or another to remind the readers that there are other approximate reasoning methods such as, for example, Approximate Analogical Reasoning method[21]. Issues of combination for these other methods should also be investigated in a similar manner in the future.

## ACKNOWLEDGEMENT

This work is supported in part by the Natural Science and Engineering Research Council of Canada, and in part by the Manufacturing Research Corporation of Ontario.

## REFERENCES

- [1] J.F. Baldwin and B.W. Pilsworth, A model of fuzzy reasoning through multi-valued logic and set theory, I.J. Man-Machine Studies 11 (1979) 351-380.
- [2] Z. Cao and A. Kandel, Applicability of some fuzzy implication operators, Fuzzy Sets and Systems, 31 (1989) 151-186.
- [3] A. Di Nola, W. Pedrycz, and S. Sessa, Fuzzy relation equations and algorithms of inference mechanism in expert systems, in: M.M. Gupta et al., Eds., Approximate Reasoning in Expert

Systems (Elsevier Science, Amsterdam, 1985) 355-367.

[4] D. Driankov, Inference with a single fuzzy conditional proposition, *Fuzzy Sets and System* 24 (1987) 51-63.

[5] D. Dubois and H. Prade, Fuzzy logics and the generalized modus ponens revisited. *I.J. of Cybernetics and Systems*, 15 (1984) 293-331.

[6] D. Dubois and H. Prade, The generalized Modus Ponens under sup-min composition -A theoretical study, in *Approximate Reasoning in Expert Systems*, M.M. Gupta et al., Eds, (Elsevier Science, Amsterdam, 1985) 217-232.

[7] D. Dubois and H. Prade, Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions, *Fuzzy Sets and Systems* 40 (1991) 143-202.

[8] S. Fukami, M. Mizumoto and K. Tanaka, Some considerations on fuzzy conditional inference, *Fuzzy Sets and Systems* 4 (1980) 243-283.

[9] L.O. Hall, The choice of ply operator in fuzzy intelligent systems, *Fuzzy Sets and Systems* 34 (1990) 135-144.

[10] J.B. Kiszka, M.E. Kochanska and D.S. Sliwinska, The inference of some fuzzy implication operators on the accuracy of a fuzzy model - part I, II, *Fuzzy Sets and Systems*, 15 (1985) 111-128, 223-240.

[11] E.M. Mamdani, Advances in linguistic synthesis of fuzzy controllers, *I.J. Man-Machine Studies* 8 (1976) 669-678.

[12] M. Mizumoto, Fuzzy inference using max- $\Lambda$  composition in the compositional rule of inference, In M. Gupta and E. Sanchez, Eds, *Approximate reasoning in decision analysis*, (Amsterdam, North-Holland, 1982).

[13] M. Mizumoto and H.J. Zimmermann, Comparison of fuzzy reasoning methods, *Fuzzy Sets and Systems*, 8 (1982) 253-283.

[14] E. Trillas and L. Valverde, On mode and implication in approximate reasoning, in : M.M. Gupta et al., Eds, *Approximate reasoning in expert systems* (Elsevier Science, Amsterdam, 1985), 157-166.

[15] I.B. Turksen, Interval valued fuzzy sets based on normal forms, *Fuzzy Sets and Systems*, 20 (1986) 191-210.

[16] I.B. Turksen, Four methods of approximate reasoning with interval-valued fuzzy sets, *I.J. of Approximate Reasoning*, 3 (1989) 121-142.

- [17] I.B. Turksen and Y. Tian, Interval-valued fuzzy sets representation of multiple antecedent S-implication and reasoning, *Fuzzy Sets and Systems*(to appear).
- [18] I.B. Turksen and Y. Tian, Search schemes of a fuzzy expert system, Submitted to *IEEE Transactions on SMC*.
- [19] I.B. Turksen and Y. Tian, Combination of rules or their consequences in fuzzy expert systems, Submitted to *Fuzzy Sets and Systems*.
- [20] I.B. Turksen, Y. Tian and M. Berg, A fuzzy expert system for a service centre of spare parts, *I. J. of Expert Systems with Applications*, 5 (1992) 447-464.
- [21] I.B. Turksen and Z. Zhong, An approximate analogical reasoning approach based on similarity measures, *IEEE Trans. on Sys., Man, and Cybern.*, Vol. SMC-18 (1988) 1049-1056.
- [22] T. Whalen and B. Schott, Alternative logics for approximate reasoning in expert systems: a comparative study, *I.J. Man-Machine Studies* 22 (1985) 327-346.
- [23] T. Whalen and B. Schott, Presumption and prejudice in logical inference, *I.J. of Approximate Reasoning* 3 (1989) 359-382.
- [24] R. Yager, An approach to inference in approximate reasoning, *I.J. Man-Machine Studies* 13 (1980) 323-338.
- [25] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems, Man and Cybernetics*, SMC-3 (1973), 28-44.
- [26] R. Zwick, E. Carlstein, and D.V. Budescu, Measurement of similarity between fuzzy concepts: A comparative analysis, *I.J. of Approximate reasoning*, 1 (1987) 221-241.

## APPROXIMATE REASONING USING TERMINOLOGICAL MODELS

John Yen  
Center for Fuzzy Logic and Intelligent Systems  
Dept. of Computer Science  
Texas A&M University  
College Station TX 77843  
yen@cs.tamu.edu

Nitin Vaidya  
Texas Transportation Institute  
Texas A&M University  
701 North Post Oak #430  
Houston TX 77024  
nvaidya@cs.tamu.edu

### ABSTRACT

Term Subsumption Systems (TSS) form a knowledge representation scheme in AI that can express the defining characteristics of concepts through a formal language that has a well-defined semantics and incorporates a reasoning mechanism that can deduce whether one concept subsumes another. However, TSS have very limited ability to deal with the issue of uncertainty in knowledge bases. The objective of this research is to address issues in combining approximate reasoning with term subsumption systems. To do this we have extended an existing AI architecture (CLASP), that is built on the top of a term subsumption system (LOOM), in the following ways. First, the assertional component of LOOM has been extended for asserting and representing uncertain propositions. Second, we have extended the pattern matcher of CLASP for plausible rule-based inferences. Third, an approximate reasoning model has been added to facilitate various kinds of approximate reasoning. And finally, the issue of inconsistency in truth values due to inheritance is addressed using justification of those values. This architecture enhances the reasoning capabilities of expert systems by providing support for reasoning under uncertainty using knowledge captured in TSS. Also, as definitional knowledge is explicit and separate from heuristic knowledge for plausible inferences, the maintainability of expert systems could be improved.

### 1. INTRODUCTION

Knowledge exists in a variety of forms [1]. While most existing expert systems employ one or two knowledge representation schemes, expressing diverse knowledge in such a limited number of representation formalisms is difficult and time-consuming. Furthermore, it may not be possible to express completely all the knowledge required in an expert system. So, there is a need to integrate different knowledge representation schemes and to deal with the issue of incompleteness in a knowledge base. The objective of this research is to address these issues by combining two knowledge representation schemes, approximate reasoning and terminological reasoning.

Approximate reasoning concerns uncertain knowledge and data in expert systems. Uncertainty in expert systems may arise because of incompleteness in data, unreliability of data, impreciseness of data, or even uncertain knowledge. For example, judgmental knowledge used in medical expert systems is uncertain in nature. Hence, expert systems need to handle uncertainty in such a way that the conclusions are understandable and interpretable by the user [10]. In approximate reasoning, fuzzy logic makes it possible to deal with different types of uncertainty within a single framework as it subsumes predicate logic. It is suitable for inferring from imprecise knowledge as all uncertainty is allowed to be expressed as a matter of degree [22]. In addition fuzzy logic provides suitable operators for the combination of uncertainty, including a generalized modus ponens following from Zadeh [22] for making inferences based on rules.

Term Subsumption Systems (TSS), on the other hand, deal with terminological (i.e. definitional) knowledge. The representation scheme of term subsumption systems can express the defining characteristics of concepts through a formal language that has a well-defined semantics. The semantics of constructs that are often used to define concepts or roles are shown in Figure 1. Term subsumption systems provide a natural organization for terminological knowledge [3] through a structured taxonomy of conceptual entities with associated descriptions, which satisfy certain restrictions as well as have specific relationships to each other and where specific concepts can indirectly inherit characteristics from more general concepts. A guiding principle is that concepts are formal representational objects and that the epistemological relationships between formal objects must be kept distinct from the things represented by these formal objects [2]. For example the concept **Rich-Person** must be kept separate from an instance of **Rich-Person**. An example of terminological knowledge is shown in figure 2. In addition, the reasoning mechanism in these languages can deduce whether one concept subsumes another [12]. An automatic classifier places a concept in its proper location in a taxonomy so as to enforce network semantics

and consistency checking of logical subsumption relations between concepts [13]. Term subsumption systems originate from the ideas presented in the KL-ONE knowledge representation system, which was itself derived from semantic network formalisms [7]. Because of the formal semantics employed, term subsumption systems can be viewed as a generalization of frames and semantic networks [6], [17].

In this work we have extended two terminological architectures for approximate reasoning; LOOM and CLASP, which is built on top of LOOM. This paper has partly originated from Yen and Bonissone, who have both addressed the issue of extending TSS for uncertainty management and outlined a generic architecture in [19], and has been derived from Vaidya in [16].

Expression e	Interpretation [e]
:primitive $C_1$	a unique primitive concept
:primitive $R_1$	a unique primitive relation
(:and $C_1 C_2$ )	$\lambda x. [C_1] \wedge [C_2]$
(:and $R_1 R_2$ )	$\lambda xy. [R_1] \wedge [R_2]$
(:at-least 1 R)	$\lambda x. \exists y. [R](x,y)$
(:exactly 1 R)	$\lambda x. \exists y. [R](x,y) \wedge \forall yz. ([R](x,y) \wedge [R](x,z)) \rightarrow y = z$
(:all R C)	$\lambda x. \forall y. [R](x,y) \rightarrow [C](y)$
(:domain C)	$\lambda xy. [C](x)$
(:range C)	$\lambda xy. [C](y)$

Figure 1. Semantics of Some Terminological Expressions

```
(defconcept RICH-PERSON :is :p)
(defconcept MILLIONAIRE :is (:and :p RICH-PERSON))
(defconcept BILLIONAIRE :is (:and :p MILLIONAIRE))
(defconcept CAR :is :p)
(defconcept NEW-CAR :is (:and :p CAR ))
(defrelation HAS-CAR :is (:and :p (:domain PERSON) (:range CAR)))
```

Figure 2. An Example of Terminological Knowledge

## 2. ISSUES IN APPROXIMATE REASONING WITH TERMINOLOGICAL MODELS

In this section we outline four issues that need to be addressed in integrating approximate reasoning with terminological systems. This paper will focus on the first three issues. The fourth issue has been addressed in [18].

(1) *Extending the assertional component of a TSS for stating uncertain propositions:* One form of uncertainty in TSS concerns the uncertainty about the "instance of" relation. For example if there is a concept **Rich-Person**, a person may be a **Rich-Person** only to a certain extent. This issue concerns representing and asserting uncertain propositions and requires extension of the assertional component of TSS (often referred as the ABox).

(2) *Maintaining consistency of truth values associated with propositions:* Another issue needs to be dealt with. This is related to the inheritance of concepts. The truth value of an instance in a concept may be inconsistent with the truth value of the same instance in another concept which subsumes the first concept or is subsumed by the first concept. For example, the degree of membership in the concept **Millionaire** can not be lower than the degree of membership in the concept **Billionaire** as a **Millionaire** subsumes a **Billionaire**. As such, a truth

maintenance mechanism is required to maintain consistency of truth values of the propositions.

(3) *Extending the semantic pattern matcher for partial matching:* Another form of uncertainty could occur in the judgmental knowledge for reasoning with assertional components of term subsumption systems. For example, an owner of a new car may or may not be a rich person. From experts or from statistical data we may obtain a number to represent the likelihood that a person who owns a new car is also a rich person. This second issue concerns integration of such uncertainty with the uncertainty represented in the assertional component of the term subsumption language. For this purpose the Semantic Pattern Matcher of CLASP needs to be modified for performing partial matching of conditions.

(4) *Extending the semantics of terminological component of TSS for making plausible inferences:* This issue concerns the representation of and reasoning with uncertainty in the terminological knowledge of term subsumption systems.

### 3. INTEGRATING APPROXIMATE REASONING WITH TERMINOLOGICAL MODELS

The architecture for integrating approximate reasoning with term subsumption systems is an extension of the architecture of CLASP. For incorporating approximate reasoning the architecture has extended LOOM to include a representation scheme for uncertain propositions, a fuzzy assertional language for asserting and retracting such propositions, a fuzzy truth maintenance system and an assertion processor. Moreover, the architecture has extended CLASP and provides for representation of uncertain rules, a fuzzy rule language, a modification to the semantic pattern matcher of CLASP for partial matches and an approximate reasoner which reasons with the uncertainty expressed in instances and rules. The architecture is represented in Figure 3.

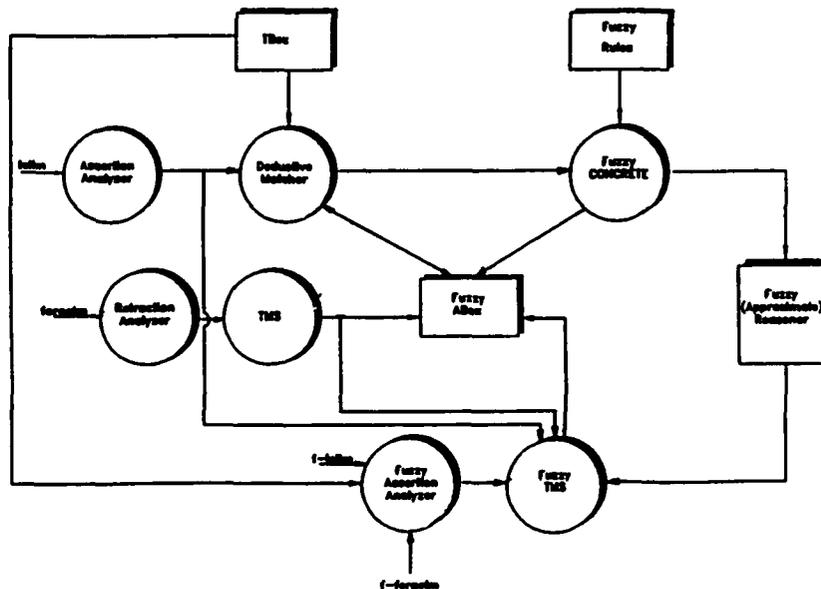


Figure 3. Architecture for Approximate Reasoning Using Terminological Models

#### 3.1 Extended Assertional Component

##### 3.1.1 A Fuzzy Assertional Language

The extended assertional language includes a truth value which expresses the degree of certainty of the membership of an instance in the corresponding concept or role. Please refer to Figure 5 for examples of the assertional language. It may be noted that the *f-tellm* statement causes assertion of propositions, whereas the *f-forgetm* statement causes retraction of propositions.

### 3.1.2 Internal Representation

The internal representation has been extended to include a representation for uncertainty in instances and also includes a justification structure for uncertainty. This representation scheme is the basis for truth maintenance and reasoning in the system. An example of internal representation of an instance is given in Figure 4.

---

```
(Instance(John)
.
(fuzzy-db-type: ((Rich-Person 0.5))
(fuzzy-role: ((Has-Car Mercedes) 0.7))
.
(justification-for-uncertainty:
  ( (RoleOrConcept: Rich-Person
    Certainty-Measure: 0.5
    Origin: Rule<New-Car-Owners-Are-Rich>)
    (RoleOrConcept: (Has-Car Mercedes)
    Certainty-Measure: 0.7
    Origin: "USER")
  )
)
)
```

Figure 4. Example of Internal Representation of an Instance.

---

## 3.2 Fuzzy Truth Maintenance System (FTMS)

The Fuzzy Truth Maintenance System (FTMS) performs consistency checking for truth values on all assertions, retractions and inferences.

### 3.2.1 Consistency Checking for Truth values of Propositions

An fuzzy proposition in a fuzzy TSS needs to be checked for consistency because the truth value of a fuzzy proposition may be constrained by the truth values of other fuzzy propositions. The truth value representing the degree of membership of an instance in a concept needs to be compared with the truth values for the same instance in other concepts below or above C in the concept subsumption hierarchy. Such a comparison is based on the following two general principles:

- (1) The truth value of an instance in a concept C cannot be greater than the truth value of the same instance in any of C's parent concepts.
- (2) The truth value of an instance in a concept C cannot be less than the truth value of the same instance in any of C's children concepts.

In summary, if  $C_1 > C_2$  then  $\mu_{C_1}(x) \geq \mu_{C_2}(x)$  where ">" denotes the subsumption relation between concepts.

To illustrate the above, assume concept  $C_1$  subsumes concept  $C_2$ , which subsumes concept  $C_3$ . Now if an instance has a degree of membership  $a_1$  in  $C_1$ ,  $a_2$  in  $C_2$  and  $a_3$  in  $C_3$ , then the condition  $a_1 \geq a_2 \geq a_3$  must be satisfied. Any assertion or retraction that result in truth values that violate this condition is inconsistent. An example of inconsistency is shown in Figure 5.

There are a number of sources that may cause inconsistency in truth values of data. Because inconsistencies due to different sources need to be handled differently we list possible sources of inconsistency below:

- (1) Inconsistency due to deduction based on the terminological model.
- (2) Inconsistency due to an assertion or retraction by the user.
- (3) Inconsistency due to an inference made by fuzzy rules.

---

Refer to terminological knowledge in Figure 2 and fuzzy-rule in Figure 7.  
Consider the following sequence of assertions:

(f-tellm ((Has-Car John Mercedes) 0.7))  
(f-tellm ((New-Car Mercedes) 0.5))  
(f-tellm ((Millionaire John) 0.6))

The the first two assertions would cause the fuzzy-rule to fire and result in the inference

((Rich-Person John) 0.5)

However, the last assertion would cause an inconsistency as the truth value of John being a Billionaire (0.6) exceeds the previously inferred truth value of his being a Rich-Person (0.5) though Rich-Person subsumes Billionaire.

#### Figure 5. Example of Inconsistency

---

To deal with inconsistency, we have developed a fuzzy truth maintenance system (FTMS) that processes these different kinds of inconsistencies. This FTMS records the justification of propositions in a list of justification structures associated with each instance. A justification structure specifies (i) a fuzzy proposition and (ii) whether the proposition was asserted by the user, deduced by the terminological model, or inferred by a rule. For example, the justification structure in Figure 5 indicates that the justification that John may be a house-owner with a truth value of 0.5 is that a rule "Rich-People-Are-House-Owners" made such an inference. Whenever a new fuzzy proposition is added to by the system, the FTMS incorporates the truth value of the current proposition with the truth values for the same proposition in the justification list. If there is an inconsistency, then the user is notified, else the modification is completed. If the proposition is a binary predicate, the consistency checking uses the role subsumption lattice. An algorithm for truth maintenance of propositions is outlined in Figure 6.

#### 3.2.2 Assertion Processor

The assertion processor translates user asserted statements and fuzzy rule inferences into internal assertional changes and propagates these changes to the deductive reasoner and the approximate reasoner. Asserted propositions have the highest precedence followed by propositions deduced by the deductive reasoner. Propositions inferred by the approximate reasoner have the lowest precedence. The deductive reasoner overrides the plausible inference of the approximate reasoner when a deduction is made, and when the deduced proposition is retracted the plausible conclusion is reactivated.

#### 3.3 Extending the Semantic Pattern Matcher for Partial Matching

We have modified CONCRETE, the pattern matcher of CLASP, for plausible rule based inferences. CONCRETE is a semantic pattern matcher which uses a combination of Forgy's Rete pattern matcher and LOOM's deductive pattern matcher [20],[21]. We first outline the fuzzy rule language. Then we describe the deductive pattern matcher of LOOM and semantic pattern matcher (CONCRETE) of CLASP and our extension to semantic pattern matcher for partial match. Finally, we describe the approximate reasoner for plausible inferences.

---

*Module Update-Fuzzy-DB(P,T)*

1. Let the fuzzy proposition, P be  $[\alpha, \mu_i]$ , where  $\alpha$  is the argument of proposition and  $\mu_i$  is the truth value of the proposition and T is the "type" of the fuzzy proposition, i.e., one of asserted by user, retracted by user, inferred by fuzzy rule or deduced by terminological model.
2. If a fuzzy proposition P is asserted by the user then perform *Consistency-Checker* for the asserted truth value  $\mu_i$ .
2. If T is either inferred by a fuzzy rule, or is deduced by the terminological model (e.g., inheritance links), or is retraced by the user, then
  - (a) (1) if a justification structure of the proposition exists then compute the new truth value  $\mu_j$  of the proposition else  
(2) Create a justification structure if it does not exist and assign the value of  $\mu_i$  to  $\mu_j$ .
  - (b) (1) Create a fuzzy proposition  $P_j$  as  $[\alpha, \mu_j]$ .  
(2) Perform *Consistency-Checker(P<sub>j</sub>)* for the resultant truth value.
3. If *Consistency-Checker(P<sub>j</sub>)* returns true then
  - (a) Update the justification structure as follows: If T is retraction by user remove the fuzzy proposition P from it else add the fuzzy proposition {P,T} to the it.
  - (b) Update the proposition in the fuzzy database to  $P_j$ .
  - (c) Return True.

*Module Consistency-Checker(P)*

1. Let the fuzzy proposition, P be  $[\alpha, \mu_j]$ , where  $\alpha$  is the argument of proposition and  $\mu_j$  is the truth value of the proposition.
2. Find all parent fuzzy propositions with the same argument  $\alpha$  in the fuzzy database.
3. Let ConsistencyCheck be the logical conjunction of the values returned by *Parent-Check(P, P<sub>j</sub>)* for each parent fuzzy proposition  $P_j = > [\alpha, \mu_s]$ .
4. Return ConsistencyCheck.

*Module Parent-Check(P, P<sub>j</sub>)*

1. If  $P_j$  subsumes P and  $\mu_s < \mu_j$ , notify the user of inconsistency. Let ReturnValue be False.
2. If P subsumes  $P_j$  and  $\mu_j < \mu_s$ , notify the user of inconsistency. Let ReturnValue be False.
3. If neither of the above, then let ReturnValue be assigned the value returned by *Update-Fuzzy-DB(P<sub>j</sub>, deduced\_by\_terminological\_model)*
4. Return the ReturnValue.

**Figure 6. Algorithm for Truth Maintenance**

---

### 3.3.1 Fuzzy Rule Language

Uncertainty in a rule may be expressed in the consequent side of the rule which is assertional in nature. Example of a fuzzy rule is given in Figure 7.

---

```
(def-fuzzy-rule New-Car-Owners-Are-Rich
  :if (:AND (NEW-CAR ?y)
            (HAS-CAR ?x ?y) )
  :then ((RICH-PERSON ?x) 0.6) )
```

**Figure 7. Example of a Fuzzy Rule**

---

Note that the actual truth value to be recorded for an inferred proposition as a result of the firing of the rule may, however, be different from the truth value of the rule as a consequence of approximate reasoning and truth maintenance.

### 3.3.2 Semantic Pattern Matching in LOOM and CLASP

Terminological knowledge can be viewed as a perspicuous encoding of bidirectional definitional rules. In classification based systems, an instance is matched to a pattern, by the realizer, by first abstracting it and then by classifying the abstraction [11]. A concept P is associated with a pattern P(x); thus matching an individual to a pattern corresponds to recognizing an instantiation relationship between the individual and the corresponding concept.

The deductive pattern matcher in LOOM is an extension to the realizer [11]. The classifier in LOOM's pattern matcher can ask questions about the individual being classified during classification, using backward chaining, and a sufficiently detailed abstraction is built up incrementally. In addition the pattern matcher can also perform a forward inference. Thus it has mixed both forward deduction and backward deduction.

The semantic pattern matcher in CLASP combines Forgy's Rete Pattern Matcher with the deductive matcher of LOOM. The rule compiler builds a concept classification Rete (CONCRETE) net as rules are loaded into the rule base. The LOOM matcher computes assertional changes that can be deduced from the terminological knowledge and it informs the CONCRETE net about relevant changes. To avoid long chains of CONCRETE nodes and early unnecessary joins a data dependency analysis is performed on the patterns [20],[21].

### 3.3.3 Semantics-based Fuzzy Pattern Matching

To deal with uncertainty, a fuzzy pattern matcher needs to handle tokens that express uncertainty. For this it needs to record the degree of match, which is the extent to which an uncertain token matches a condition of a rule, in appropriate nodes in the CONCRETE net. The pattern matcher also needs to combine the partial matches as tokens are propagated down the CONCRETE net. The fuzzy pattern matcher also needs to generate instantiations of fuzzy rules. In addition, as concept nodes of type TRUE do not have their own memory, the pattern matcher needs to query LOOM about partial class memberships.

The pattern matcher of CLASP, CONCRETE, has been modified in three ways.

- (1) The pattern matcher has been extended to query LOOM for partial class memberships.
- (2) The instantiation structure of the CONCRETE has been extended to represent the degree of partial matching.
- (3) The updating mechanism for a node has been modified to calculate or update the matching degree

of instantiations stored in the node's memory.

### 3.3.4 Approximate Reasoner

The approximate reasoner makes plausible inferences based on terminological knowledge, fuzzy propositions and uncertain rules. It also interacts with the FTMS to maintain consistency of the propositions database and to infer truth values to be used in the recording of inferred propositions. The use of justification structures in FTMS also helps in the combination of truth values associated with the same inference in different rules. In addition the approximate reasoner informs the deductive reasoner about only those additions or deletions to the propositions database whose certainty degree is one. Moreover, the deductive reasoner informs the approximate reasoner about all additions or deletions to the propositions database.

#### 3.3.4.1 Uncertainty Calculi

The approximate reasoning model can support different kinds of approximate reasoning. The user may specify the model he wishes to chose. At present two models are supported. Both are based on triangular norms in fuzzy logic [4].

Uncertainty is propagated using T-norm operators in fuzzy logic. T-norms are binary functions that satisfy conjunction while T-conorms are binary functions that satisfy disjunction. Both are 2-place  $[0,1] \times [0,1]$  to  $[0,1]$  functions that are monotonic, commutative and associative and their corresponding boundary conditions satisfy the truth tables of the logical AND and OR operators. A function  $T(a,b)$  aggregates the degree of certainty of two clauses in the same premise. A function  $S(a,b)$  aggregates the degree of certainty of the same conclusions derived from two rules. The associativity property may be used for representation of conjunction of a large number of clauses.

The user may select one of the two following types of T-norm operators:

- (a)  $T1(a,b) = ab$  and  $S1(a,b) = a + b - ab$
- (b)  $T2(a,b) = \min(a,b)$  and  $S2(a,b) = \max(a,b)$

#### 3.3.4.2 Inference Mechanism

The reasoner performs plausible inference in a data driven, forward-chaining manner. Fuzzy rules only specify plausible inferences which in turn update instances. As a result of firing of these fuzzy rules, the truth-value of an instance in a concept or in a role may be added or updated.

A fuzzy rule, after firing once, can be instantiated again if

- (1) When one of the conditions in its pattern is no longer satisfied, or
- (2) An assertion or inference by another rule updates the truth-value of an existing proposition.

## 4. RELATED WORK

Lokendra Shastri has developed a framework, based on the principle of maximum entropy, for dealing with representation of and reasoning with semantic networks [14],[15]. His framework treats statements as evidential assertions, assigning a number to each to represent the evidential import. Given statistical data it can answer questions like "given the state of knowledge of an agent, which choice is most probably correct". While his framework can handle exceptions, multiple inheritance and ambiguities, it has two limitations. First, his approach is based on the availability of statistical data which may not be available. Second, there is no classifier to maintain the consistency of the terminology because the concepts and roles are not of the definitional type.

Heinsohn and Owsnicki have proposed a model of probabilistic reasoning in hybrid term subsumption systems

[9]. Uncertain knowledge is represented as probabilistic implications and probabilistic inheritance is used as a reasoning mechanism. They consider universal knowledge to be related to the extensions of concepts, i.e., the set of real world objects. This empirical or belief knowledge is stored in a Probabilistic Box (PBox). They have extended a term subsumption language by defining the syntax and semantics of probabilistic implication, which quantifies the relative degree of intersection of two extensions. While the range of applicability of hybrid term subsumption systems may be enlarged with this model, it is limited in the kind of uncertain knowledge it can represent. Most rules in expert systems involve complex conditions which may not be completely expressible as concept definitions. Therefore, probabilistic implications need to be extended before these could be used for building expert systems.

Bonissone *et al.* have developed a T-norm based reasoning architecture, RUM, for frame based systems [5]. The premise is that treatment of uncertainty must address representation, inference and control layers in expert systems. The representation uses a certainty frame with set of associated slots. However, the limitation of RUM is that it cannot use terminological knowledge, unlike term subsumption systems.

## 5. SUMMARY

An architecture has been implemented and described for approximate reasoning with terminological systems. The assertional component has been extended for representing and reasoning with uncertain propositions. Using terminological knowledge, fuzzy-rules, T-norm based uncertainty calculi and a fuzzy truth maintenance system, plausible inference can be made. The fuzzy truth maintenance system ensures the consistency of truth values of propositions and the assertion processor translates and propagates internal changes.

*This architecture presents some benefits for developing expert systems. First, expert systems can be built which can refer to terminological knowledge and also reason under uncertainty. Second, it allows for representation and reasoning using uncertainty in the assertional component as well as uncertainty in judgmental knowledge. These two features improve the reasoning capability of expert system. Third, terminological knowledge is applied to both deductive and approximate reasoning, i.e., it is reusable. And fourth, the maintainability and explanation capabilities of expert systems could be improved because meanings of terms are explicitly represented and are separated from heuristic knowledge that is used for plausible inferences.*

## REFERENCES

- [1] G. Abrett and M.H.Burstein, "The KREME Knowledge Editing Environment," *International Journal of Man-Machine Studies*, 27, 103-126, 27, April, 1987.
- [2] R. L. Bates and R. MacGregor, "Knowledge Representation in NIKL," in *AI Tools and Techniques*, editor Mark Richer. Norwood,NJ: Ablex Publishing Corporation, 1989, Chapter 6, pp 183-196.
- [3] D. G. Bobrow and T. Winograd, "An Overview of KRL, a Knowledge Representation Language," in *Readings in Knowledge Representation*, editors R. G. Brachman and H. J. Levesque. Los Altos, CA: Kaufmann Publishers, pp 263-286, 1985.
- [4] P. P. Bonissone, "Summarizing and Propagating Uncertain Information with Triangular Norms," *International Journal of Approximate Reasoning* ,1. Amsterdam, Holland: Elsevier Publishing B.V., pp 71-101, 1987.
- [5] P. P. Bonissone, S. Gans and K. S. Decker, "RUM: A Layered Architecture for Reasoning with Uncertainty," In *Proceedings of the 10th International Joint Conference on Artificial Intelligence, AAAI*, pp 891-898, August 1987.
- [6] R.J.Brachman, "What is-a is and isn't: An Analysis of Taxonomic Links in Semantic Networks," *Computer*, 16(10):30-36, October 1983.

- [7] R.J.Brachman and J.G.Schmolze, "An Overview of the KL-ONE Knowledge Representation System," *Cognitive Science* 9, 171-216, August 1985.
- [8] C. L. Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence*, 19(1982)17-37.
- [9] J. Heinsohn and B. Owsnicki-Klewe, "Probabilistic Inheritance and Reasoning in Hybrid Knowledge Representation System," in *Proceedings of the 12th German Workshop on Artificial Intelligence (GWAI-88)*, Berlin:Springer, pp 51-60, 1988.
- [10] N. S. Lee, Y. L. Grize and K. Dehnad, "Quantitative Models for Reasoning under Uncertainty in Knowledge-Based Systems," *International Journal of Intelligent Systems* Vol II, 15-38(1987).
- [11] R. MacGregor. "A Deductive Pattern Matcher," in *Proceedings of AAAI-88*, 1988.
- [12] P. F. Patel-Schneider, B. Owsnicki-Klewe, A.Kobsa, N. Guarino, R. MacGregor, W. S. Mark, D. McGuinness, B. Nebel, A. Schmiedel and J. Yen, "Report on the Workshop on Term Subsumption Languages in Knowledge Representation," *AI Magazine*, 11:16-23, 1990.
- [13] J. G. Schmolze and T. A. Lipkis, "Classification in the KL-ONE Knowledge Representation System," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence. IJCAI*, 1983.
- [14] L. Shastri, "Default Reasoning in Semantic Networks: A Formalization of Recognition and Inheritance," *Artificial Intelligence*, 39(3):283-355, July, 1989.
- [15] L. Shastri and J. A. Feldman, "Evidential Reasoning in Semantic Networks," in *Proceedings of IJCAI-85*, Los Angeles, California, pp 18-23, August 1985.
- [16] N. Vaidya, "Integrating Approximate Reasoning with Term Subsumption Systems", Masters Thesis, Department of Computer Science, Texas A&M University, College Station, Texas, August 1991.
- [17] W. A. Woods, "What's in a Link: Foundations for Semantic Network," *Representation and Understanding: Studies in Cognitive Science*, D. G.Bobrow and A. M. Collins, editors. Academic Press, pp 35-82, 1975.
- [18] J. Yen, "Generalizing Term Subsumption Languages to Fuzzy Logic", in *Proceedings of the 12th International Joint conference on Artificial Intelligence (IJCAI-91)*, pp 472-477, Sydney, Australia, August 1991.
- [19] J. Yen and Piero P. Bonissone, "Extending Term Subsumption Systems for Uncertainty Management", in *Proceedings of 6th Conference on Uncertainty in Artificial Intelligence*, pp 468-473, July 1990.
- [20] J. Yen, R Neches and R. MacGregor, "Classification-based Programming: A Deep Integration of Rules and Frames," Technical Report ISI/RR-88-213, USC / Information Sciences Institute, Marina del Rey, CA, 1989.
- [21] J. Yen, R. Neeches and R. MacGregor, "Integrating Term Subsumption Systems and Production Systems," *IEEE Transactions on Knowledge and Data Engineering*, Vol3, No.1, pp 25-32, March 1991.
- [22] L.A.Zadeh, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *Approximate Reasoning in Expert Systems*, M.M.Gupta, A.Kandell, W.Bandler, J.B.Kiszka (editors), (North-Holland): Elsevier Publishers B.V., pp 3-31, 1985.

# Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions p. 10

Raghu Krishnapuram, James M. Keller, and Yibing Ma  
 Department of Electrical and Computer Engineering  
 University of Missouri, Columbia, MO 65211  
 phone: (314) 882 - 7766  
 fax: (314) 882 - 0397  
 eceraghu@umcvmb.missouri.edu

## ABSTRACT

Properties of objects and spatial relations between objects play an important role in rule-based approaches for high-level vision. The partial presence or absence of such properties and relationships can supply both positive and negative evidence for region labeling hypotheses. Similarly, fuzzy labeling of a region can generate new hypotheses pertaining to the properties of the region, its relation to the neighboring regions, and finally, the labels of the neighboring regions. In this paper, we present a unified methodology to characterize properties and spatial relationships of object regions in a digital image. The proposed methods can be used to arrive at more meaningful decisions about the contents of the scene.

## 1. Introduction

The determination of properties of image regions and spatial relationships among regions is critical for higher level vision processes involved in tasks such as autonomous navigation, medical image analysis, or more generally, scene interpretation. In a rule-based system to interpret outdoor scenes, typical rules may be

IF a REGION is *THIN AND SOMEWHAT NARROW*  
 THEN it is a ROAD  
 IF a REGION is *RATHER BLUE AND HOMOGENEOUS AND*  
 IF THE REGION is *ABOVE* a TREE REGION  
 THEN it is SKY

Although humans may have an intuitive understanding of words such as "thin" and "narrow", such concepts defy precise definitions, and they are best modeled by fuzzy sets. Similarly, humans are able to quickly ascertain the spatial relationship between two objects, for example "B is above A", but this has turned out to be a rather elusive task for automation. When the objects in a scene are represented by crisp sets, the all-or-nothing definition of the subsets actually adds to the problem of generating such relational descriptions. It is our belief that definitions of properties and spatial relationships based on fuzzy set theory, coupled with a fuzzy segmentation will yield realistic results.

Rosenfeld[1-3] defined many terms used in the analysis of spatial properties of objects represented by fuzzy sets. Pal has defined similar geometric attributes (such as index of area coverage) and have developed low- and intermediate-level algorithms based on such attributes [4]. Dubois and Jaulent[5] generalized Rosenfeld's definitions using both fuzzy set and evidence theories.

Approximate spatial relation analysis has also attracted the attention of many researchers in the past several years. In many situations, precise description of relations among objects may be too complex and computationally too expensive. Approximate spatial relation analysis provides a natural way to solve real world problems with a reasonable cost. Freeman[6] was among the first to recognize that the nature of spatial relations among objects requires that they be described in an approximate(fuzzy) framework. Retz[7] has examined the intrinsic, deictic, and extrinsic use of spatial prepositions and has designed a system called CITYTOUR that answers natural language questions about spatial relations between objects in a scene and about the movement of objects. More recently, Dutta[8] has applied fuzzy inference and used a generalization of Warshall's algorithm to reason about object spatial positions and motion. However, modeling spatial relations among image objects is not addressed in any of these papers. Keller and Sztandera[9] addressed the problem of defining some spatial relationships between fuzzy subsets of an image by using dominance relations of projections of the regions onto coordinate axes.

In this paper, we propose direct methods to analyze properties of fuzzy image regions and spatial relations between fuzzy image regions quantitatively. The methods use membership functions generated by a fuzzy segmentation algorithm such as the fuzzy C-means algorithm [10]. The partition generated by the segmentation process is assumed to define  $C$  fuzzy subsets, one representing each object or region in the image. We express the membership function of each object in terms of its  $\alpha$ -cut level sets and perform all computations on the level sets to obtain spatial properties of objects. We determine the relative positions of the level sets based on certain measurements on the elements of the level sets, and then we map the aggregated angle measurements into the interval [0,1] using suitable membership functions to define spatial relations between regions as fuzzy sets over the domain of  $\alpha$ -levels.

In section 2, we describe methods to generate fuzzy subsets that describe the objects (regions) in the image. In section 3, we review the existing methods to compute geometric properties and attributes of fuzzy image regions, and suggest how these methods can be easily extended to nongeometric properties and attributes. In section 4, we describe our method to compute membership functions for spatial relations between fuzzy regions. The relations include LEFT OF, RIGHT OF, ABOVE, BELOW, BEHIND, IN FRONT OF, NEAR, FAR, INSIDE, OUTSIDE, and SURROUND. In section 5, we show some typical experimental results of attribute and spatial relation analysis involving fuzzy image regions. Section 6 contains the summary and conclusions.

## 2. Generation of Fuzzy Subsets to Describe Objects in the Image

Prewitt [11] was the first to suggest that the results of segmentation be fuzzy subsets of the image. In a fuzzy representation of an image, each object is represented by a fuzzy region  $F$ , where  $F$  is defined over a referential set  $\Omega$ . Here,  $\Omega$  is the domain over which the image function is defined. In this paper, we are mainly concerned with the discrete case, and hence  $\Omega$  may be considered as a two-dimensional array. The membership function  $\mu_F$  for the object is defined by:  $\mu_F: \Omega \rightarrow [0,1]$ . Each point  $x = (x,y)$  in  $\Omega$  is assigned a membership grade  $\mu_F(x)$ . It is further convenient to represent this region in terms of  $\alpha$ -cut level sets  $F^\alpha$  as:  $F^\alpha = \{x \mid \mu_F(x) \geq \alpha\}$ , where  $\alpha \in [0,1]$ . In a real image, the number of membership values present is finite, and can be made quite small by quantizing the values. Hence, they can be enumerated as  $1 = \alpha_1 > \alpha_2 > \dots > \alpha_n$ . In what follows,  $\alpha_{n+1}$  will be assumed to be 0. The level sets are nested, i.e.,  $F^{\alpha_i} \supseteq F^{\alpha_j}$  for  $\alpha_i < \alpha_j$ .

In addition, for each  $\alpha$ -cut level set  $F^{\alpha_i}$ , we can associate a basic probability assignment  $m(F^{\alpha_i})$ , where  $m(F^{\alpha_i})$  satisfies:  $\sum m(F^{\alpha_i}) = 1$  [5].

One popular method for assigning multi-class membership values to pixels, for either segmentation or other types of processing, is the fuzzy C-means(FCM) algorithm[10]. The normalization of the memberships across classes in that approach sometimes leads to counter-intuitive memberships. The partition generated by FCM may also be sensitive to noisy features and outliers. Also, the number of classes must be specified for the algorithm to run. The possibilistic C Means algorithm and the unsupervised clustering algorithms proposed by the authors overcome many of these problems [12-13].

### 3. Properties and Attributes of Fuzzy Regions

There are many ways to describe properties and attributes of an object. Properties and attributes of fuzzy image regions may be both geometric and non-geometric. In practical applications, some of the geometric properties that are frequently encountered are area, height, extrinsic diameter, intrinsic diameter, roundness, elongatedness, etc. [3]. Examples of non-geometric properties are brightness, color and texture. We now briefly summarize some geometric properties and their definitions from the existing literature [3].

The area of a fuzzy region  $F$  is defined as the scalar cardinality of  $F$ , i.e.,

$$a(F) = \sum_{x \in \Omega} \mu_F(x) \quad (1)$$

The height  $h$  of a fuzzy region  $F$  along the direction  $u$  is defined as

$$h_u(F) = \sum_v \max \mu_F(u,v) \quad (2)$$

where  $v$  is the direction perpendicular to  $u$ . Rosenfield[2] defined the extrinsic diameter of a fuzzy region  $F$  as

$$E(F) = \max_u h_u(F) \quad (3)$$

where  $h_u$  is defined as above. The geometric property "elongatedness" may be defined in terms of the ratio of the minor extrinsic diameter and the major extrinsic diameter, i.e.,

$$\mu_{EL}(F) = 1 - \frac{\max_u h_u(F)}{E(F)} \quad (4)$$

Conversely, the geometric property "roundness" may be defined as the complement of "elongatedness".

The geometric properties of objects can also be defined with respect to  $\alpha$ -cut level sets [5].

Assume we have nested  $\alpha$ -cut level sets  $\{F^{\alpha_1} \subseteq F^{\alpha_2} \subseteq \dots \subseteq F^{\alpha_n}\}$ , with a basic probability assignment  $m$  defined by

$$m(F^{\alpha_i}) = \alpha_i - \alpha_{i+1}, \quad (5)$$

where  $\alpha_1 = 1$ , and  $\alpha_{n+1} = 0$ . Then, for any  $x \in F^{\alpha_i} - F^{\alpha_{i-1}}$ ,  $\mu_F(x) = \alpha_i$ . The expected value of a property  $P(F)$ , may be measured as:

$$\bar{P}(F) = \sum_{i=1}^n m(F^{\alpha_i}) P(F^{\alpha_i}) = \sum_{i=1}^n (\alpha_i - \alpha_{i+1}) P(F^{\alpha_i}). \quad (6)$$

$\bar{P}(F)$  is the expected value of  $P(F)$ . Since  $F^{\alpha_i}$  is a crisp set, traditional techniques can be used to compute  $P(F^{\alpha_i})$ . For example, one may simply average the value of the property in the crisp region denoted by  $F^{\alpha_i}$  to obtain  $\bar{P}(F^{\alpha_i})$ . Dubois and Jaulent proved [5] that  $a(F)$  is the expected area  $\bar{a}(F)$  and the height of  $F$  along the  $y$ -axis is equal to the expected height along the  $y$ -axis of  $F$ . For the expected extrinsic diameter, the following inequality is true,

$$\bar{e}(F) \geq E(F). \quad (7)$$

#### 4. Spatial Relations between Fuzzy Regions

The primitive spatial relations between two objects are [6]: 1) LEFT OF, 2) RIGHT OF, 3) ABOVE, 4) BELOW, 5) BEHIND, 6) IN FRONT OF, 7) NEAR, 8) FAR, 9) INSIDE, 10) OUTSIDE, and 11) SURROUND. In the following, we present detailed definitions and methods for computation of memberships for some of the relations.

We define the relations as fuzzy sets over the universe of discourse of the  $\alpha$ -cut values  $\{\alpha_1, \dots, \alpha_n\}$ . The general approach we use is as follows. Let  $A$  and  $B$  be two fuzzy sets defined on  $\Omega$ . At each  $\alpha$ -cut value  $\alpha_i$ , we compute the membership value for " $A^{\alpha_i}$  RELATION  $B^{\alpha_i}$ " based on certain measurements  $\gamma$  on the relative positions of the pairs of elements  $(a, b)$ ,  $a \in A^{\alpha_i}$  and  $b \in B^{\alpha_i}$ . These measurements are aggregated for all pairs elements to give an aggregated measurement  $\gamma_i$ . The membership value for " $A^{\alpha_i}$  RELATION  $B^{\alpha_i}$ " denoted by  $\mu_{A\_REL\_B}(\alpha_i)$  is then computed by evaluating a membership function  $\mu_{REL}$  at  $\gamma_i$ . We are currently investigating methods to compute the overall membership for "A RELATION B", once the memberships for " $A^{\alpha_i}$  RELATION  $B^{\alpha_i}$ " is computed for all  $\alpha_i$ . This may be achieved via a fuzzy aggregation operator, or from a method suggested by Dubois and Jaulent [5]. Ternary relations (such as "A IS BETWEEN B and C") can also be handled in a similar fashion.

In the following, we discuss the membership functions  $\mu_{REL}$  for some of the relations listed above in more detail. In Section 5, we show examples of membership computations for a variety of relations in different situations.

##### 4.1 LEFT OF

Human perception of spatial positions between two objects is closely related to angular information. For example, one would search a sector area subtending an angle of approximately  $180^\circ$  left of oneself to find an object that supposedly lies to one's left. Here, the distance between the person and the object is relatively unimportant. Based on this observation, we define most of the spatial relations in terms of angular measurements.

Suppose we have two points  $A$  and  $B$ . Denote  $AB$  as the line connecting  $A$  and  $B$ . Let  $\theta$  be the angle between  $AB$  and the horizontal line, as shown in Figure 1. The membership function for "A is to the LEFT of B" may be defined as a function of  $\theta$  as

$$\mu_{\text{LEFT}}(\theta) = \begin{cases} 1 & |\theta| < a\pi/2 \\ \frac{\pi/2 - |\theta|}{\pi/2(1-a)} & a\pi/2 \leq |\theta| \leq \pi/2 \\ 0 & |\theta| > \pi/2 \end{cases} \quad (8)$$

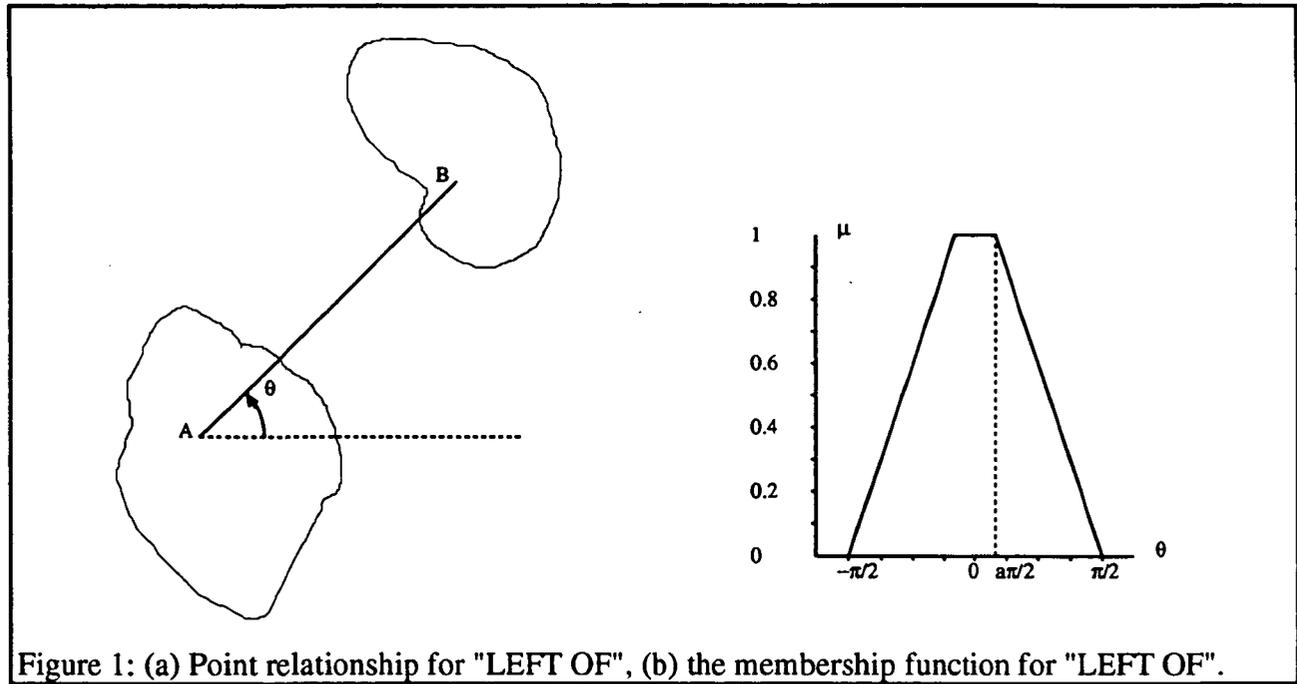


Figure 1: (a) Point relationship for "LEFT OF", (b) the membership function for "LEFT OF".

A large value for  $a$  tends to give an optimistic result, and a small value would give a pessimistic result. Other symmetric functions may also be used to define  $\mu_{\text{LEFT}}$ . The definition in (8) assumes that  $A$  and  $B$  are points. If they are two fuzzy regions, the angles described above are computed and averaged for all pairs of elements  $(a,b)$ ,  $a \in A^{\alpha_i}$  and  $b \in B^{\alpha_i}$ . The membership grade for " $A^{\alpha_i}$  LEFT OF  $B^{\alpha_i}$ " is obtained by mapping the averaged angle  $\theta_0$  through the membership function defined in (8).

#### 4.2 RIGHT OF, ABOVE, BELOW, BEHIND, IN FRONT OF

These relations may be calculated similar to the relation "LEFT OF", using aggregated values of angles made by lines joining pairs of points along with a corresponding trapezoidal membership function. Due to the symmetry in our definitions, the membership grade for " $A$  is to the LEFT OF  $B$ " is the same as that for " $B$  is to the RIGHT OF  $A$ ". The symmetric property also applies to the relation pairs "ABOVE" - "BELOW", and "BEHIND" - "IN FRONT OF". It is to be noted that some of the terms mentioned above actually contain three dimensional information. As images are usually represented in a 2D space, some of these terms may not have any meaning.

#### 4.3 INSIDE, OUTSIDE

For two level sets  $A^{\alpha_i}$  and  $B^{\alpha_i}$ , the membership function for the spatial relation " $A^{\alpha_i}$  is INSIDE  $B^{\alpha_i}$ " may be defined as,

$$\mu_{INSIDE}(\alpha_i) = \frac{|A^{\alpha_i} \cap B^{\alpha_i}|}{|B^{\alpha_i}|}, \quad (9)$$

where  $|A^{\alpha_i} \cap B^{\alpha_i}|$  and  $|B^{\alpha_i}|$  are the cardinalities of the level sets  $A^{\alpha_i}$  and  $B^{\alpha_i}$  respectively. In a digital image, cardinality of a set is the number of pixels that belong to the corresponding level set. The membership function for "A is OUTSIDE B" can be defined as the complement of that for "A is INSIDE B".

#### 4.4 SURROUND

If we assume that all the level sets of an object are connected regions, at each  $\alpha$ -cut level set, we can find two lines  $l_1$  and  $l_2$  for each point in  $B^{\alpha_i}$ , as shown in Figure 2. Let  $\theta$  denote the angle between the two lines as shown. The membership grade for " $A^{\alpha_i}$  SURROUNDS  $B^{\alpha_i}$ " may be calculated by first computing the average  $\theta_i$  of the angles  $\theta$  for every element of  $B^{\alpha_i}$ , and then applying the following mapping at  $\theta = \theta_i$ .

$$\mu_{SURROUND}(\theta) = \begin{cases} 1 & \theta > (2-a)\pi \\ \frac{\pi - \theta}{\pi(1-a)} & \pi \leq \theta \leq (2-a)\pi \\ 0 & \theta < \pi \end{cases} \quad (10)$$

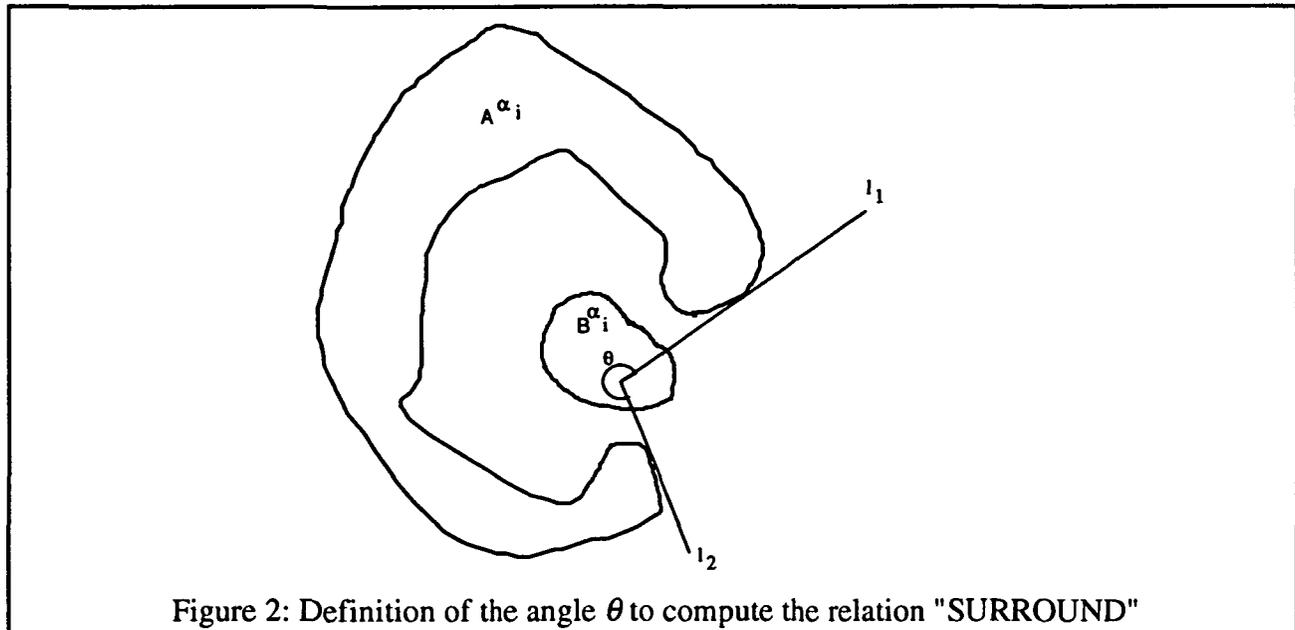


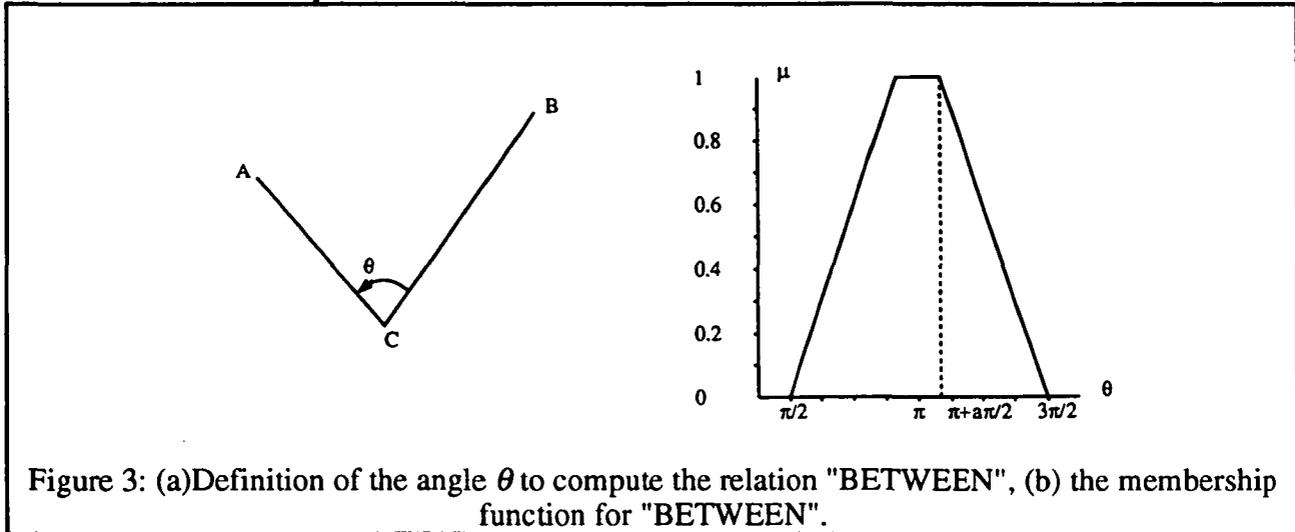
Figure 2: Definition of the angle  $\theta$  to compute the relation "SURROUND"

#### 4.6 Spatial Relations among Objects (BETWEEN)

Consider three points  $A$ ,  $B$  and  $C$  as shown in Figure 3. The membership value for " $C$  is BETWEEN  $A$  and  $B$ " may be defined using a trapezoidal shape as shown in Figure 3.

$$\mu_{BETWEEN} = \begin{cases} 1 & |\theta - \pi| \leq a\pi/2 \\ \frac{\pi/2 - |\pi - \theta|}{(1-a)\pi/2} & a\pi/2 \leq |\theta - \pi| \leq \pi/2 \\ 0 & |\theta - \pi| \geq \pi/2 \end{cases} \quad (11)$$

The membership value for " $C\alpha_i$  is BETWEEN  $A\alpha_i$  and  $B\alpha_i$ " may be computed by evaluating the membership function in (11) at  $\theta = \theta_i$ , where  $\theta_i$  is the average of all the angles between lines  $(a,c)$  and  $(c,b)$ , where  $a \in A\alpha_i$  and  $b \in B\alpha_i$  and  $c \in C\alpha_i$ . Other spatial relations among objects may be defined in a similar way.



## 5. Examples of Spatial Relation Analysis

Extensive simulations were conducted before we applied the proposed methods to real images. In the simulations, we chose objects with various membership function distributions, such as Gaussian shapes, triangular shapes, and exponential shapes. Relative positions and sizes of objects were also altered to observe the influence on the resulting membership functions for spatial relations. We first present two typical examples from our simulation experiments. We then present examples involving real images.

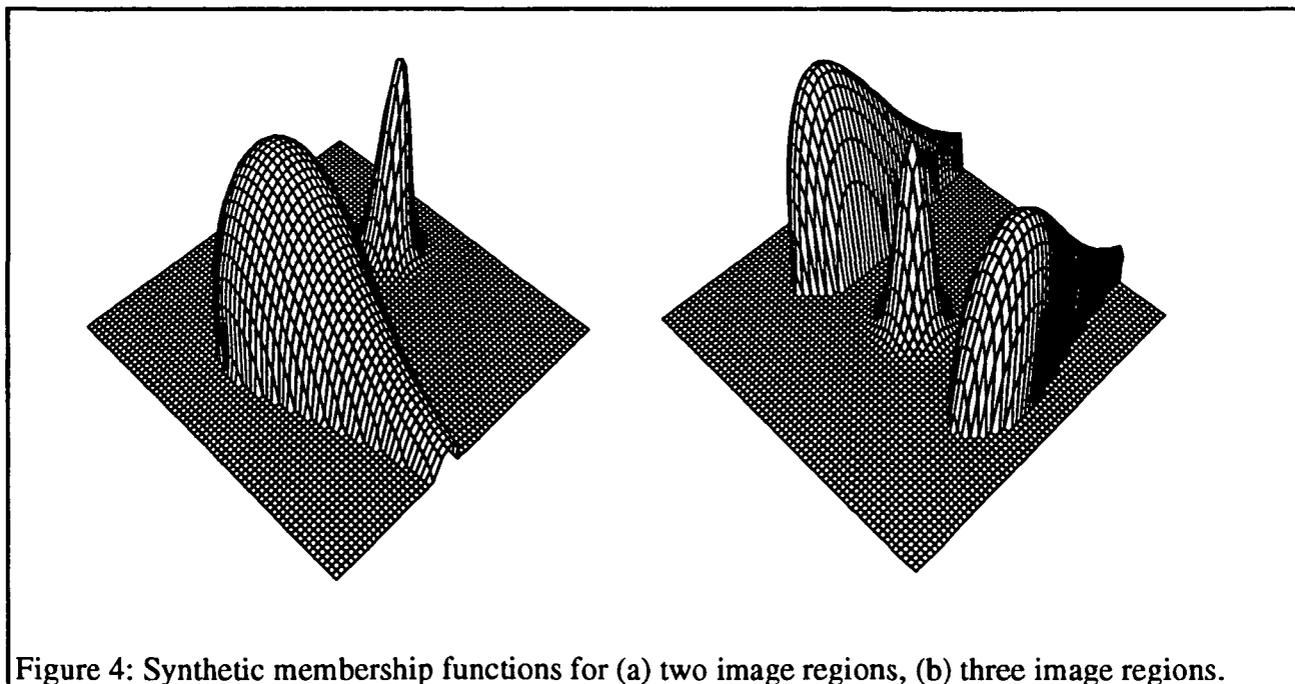


Figure 4(a) shows the fuzzy membership functions of two objects in an image and Figure 4(b) shows the fuzzy membership functions of three objects. The z-axis represents the membership grades for the objects. In Figure 4(a), the large object lies below the small object. In Figure 4(b), the small object lies in between the two large ones. It is to be noted that the membership functions for the large objects are not symmetric about the peak value. The membership grades of two spatial relations in the two images are shown in Figure 5. From Figure 5(a), we notice that at small  $\alpha$ -cut levels, object *A* (large one) lies somewhat to the right of *B*. However, it is definitely below *B*. Therefore we have a reasonably high membership grade of about 0.85 for small  $\alpha$ -cut levels for the relation "A is BELOW B". As the  $\alpha$ -cut level increases, object *A* shrinks more and more to a position perfectly below object *B*. This results in a gradual increase of the membership grades to one. Similarly, in Figure 5(b), we initially have a low membership grade for A is BETWEEN B and C and as the  $\alpha$ -cut level rises, object *A*'s position is more BETWEEN B and C. Therefore the membership grades related to the spatial relation also increases accordingly.

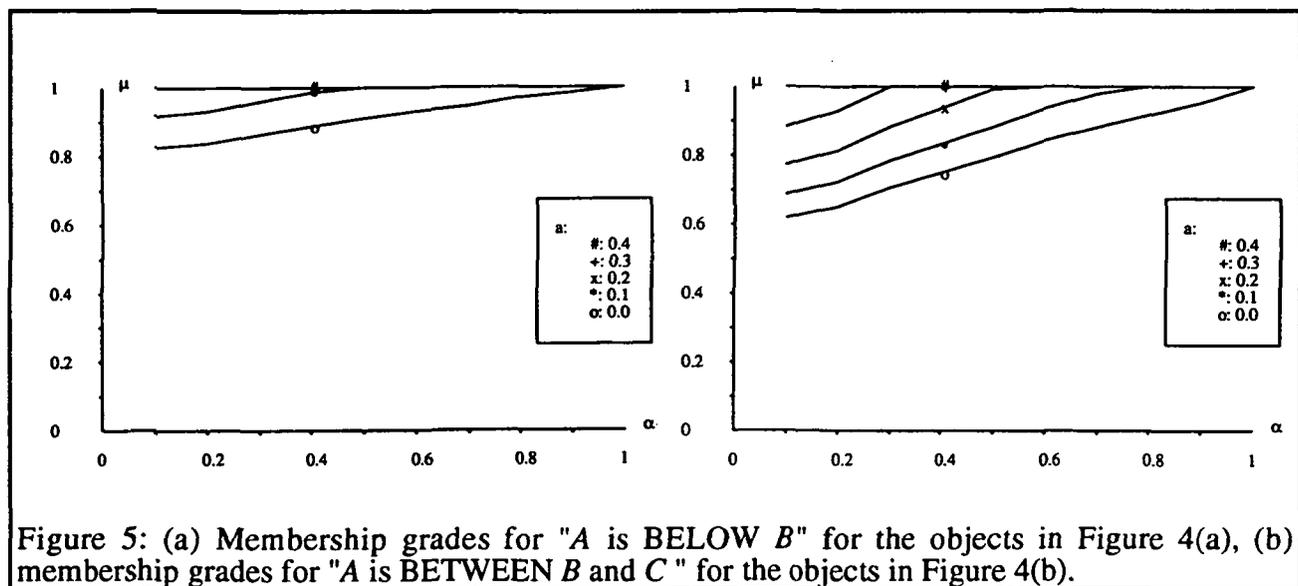


Figure 5: (a) Membership grades for "A is BELOW B" for the objects in Figure 4(a), (b) membership grades for "A is BETWEEN B and C" for the objects in Figure 4(b).

We next present some typical examples of our experimental results with real images. Figure 6 shows a  $256 \times 256$  image of a natural scene as well as its segmentation by the Gustafson-Kessel algorithm [12]. (The closest crisp partition is shown.) A texture feature (homogeneity) and three color features (red, green, and blue) were used to perform the segmentation. The segmented image shows three main objects: sky, road, and trees. Figures 7(a) and 7(b) show the membership grades for the "correct" spatial relation "The sky is ABOVE the trees" and the "false" spatial relation "The sky is to the LEFT of the trees". In the image, a considerable portion of the sky is actually lower than the tree region. However, our method still generated high membership grades for the true hypothesis. This shows that our method of aggregating relations is very effective in capturing the intuitively correct overall spatial relation between regions. The membership grades for "The sky is to the LEFT of the trees" is low, as expected. Figure 7(e) shows the plot of the membership function for the ternary relation "The Trees are BETWEEN the SKY and the ROAD", for the segmentation shown in Figure 6(b). As expected, our method generated high membership grades for this correct hypothesis.

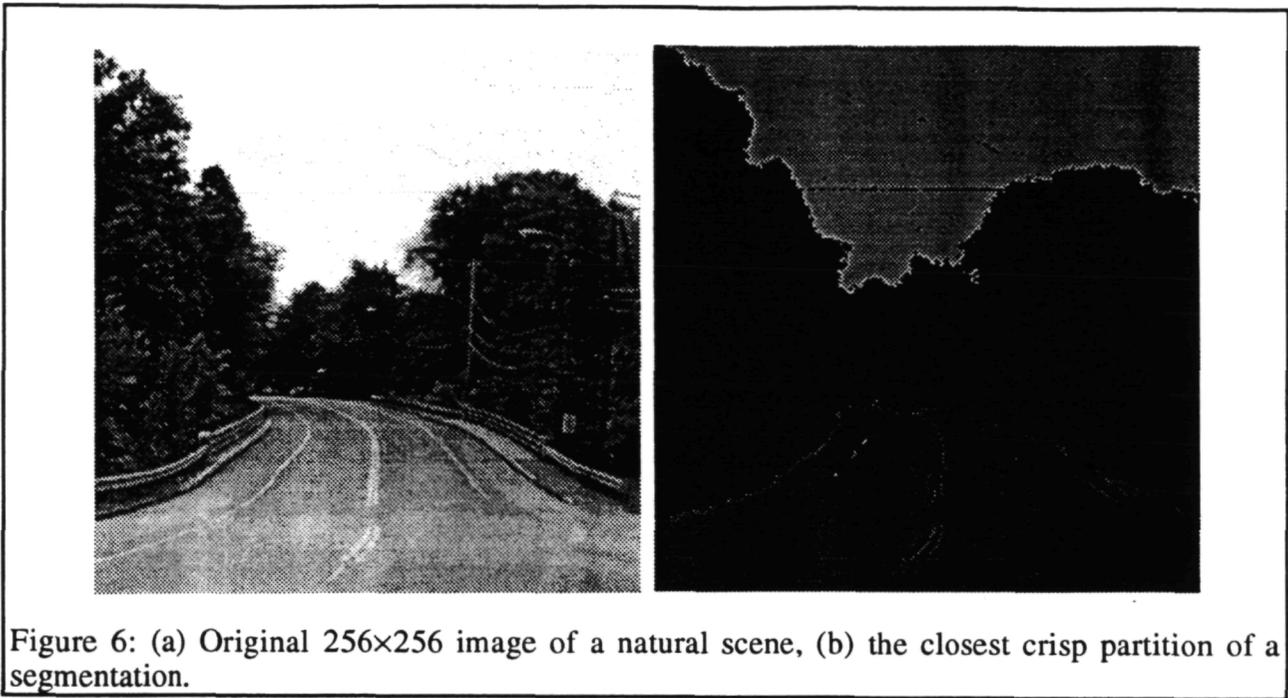


Figure 6: (a) Original 256x256 image of a natural scene, (b) the closest crisp partition of a segmentation.

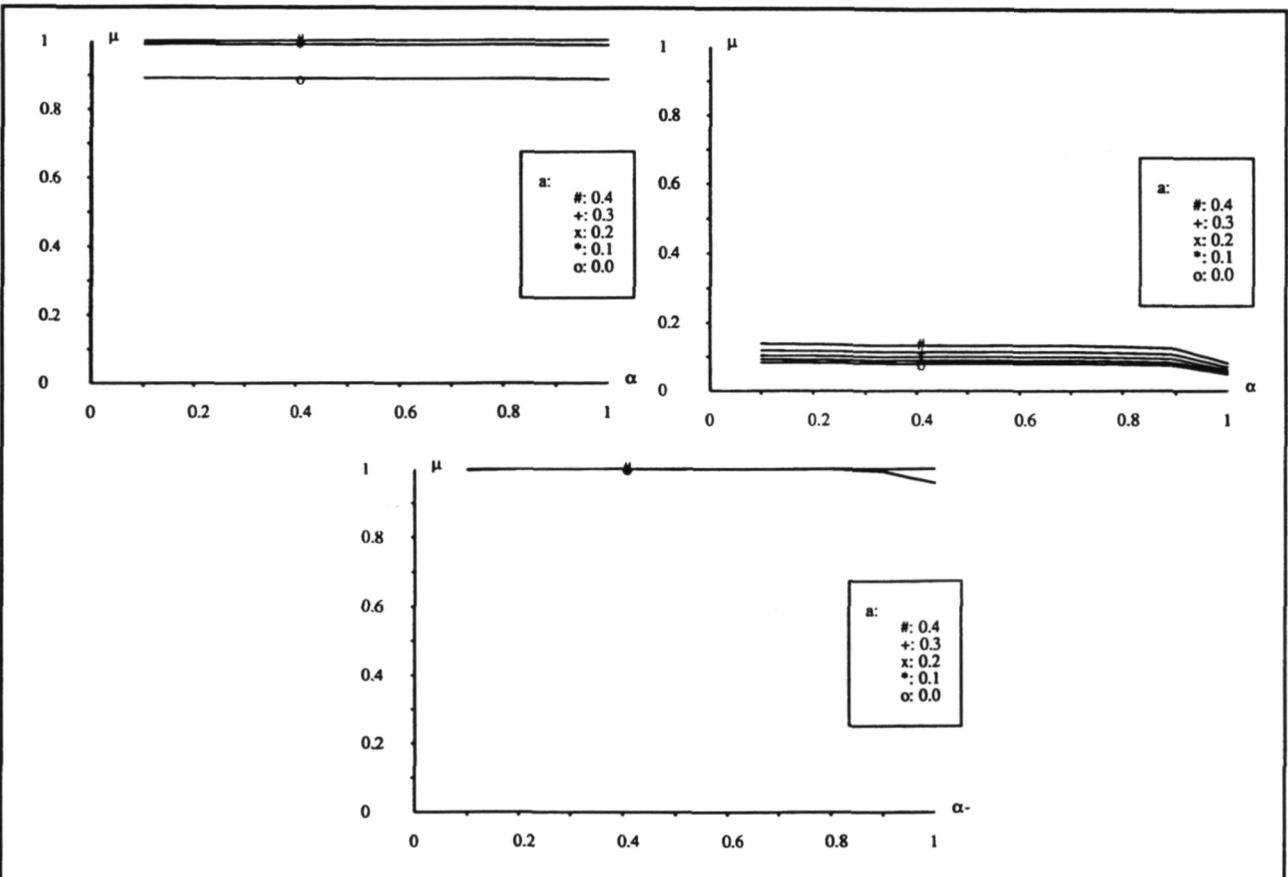


Figure 7: The membership grades for (a) the "correct" spatial relation "The sky is ABOVE the trees" and (b) the "false" spatial relation "The sky is to the LEFT of the trees", and (c) the relation "The Trees are BETWEEN the SKY and the ROAD".

## 6. Conclusions

In this paper, introduce a new approach to analyze spatial relations between objects and among objects. In this approach, objects in the image are viewed as fuzzy regions, and spatial relations between fuzzy regions are viewed as membership functions (possibility distributions) defined over the set of  $\alpha$ -cut sets of the fuzzy regions. This  $\alpha$ -cut approach is similar to the approach introduced by Dubois and Jaulent ; and hence is consistent with the existing definition for the geometric properties of spatial regions. Since the properties and spatial relations are defined over the set of  $\alpha$ -cut sets, efficient algorithms to compute these relations can be devised, and these algorithms save considerable computation time. The methodology expressed in the paper can be widely used in such areas as image understanding, rule-based reasoning, and motion analysis.

## 7. References

1. A. Rosenfeld, "Fuzzy digital topology", *Information and Control*, Vol. 40, pp. 76-87, 1979.
2. A. Rosenfeld, "The fuzzy geometry of image subsets", *Pattern Recognition Letters*, Vol. 2, pp. 311-317, 1984.
3. A. Rosenfeld, "Fuzzy Geometry: An Overview", *Proceedings of the First IEEE Conference on Fuzzy Systems, March 1992*, San Diego, pp. 113-118.
4. S. K. Pal, "Fuzziness, image information and scene analysis", in *An Introduction to fuzzy logic applications in intelligent systems*, R. R. Yager and L. A. Zadeh (Ed.), Kluwer Academic, Norwell MA, 1992, pp. 147-184.
5. D. Dubois and M.-C. Jaulent, "A general approach to parameter evaluation in fuzzy digital pictures", *Pattern Recognition Letters*, Vol. 6, pp. 251-259, 1987.
6. J. Freeman, "The modeling of spatial relations", *Computer Graphics and Image Processing*, vol. 4, pp. 156-171, 1975.
7. G. Retz-Schmidt, "Various views on spatial prepositions", *AI Magazine*, Summer, 1988, pp. 95-105.
8. S. Dutta, "Approximate spatial reasoning: integrating qualitative and quantitative constraints", *International Journal of Approximate Reasoning*, vol. 5, pp. 307-331, 1991.
9. J. Keller and L. Sztandera, "Spatial Relationships Among Fuzzy Subsets of an Image", *Proc. Int. Symposium in Uncertainty Modeling and Analysis*, Univ. of Maryland, December, 1990, pp. 207-211.
10. J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, 1981.
11. J. M. Prewitt, "Object enhancement and extraction", in *Picture Processing and Psychopictorics*, B.S. Lipkin and A. Rosenfeld (Eds.), Academic Press, New York, 1970, pp. 75-149.
12. R. Krishnapuram, and C.-P. Freg, "Fuzzy algorithms to find linear and planar clusters and their applications", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Hawaii, June, 1991. pp.426-431.
13. R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering", submitted to the *IEEE Transactions on Fuzzy Systems*.

# A Fuzzy Clustering Algorithm to Detect Planar and Quadric shapes p-10

Raghu Krishnapuram, Hichem Frigui and Olfa Nasraoui  
Department of Electrical and Computer Engineering  
University of Missouri  
Columbia, MO 65211, USA

## ABSTRACT

In this paper, we introduce a new fuzzy clustering algorithm to detect an unknown number of planar and quadric shapes in noisy data. The proposed algorithm is computationally and implementationally simple, and overcomes many of the drawbacks of the existing algorithms that have been proposed for similar tasks. Since the clustering is performed in the original image space, and since no features need to be computed, this approach is particularly suited for sparse data. The algorithm may also be used in pattern recognition applications.

## 1. Introduction

Boundary detection and surface approximation are important components of intermediate-level vision. They are the first step in solving problems such as object recognition and orientation estimation. Recently, it has been shown that these problems can be viewed as clustering problems with appropriate distance measures and prototypes [1-4]. Dave's Fuzzy C Shells (FCS) algorithm [1] and the Fuzzy Adaptive C-Shells (FACS) algorithm [4] have proven to be successful in detecting clusters that can be described by circular arcs, or more generally by elliptical shapes. Unfortunately, these algorithms are computationally rather intensive since they involve the solution of coupled nonlinear equations for the shell (prototype) parameters. These algorithms also assume that the number of clusters are known. To overcome these drawbacks we recently proposed a computationally simpler Fuzzy C Spherical Shells (FCSS) algorithm [3] for clustering hyperspherical shells and suggested an efficient algorithm to determine the number of clusters when this is not known. We also proposed the Fuzzy C Quadric Shells (FCQS) algorithm [2] which can detect more general quadric shapes. One problem with the FCQS algorithm is that it uses the algebraic distance, which is highly nonlinear. This results in unsatisfactory performance when the data is not very "clean" [4]. Finally, none of the algorithms can handle situations in which the clusters include lines/planes and there is much noise. To summarize, the existing algorithms to detect quadric shell clusters have one or more of the following drawbacks: i) they are computationally expensive, ii) the distance measure used in the objective function can yield distorted estimates of prototype parameters if the data is not well behaved, iii) they assume that the number of clusters  $C$  is known, iv) their formulations do not allow the degenerate case of lines/planes, and v) they are not very robust in the presence of noise. In this paper, we address these drawbacks in more detail and propose a new algorithm to overcome these drawbacks.

## 2. The Fuzzy C Quadrics Algorithm

Let  $x_j = [x_{j1}, x_{j2}, \dots, x_{jn}]$  be a point in the  $n$ -dimensional feature space. We may define the algebraic (or residual) distance from  $x_j$  to a prototype  $\beta_j$  that resembles a second-degree curve as :

$$\begin{aligned} d_{Qj}^2 &= d_Q^2(x_j, \beta_j) = (p_{i1}x_{j1}^2 + p_{i2}x_{j2}^2 + \dots + p_{in}x_{jn}^2 + p_{i(n+1)}x_{j1}x_{j2} + p_{i(n+2)}x_{j1}x_{j3} + \dots \\ &+ p_{is}x_{j(n-1)}x_{jn} + p_{i(s+1)}x_{j1} + p_{i(s+2)}x_{j2} + \dots + p_{i(s+n)}x_{jn} + p_{i(s+n+1)})^2 \\ &= p_i^T q_j q_j^T p_i = p_i^T M_j p_i \end{aligned} \quad (1)$$

The prototypes  $\beta_j$  are represented by the parameter vectors  $p_i = [p_{i1}, p_{i2}, \dots, p_{ir}]^T$  with  $r = s+n+1 = \frac{n(n+1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}$  components, which define the equation of the curve. The  $M_j$  in (1) are given by

$$M_j = q_j q_j^T, \text{ with } q_j^T = [x_{j1}^2, x_{j2}^2, \dots, x_{jn}^2, x_{j1}x_{j2}, \dots, x_{j(n-1)}x_{jn}, x_{j1}, x_{j2}, \dots, x_{jn}, 1]. \quad (2)$$

We may now minimize the following objective function which is similar to the objective function used in Fuzzy C-Means algorithm (FCM) [6] except for the distance measure.

$$J_Q(B,U) = \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m d_{Qij}^2 = \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m p_i^T M_j p_i, \quad (3)$$

where  $B = (\beta_1, \dots, \beta_C)$ ,  $C$  is the number of clusters,  $N$  is the total number of feature vectors and  $U = [\mu_{ij}]$  is the  $C \times N$  fuzzy  $C$ -partition matrix satisfying the following conditions.

$$\mu_{ij} \in [0,1] \text{ for all } i \text{ and } j, \sum_{i=1}^C \mu_{ij} = 1 \text{ for all } j, \text{ and } 0 < \sum_{j=1}^N \mu_{ij} < N \text{ for all } i. \quad (4)$$

Note,  $J_Q(B,U)$  is homogeneous with respect to  $p_i$ . Therefore, we need to constrain the problem in order to avoid the trivial solution. Some of the possibilities are:

$$(i) p_{i1} = 1, (ii) p_{ir} = 1, (iii) \|p_i\|^2 = 1, \text{ and} \\ (iv) \|p_{i1}^2 + p_{i2}^2 + \dots + p_{in}^2 + \frac{p_{i(n+1)}^2}{2} + \frac{p_{i(n+2)}^2}{2} + \dots + \frac{p_{is}^2}{2}\|^2 = 1. \quad (5)$$

In [4] Dave et al have also proposed a Fuzzy C Quadrics (FCQ) algorithm using constraint (i). This constraint is more restrictive than constraint (iv) used in the FCQS algorithm proposed in [3]. Moreover, the resulting distance measure is not invariant to translations and rotations of the prototypes. Constraints (ii) and (iii) are also not suitable for the same reason. In other words, these constraints make the distance  $d_{Qij}^2$  a function of not just the relative location of point  $x_j$  to curve  $\beta_i$ , but also the actual location and orientation of the curve  $\beta_i$  in feature space, which is undesirable. However, constraint (iv) makes the distance invariant to translations and rotations [5]. Other constraints are also possible, and one of them will be discussed in Section 4. With constraint (iv) the minimization of (3) reduces to an eigenvector problem, and its implementation is straightforward. Minimization with respect to the memberships  $\mu_{ij}$  is similar to the FCM case [6]. It is easy to show that the memberships are updated according to

$$\mu_{ij} = \begin{cases} \frac{1}{\sum_{k=1}^C \left(\frac{d_{Qij}}{d_{Qkj}}\right)^{\frac{2}{m-1}}} & \text{if } I_j = \Phi \\ 0 & i \notin I_j \text{ if } I_j \neq \Phi \\ \sum_{i \in I_j} \mu_{ij} = 1 & \text{if } I_j \neq \Phi \end{cases} \quad (6)$$

where  $I_j = \{i \mid 1 \leq i \leq C, d_{Qij}^2 = 0\}$ . The original FCQS algorithm is summarized below.

#### THE FUZZY C QUADRIC SHELLS (FCQS) ALGORITHM:

Fix the number of clusters  $C$ ; fix  $m$ ,  $1 < m < \infty$ ;  
 Set iteration counter  $l = 1$ ;  
 Initialize the fuzzy  $C$ -partition  $U^{(0)}$  using the FCSS algorithm;  
**Repeat**  
     Compute  $p_i^{(l)}$  for each cluster  $\beta_i$  by minimizing (3) subject to (5);  
     Update  $U^{(l)}$  using (6);  
     Increment  $l$ ;  
**Until** ( $\|U^{(l-1)} - U^{(l)}\| < \epsilon$ );

The FCQS algorithm has the following drawbacks: i) Since the algebraic distance given by (1) is highly nonlinear, the membership assignments are not very meaningful, ii) the constraint in (5) strictly speaking does not allow us to fit linear (or planar) clusters. We now address these drawbacks in more detail and propose modifications of the algorithm to overcome these drawbacks.

### 3. The Modified Fuzzy C Quadric Shells Algorithm

To overcome the problem due to the nongeometric nature of  $d_{Qij}^2$ , one may use the geometric (perpendicular) distance (denoted by  $d_{p;ij}^2$ ) between the point  $x_j$  and the shell  $\beta_i$  given by

$$d_{p_{ij}}^2 = \min \|x_j - z_{ij}\|^2 \text{ such that} \\ (z_{ij}^T A_i z_{ij} + z_{ij}^T v_i + b_i) = 0, \quad (7)$$

where  $z_{ij}$  is a point lying on the quadric curve describing cluster  $\beta_i$ . Using a Lagrange multiplier  $\lambda$ , Equation (7) can be solved for  $z_{ij}$  as

$$z_{ij} = \frac{1}{2} (I - \lambda A_i)^{-1} (\lambda v_i + 2x_j). \quad (8)$$

Substituting (8) in (7) yields a quartic (fourth-degree) equation in  $\lambda$  in the 2-D case, which has at most four real roots  $\lambda_k$ ,  $1 \leq k \leq 4$ . They can be easily computed using the standard solution from mathematical tables. For each real root  $\lambda_k$  so computed, we calculate the corresponding  $(z_{ij})_k$  using (8). Then, we may compute  $d_{p_{ij}}^2 = \min_k \|x_j - (z_{ij})_k\|^2$ .

One can formulate the FCQS algorithm using  $d_{p_{ij}}^2$  as the underlying distance measure [15]. Minimizing the resulting objective function with respect to  $U$  yields an equation identical to (6) where  $d_{Q_{ij}}^2$  is replaced by  $d_{p_{ij}}^2$ . However, minimizing the objective function with respect to the parameters  $p_i$  results in coupled nonlinear equations with no closed-form solution. To overcome this problem, we may assume that we can obtain approximately the same solution by minimizing (3) subject to (5), which will be true if all the feature points lie reasonably close to the hyperquadric shells. This assumption leads to the Modified FCQS (MFCQS) algorithm. Our experimental results show that in the 2-D case the modified FCQS algorithm gives much better results and converges much faster than the original version. In fact, our extensive simulations indicate that the performance of this algorithm is excellent, as long as the data points are all reasonably close to the curves (i. e., as long as the data is not highly scattered), which will be true in most computer vision applications. This may be attributed to the fact that the membership assignment based on the perpendicular distance is more reasonable.

The MFCQS algorithm can also be used to find linear clusters, even though the constraint in (5) forces all prototypes to be of second degree. This is because the algorithm usually fits either two coincident lines (for a single line), or an extremely elongated ellipse (for two parallel lines) or a hyperbola (for two crossing lines). It is quite simple to recognize these situations from the parameters of the prototypes, and when these situations occur, we can simply split such prototypes to a pair of lines after the algorithm converges.

It is to be noted that  $d_{p_{ij}}^2$  has a closed-form solution only in the 2-D case. In higher dimensions, solving for  $d_{p_{ij}}^2$  is not trivial. For example, in the three dimensional case, this results in a sixth degree equation, which needs to be solved iteratively. This makes the algorithm slow. We now propose an alternative formulation of the algorithm to overcome this problem.

#### 4. The Fuzzy C Plano-Quadric Shells Algorithm

When the exact (geometric) distance has no closed-form solution, one of the methods suggested in the literature is to use what is known as the "approximate distance" which is the first-order approximation of the exact distance. It is easy to show [7] that the approximate distance of a point from a curve is given by

$$d_{A_{ij}}^2 = d_{A^2}(x_j, \beta_i) = \frac{d_{Q_{ij}}^2}{|\nabla d_{Q_{ij}}^2|^2} = \frac{d_{Q_{ij}}^2}{p_i^T D_j D_j^T p_i}, \quad (9)$$

where  $\nabla d_{Q_{ij}}^2$  is the gradient of the distance functional

$$p_i^T q = [p_{i1}, p_{i2}, \dots, p_{in}] [x_1^2, x_2^2, \dots, x_n^2, x_1 x_2, \dots, x_{(n-1)} x_n, x_1, x_2, \dots, x_n, 1]^T \quad (10)$$

evaluated at  $x_j$ . In (9) the matrix  $D_j$  is simply the Jacobian of  $q$  evaluated at  $x_j$ .

One can easily reformulate the quadric shell clustering algorithm with  $d_{A_{ij}}^2$  as the underlying distance measure. The objective function to be minimized in this case becomes

$$J_{A(B,U)} = \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m d_{A_{ij}}^2 = \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m \frac{p_i^T M_j p_i}{p_i^T D_j D_j^T p_i}. \quad (11)$$

Unfortunately, the minimization of the resulting objective function with respect to  $p_i$  in general leads to coupled nonlinear equations which can only be solved iteratively. To avoid this problem, we choose the constraints

$$p_i^T \left[ \sum_{j=1}^N (\mu_{ij})^m D_j D_j^T \right] p_i = \sum_{j=1}^N (\mu_{ij})^m, \text{ or } p_i^T G_i p_i = N_i, \quad i = 1, \dots, C, \quad (12)$$

where

$$G_i = \sum_{j=1}^N (\mu_{ij})^m D_j D_j^T \text{ and } N_i = \sum_{j=1}^N (\mu_{ij})^m. \quad (13)$$

The above constraint has been applied in the hard case by Taubin [8] with good results when there is only one curve to be fitted. Our contribution is to extend it to the fuzzy case and to fit  $C$  curves simultaneously. Using (12) and Lagrange multipliers, we may now minimize

$$\begin{aligned} & \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m d_{Aij}^2 - \sum_{i=1}^C \lambda_i (p_i^T G_i p_i - N_i) \\ &= \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m \frac{p_i^T M_j p_i}{p_i^T D_j D_j^T p_i} - \sum_{i=1}^C \lambda_i \left[ \sum_{j=1}^N (\mu_{ij})^m p_i^T D_j D_j^T p_i - \sum_{j=1}^N (\mu_{ij})^m \right] \end{aligned} \quad (14)$$

When most of the data points are close to the prototypes, the memberships  $\mu_{ij}$  will be quite hard (i. e., they will be close to 0 or 1). This assumption is also quite good if we use possibilistic memberships [9] to be discussed in Section 5. This means that when the constraint in (12) is satisfied, we may say that  $p_i^T D_j D_j^T p_i \approx 1$ . In fact, it is easy to show that the condition  $p_i^T D_j D_j^T p_i = 1$  is exactly true for the case of lines/planes and certain quadrics such as circles and cylinders. Hence, we will obtain approximately the same solution if we minimize

$$\sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m p_i^T M_j p_i - \sum_{i=1}^C \lambda_i \left[ \sum_{j=1}^N (\mu_{ij})^m p_i^T D_j D_j^T p_i - \sum_{j=1}^N (\mu_{ij})^m \right]$$

If we assume that the prototypes are independent of each other, then this is equivalent to independently minimizing

$$\begin{aligned} & \sum_{j=1}^N (\mu_{ij})^m p_i^T M_j p_i - \lambda_i \left[ p_i^T \sum_{j=1}^N (\mu_{ij})^m D_j D_j^T p_i - \sum_{j=1}^N (\mu_{ij})^m \right] \\ &= p_i^T F_i p_i - \lambda_i (p_i^T G_i p_i - N_i) \end{aligned} \quad (15)$$

where

$$F_i = \sum_{j=1}^N (\mu_{ij})^m M_j. \quad (16)$$

The solution of (16) is given by the generalized eigenvector problem

$$F_i p_i = \lambda_i G_i p_i, \quad (17)$$

which can be converted to the standard eigenvector problem if the matrix  $G_i$  is not rank-deficient. Unfortunately this is not the case. In fact, the last row of  $D_j$  is always  $[0, \dots, 0]$ . Equation (17) can still be solved using other techniques that use the modified Cholesky decomposition [8], and the solution is computationally quite inexpensive when the feature space is 2-D or 3-D. Another advantage of this constraint is that it can also fit lines and planes in addition to quadrics. Minimization of (11) with respect to the memberships  $\mu_{ij}$  leads to

$$\mu_{ij} = \begin{cases} \frac{1}{\sum_{k=1}^C \left( \frac{d_{Aij}}{d_{Aki}} \right)^{\frac{2}{m-1}}} & \text{if } I_j = \Phi \\ 0 & i \notin I_j \text{ if } I_j \neq \Phi \\ \sum_{i \in I_j} \mu_{ij} = 1 & \text{if } I_j \neq \Phi \end{cases} \quad (18)$$

In the 2-D case,  $d_{Aij}^2$  in the above equation may also be substituted by  $d_{Pij}^2$ . We notice that in practice this gives more rapid convergence. The resulting clustering algorithm, which we call the Fuzzy C Plano-Quadric Shells algorithm, is summarized below.

**THE FUZZY C PLANO-QUADRIC SHELLS (FCPQS) ALGORITHM:**

Fix the number of clusters  $C$ ; fix  $m$ ,  $1 < m < \infty$ ;  
 Set iteration counter  $l = 1$ ;  
 Initialize the fuzzy  $C$ -partition  $U^{(0)}$ ;  
**Repeat**  
     Compute the matrices  $F_i$  and  $G_i$  using (13) and (16)  
     Compute  $p_i^{(l)}$  for each cluster  $\beta_i$  solving (17)  
     Update  $U^{(l)}$  using (18);  
     Increment  $l$ ;  
**Until** ( $\|U^{(l-1)} - U^{(l)}\| < \epsilon$ );

**5. Robust Shell Clustering**

The algorithms discussed above will be sensitive to outlier points even when the objective function based on the approximate distance is minimized. To overcome this problem, we have converted the algorithm to a possibilistic algorithm [9]. This is very easily achieved by updating the memberships according to

$$\mu_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{\frac{1}{m-1}}} \quad (23)$$

instead of (18). In (23), one attractive choice for  $\eta_i$  in practice is the average fuzzy intra-cluster distance given by

$$\eta_i = \frac{\sum_{j=1}^N \mu_{ij}^m d_{ij}^2}{\sum_{j=1}^N (\mu_{ij})^m} \quad (24)$$

Our experimental results show that the resulting algorithm, which we call the Possibilistic C Plano-Quadric Shells (PCPQS) algorithm, is quite robust in the presence of poorly defined boundaries (i. e., when the edge points are somewhat scattered around the ideal boundary curve in the 2-D case and when the range values are not very accurate in the 3-D case). It is also very immune to impulse noise and outliers, as can be seen in the examples presented in Section 7. A possibilistic version of the Modified FCQS algorithm (denoted by MPCQS) was also implemented.

**6. Determination of Number of Clusters**

The number of clusters  $C$  is not known *a priori* in some pattern recognition applications and most computer vision applications. When the number of clusters is unknown, one method to determine this number is to perform clustering for a range of  $C$  values, and pick the  $C$  value for which a suitable validity measure is minimized (or maximized) [10,12]. However this method is rather tedious, especially when the number of clusters is large. Also, in our experiments, we found that the  $C$  value obtained this way may not be optimum. This is because when  $C$  is large, the clustering algorithm sometimes converges to a local minimum of the objective function, and this may result in a bad value for the validity of the clustering, even though the value of  $C$  is correct. Moreover, when  $C$  is greater than the optimum number, the algorithm may split a single shell cluster into more than one cluster, and yet achieve a good value for the overall validity. To overcome these problems, we propose an alternative Unsupervised C Shell Clustering algorithm which is computationally more efficient, since it does not perform the clustering for an entire range of  $C$  values.

Our proposed method progressively clusters the data starting with an overspecified number  $C_{max}$  of clusters. Initially, the FCPQS algorithm is run with  $C=C_{max}$ . After the algorithm converges, spurious clusters (with low validity) are eliminated; compatible clusters are merged; and points assigned to clusters with good validity are temporarily removed from the data set to reduce computations. The FCPQS algorithm is invoked again with the remaining feature points. The above procedure is repeated until no more elimination, merging, or removing occurs, or until  $C=1$ . This algorithm is summarized below.

**THE UNSUPERVISED POSSIBILISTIC C PLANO-QUADRIC SHELLS (UPCPQS) ALGORITHM:**

```

Set  $C = C_{max}$  ; fix  $m$  ,  $1 < m < \infty$ ;
 $CRemoved := 0$ ;  $MergeFlag := EliminateFlag := RemoveFlag := TRUE$ ;
While  $C > 1$  and ( $MergeFlag = TRUE$  or  $EliminateFlag = TRUE$  or
   $RemoveFlag = TRUE$ ) do
   $MergeFlag := EliminateFlag := RemoveFlag := FALSE$ ;
  Perform the PCPQS algorithm with the number of clusters =  $C$ ;
  Eliminate spurious clusters using validity, decrement  $C$  accordingly,
    and set  $EliminateFlag = TRUE$  if any elimination has occurred;
  Merge compatible prototypes among the  $C$  prototypes , update  $C$ ,
    and set  $MergeFlag = TRUE$  if merging has occurred;
  Remove good clusters using validity, update  $C$ , and set  $RemoveFlag = TRUE$  if
    any good clusters are removed;
  Save the remaining clusters' prototypes ;
End While
Replace all the removed feature points back into the data set.
Append the list of remaining clusters' prototypes from the last iteration in the while loop
  to the list of removed clusters' prototypes;
Do
  Perform the PCPQS algorithm with the new  $C$ ;
  Merge compatible prototypes in the prototype list and update  $C$  ;
  Eliminate tiny clusters and decrement  $C$  accordingly;
Until No more merging or elimination takes place;

```

One way to determine if two clusters are compatible (i. e., whether they can be merged), is to estimate the best fit for all the points having a membership greater than an  $\alpha$ -cut in the two clusters. If the validity for the resulting cluster is good, then the two clusters are considered mergeable. The above algorithm also requires a validity measure to discriminate between good and bad clusters. Several cluster validity criteria have been presented in the literature. For example, performance measures based on the memberships in the partition matrix  $U$  have been proposed by some researchers [6,10]. Unfortunately, these are not very effective for shell clusters, since they do not reflect the actual geometric structure of the data set. One possible validity measure we may define is the shell thickness measure, which is simply the sum of the squared errors of the fit for the  $i$ th cluster given by

$$T_i = \sum_{j=1}^N (\mu_{ij})^m \alpha_{p_{ij}}^2 \quad (19)$$

However, it is difficult to estimate a "good" value for this validity measure in noisy conditions. Validity measures may also be defined using hypervolume and density [11,12]. To do this, the distance vector from a feature point to a shell prototype is first defined as  $\delta_{ij} = (x_j - z_i)$ , where  $z_i$  is the closest point on the curve (or surface) to the feature point  $x_j$  in the approximate distance sense. The fuzzy spherical shell covariance matrix is defined by

$$\Sigma_i = \frac{\sum_{j=1}^N (\mu_{ij})^m \delta_{ij} \delta_{ij}^T}{\sum_{j=1}^N (\mu_{ij})^m} \quad (20)$$

Using (15) the fuzzy shell hypervolume and the shell density may be defined as

$$V_i = \sqrt{\det(F_i)}, \quad \text{and} \quad D_i = \frac{S_i}{V_i}, \quad (21)$$

where  $S_i$  is the sum of close members of shell  $\beta_i$  given by

$$S_i = \sum_j \mu_{ij} \quad \text{such that} \quad \delta_i^T \Sigma_i^{-1} \delta_i < 1. \quad (22)$$

However, the above measures are not very reliable because their values can vary widely for good clusters, depending on the sizes of the clusters and noise. They can also be "good" for spurious clusters. Therefore, we have developed a new validity measure for shell clusters based on the idea of curve (surface) density, which is a measure of the number of feature points per unit length (surface area) of the shell cluster. We have also developed methods to estimate the effective curve length (surface area) of the shell clusters when the curves (surfaces) are partial. A more detailed discussion of this validity measure will be the subject of a future paper.

## 7. Experimental Results

Although the algorithms presented in the previous sections are applicable to feature spaces of any dimension, in this paper we present only results of two-dimensional data sets. In all the examples shown in this paper, the UPCPQS algorithm was applied with the fuzzifier  $m = 2$  and  $C_{max} = 25$ . To obtain a good initialization of the fuzzy  $C$ -partition  $U^{(0)}$ , we run the Gustafson-Kessel algorithm with  $m = 1.5$  for a few iterations (which gives an excellent linear approximation of the data) followed by the Fuzzy  $C$  Spherical Shells algorithm [2]. This was observed to give excellent results. The data sets consists of object edges obtained by applying an edge operator to real images. Uniformly distributed noise with an interval of 30 was added to the images to make them noisy. The edge images were then thinned [14] to reduce the number of pixels to be processed. The resulting input images typically had about 2000 points. The PCPQS algorithm still sometimes fits second-degree curves for linear clusters, especially when the data is scattered. Therefore, the algorithm was modified to identify such situations and split such clusters into lines after convergence, as explained in Section 3. In practice, there seems to be very little difference between the PCPQS and MPCQS algorithms in the 2-D case.

Figure 1(a) shows the original noisy image of a box with holes. The edge-detected and thinned image is shown in Figure 1(b). As can be seen, there are many noise points, and the pixel boundaries are not always well-defined. Figure 1(c) shows the result of the UPCPQS algorithm. The final prototypes are shown superimposed on the edge image. The prototypes are virtually unaffected by noise and poor boundaries. Figure 1(d) shows the "cleaned" edge map. This is obtained by plotting the boundaries generated by the prototypes only in locations where there is at least one pixel with a high membership value in a  $3 \times 3$  neighborhood. Figures 2 and 3 show similar results for images with collections of objects of various sizes and shapes.

## 8. Summary

In this paper, we propose a new approach to boundary and surface approximation in computer vision. Current techniques to describe boundaries and surfaces in terms of parametrized or algebraic forms have the following disadvantages: i) Many techniques apply in cases when the boundaries/surfaces belonging to different objects have already been segmented, ii) they look for local structures and use edge following or region growing and hence would be sensitive to local aberrations and deviations in shapes, iii) they are computationally intensive and the memory requirement are high, iv) they require features (such as curvature and surface normals) to be calculated and hence are sensitive to noise and the computed features are inaccurate at boundaries of surfaces, v) most of the feature-based techniques assume dense data and hence are not suitable if the data is sparse or if there are gaps in the data, and vi) some methods are not invariant to rigid transformations. The approach we propose overcomes these drawbacks. If the clustering is performed in the feature space, it can have the disadvantages of high dimensionality, and loss of pixel adjacency information. However, since the proposed methods apply clustering techniques directly to image data, they do not suffer from these disadvantages. Another disadvantage of clustering methods is that the number of clusters has to be known in advance. The proposed approach overcomes this problem by using new cluster validity measures and compatible cluster merging.

Linear and Quadric shapes are not sufficiently general for all computer vision applications. We propose to extend our algorithm to more general shells such as those represented by algebraic curves, or superquadrics. Currently there are no algorithms that simultaneously fit an unknown number of general curves (or surfaces) to noisy and/or scattered data. This includes boundaries and surfaces that are locally very noisy, and boundaries and surfaces that are sparsely sampled. Methods based on feature computation and region growing do not work in these cases.

## 9. References

1. R. N. Dave, "Fuzzy-shell clustering and applications to circle detection in digital images", *International Journal of General Systems*, vol. 16, 1990, pp. 343-355.
2. R. Krishnapuram, H. Frigui, and O. Nasraoui, "New fuzzy shell clustering algorithms for boundary detection and pattern recognition", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, Nov. 1991, pp.458-465.
3. R. Krishnapuram, O. Nasraoui, and H. Frigui, "The fuzzy  $C$  spherical shells algorithm: A new approach", *IEEE Transactions on Neural Networks*, vol. 3, no. 5, September 1992.
4. R. N. Dave and K. Bhaswan, "Adaptive fuzzy  $C$ -shells clustering and detection of ellipses", to appear in the *IEEE Transactions on Neural Networks*, vol. 3, no. 5, 1992.

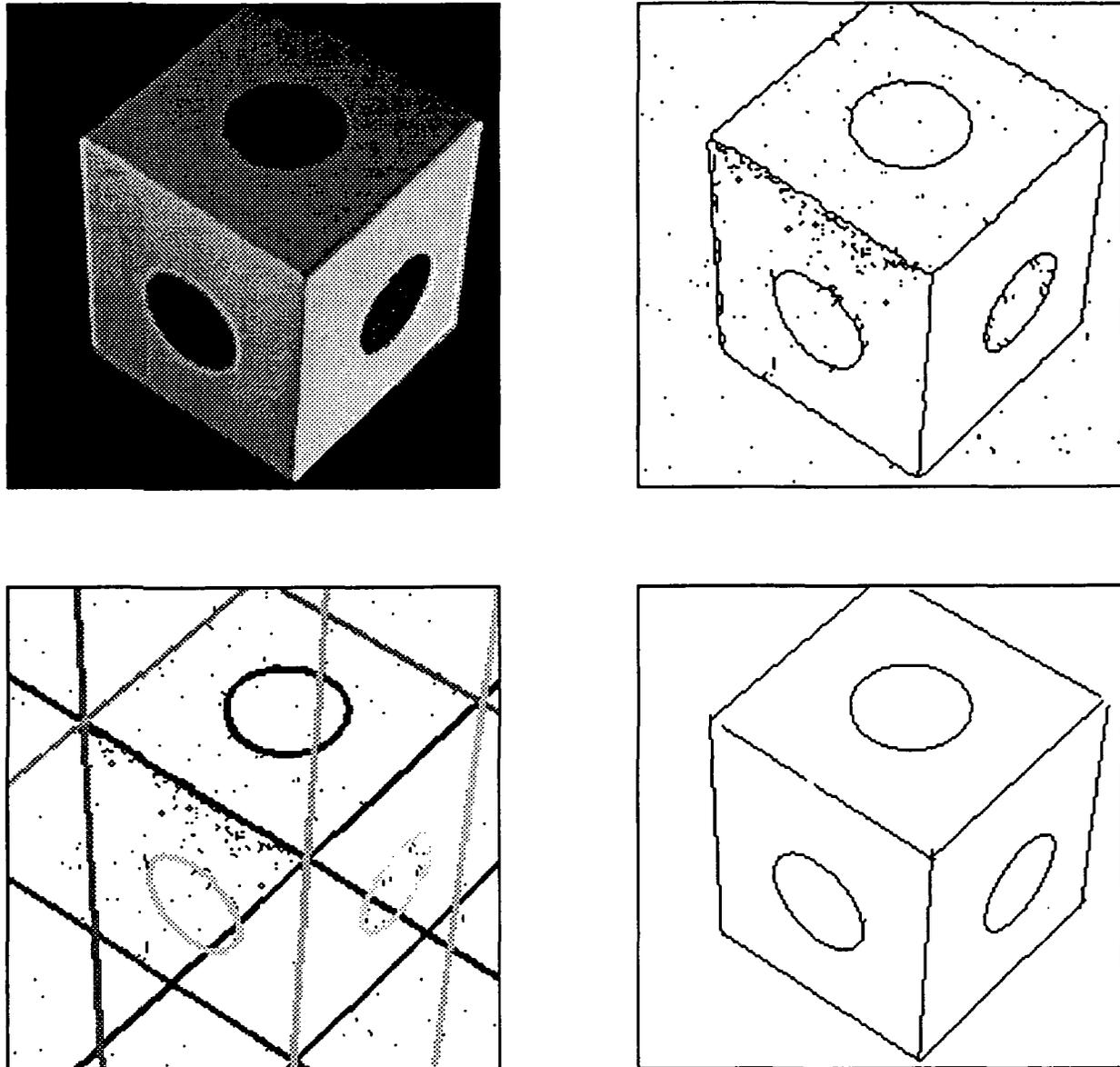


Figure 1: Results of the UPCPQS algorithm for the "box" image. (a) original image, (b) edge pixels obtained after thinning, (c) detected clusters with prototypes superimposed on the edge image in (b), and (d) the "cleaned" edge image.

5. F. L. Bookstein, "Fitting conic sections to scattered data", *Computer Vision, Graphics, and Image Processing*, vol. 9, pp. 56-71, 1979.
6. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
7. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. I, Addison Wesley, Reading, MA, 1992, Chapter 11.
8. G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, Nov. 1991, pp. 1115-1138.
9. R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering", submitted to *IEEE Transactions on Fuzzy Systems*.

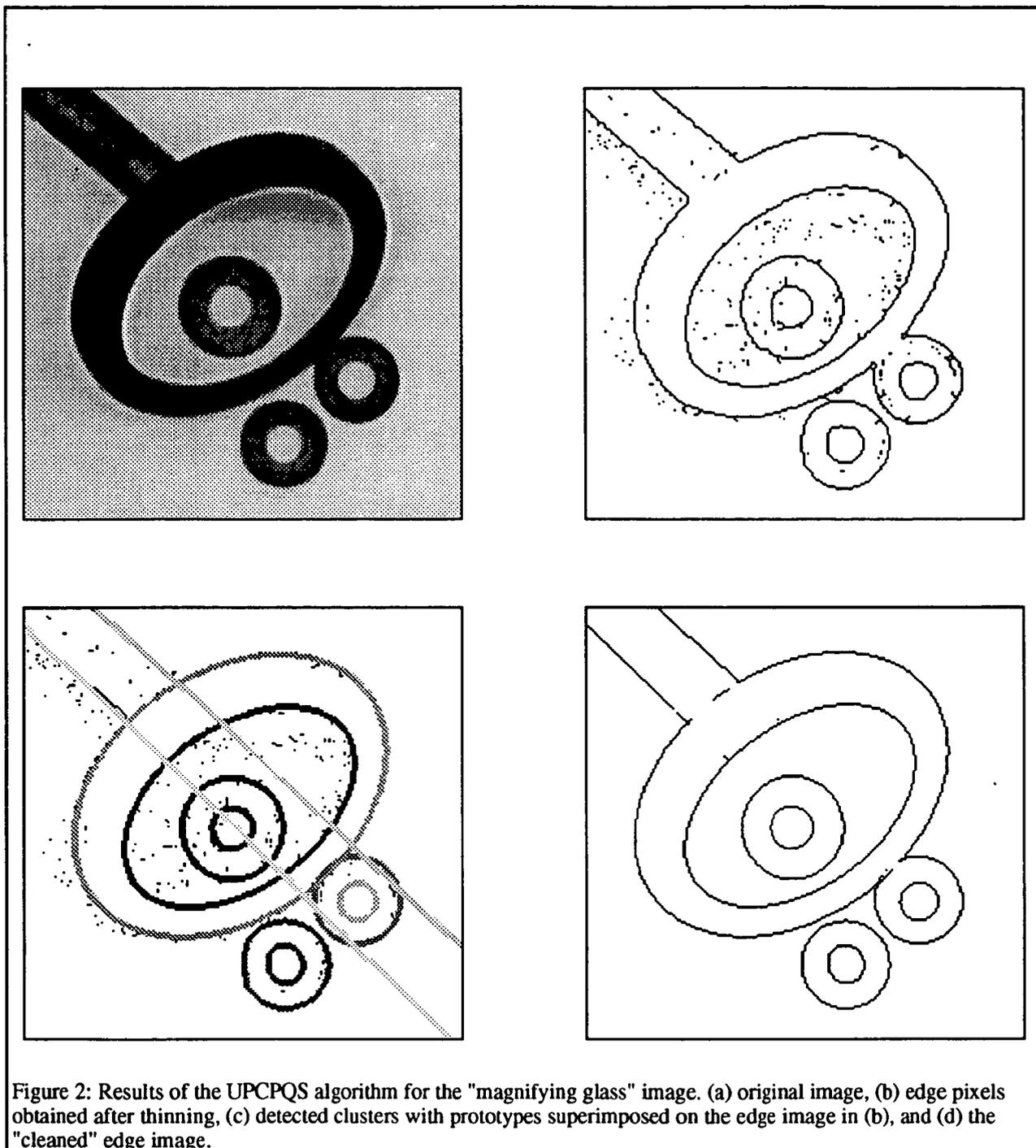


Figure 2: Results of the UPCPQS algorithm for the "magnifying glass" image. (a) original image, (b) edge pixels obtained after thinning, (c) detected clusters with prototypes superimposed on the edge image in (b), and (d) the "cleaned" edge image.

10. E. Backer and A. K. Jain, "A Clustering Performance Measure Based on Fuzzy Set Decomposition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 66, 1981.
11. I. Gath and A. B. Geva, Unsupervised optimal fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, July 1989, 773-781.
12. R. N. Dave, "New measures for evaluating fuzzy partitions induced through C-shells clustering", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, Nov. 1991, pp. 406-414.
13. R. Krishnapuram and C.-P. Freg, "Fitting an Unknown Number of Lines and Planes to Image Data through Compatible Cluster Merging", *Pattern Recognition*, vol. 25, no. 4, 1992.

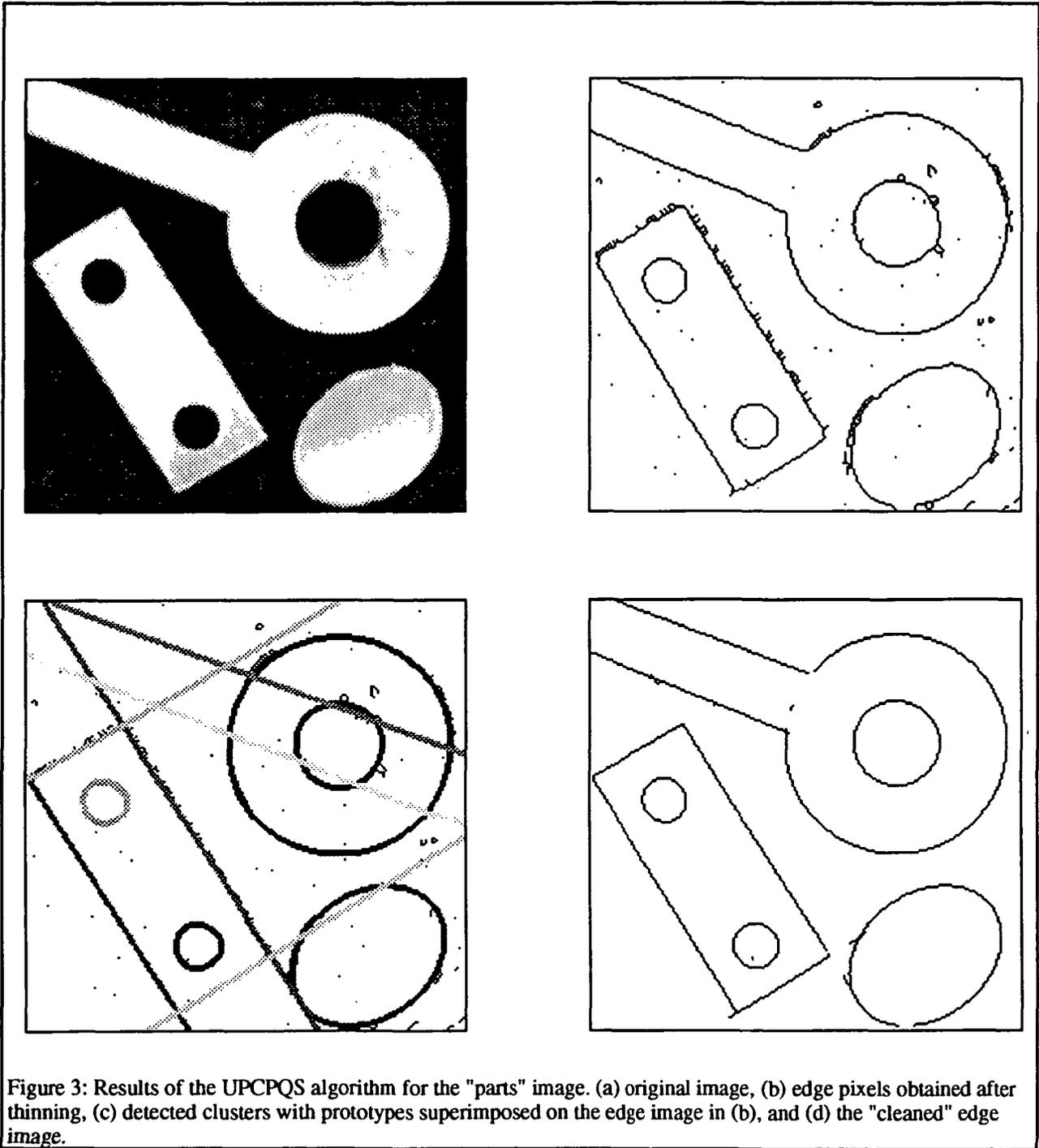


Figure 3: Results of the UPCPQS algorithm for the "parts" image. (a) original image, (b) edge pixels obtained after thinning, (c) detected clusters with prototypes superimposed on the edge image in (b), and (d) the "cleaned" edge image.

14. R. Krishnapuram and L. Chen, "Implementation of Parallel Thinning Algorithms Using Iterative Neural Networks", to appear in the *IEEE Transactions on Neural Networks*, 1992.
15. R. Krishnapuram, H. Frigui, and O. Nasraoui, "Quadric shell clustering algorithms and their applications", submitted to *Pattern Recognition Letters*.

## A Fuzzy Measure Approach to Motion Frame Analysis for Scene Detection

Albert B. Leigh, McDonnell Douglas Space Systems  
Sankar K. Pal, Indian Statistical Institute, Calcutta  
NASA Johnson Space Center  
Software Technology Branch/PT4  
Houston, TX 77058

### ABSTRACT

This paper addresses a solution to the problem of scene estimation of motion video data in the fuzzy set theoretic framework. Using fuzzy image feature extractors, a new algorithm is developed to compute the change of information in each of two successive frames to classify scenes. This classification process of raw input visual data can be used to establish structure for correlation. The algorithm attempts to fulfill the need for non-linear, frame-accurate access to video data for applications such as video editing and visual document archival/retrieval systems in multimedia environments.

### 1. INTRODUCTION

With rapid advancements in multimedia technology, it is increasingly common to have time-varied data like video as computer data types. Existing database systems do not have the capability to search within such information. It is a difficult problem to determine one scene from another because there are no precise markers that identify where they begin and end. And, divisions of scenes can be subjective especially if transitions are subtle. One way to estimate scene transitions is to mathematically approximate the change of information between each of two successive frames by computing the distance between their discriminatory properties. A fuzzy theoretic approach in image processing and pattern recognition provides convenient methods for such ambiguous or uncertainty measure.

#### 1.1 Fuzzy Image Concepts

In classical image processing, given a digital image, which has a  $M$  by  $N$  dimension with  $L$  gray levels, each picture element or pixel is represented as a spatial brightness function or gray information. Using fuzzy notion, an image can be considered as an array of fuzzy singletons, each having a value of membership denoting its degree of brightness relative to some brightness level,  $l$ , where  $l = 0, 1, 2, \dots, L-1$ . The fuzzy notation can be written as follows:

$$X = \{ \mu_{mn}(x_{mn}) = \mu_{mn} / x_{mn} : m = 1, 2, \dots, M; n = 1, 2, \dots, N \}$$

$$\text{or } X = \bigcup_m \bigcup_n \mu_{mn} / x_{mn}, m = 1, 2, \dots, M; n = 1, 2, \dots, N$$

where  $\mu_X(x_{mn})$  or  $\mu_{mn} / x_{mn}$ , ( $0 \leq \mu_{mn} \leq 1$ ) denotes the grade of possessing some property  $\mu_{mn}$  (e.g., brightness, edginess, smoothness) by the (m,n)th pixel intensity  $x_{mn}$ . In other words, a fuzzy subset of an image  $X$  is mapping  $\mu$  from  $X$  into  $[0,1]$  (Figure 1.1). For any point  $p \in X$ ,  $\mu(p)$  is called the degree of membership of  $p$  in  $\mu$  [11].

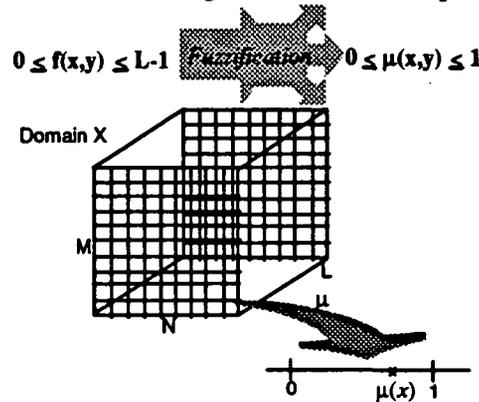


Figure 1.1: Fuzzy representation of an image X

## 2. IMAGE PROPERTIES

There are many spatial and geometric properties or features that can be measured or extracted from an image. They are used for pattern classifications and scene analysis. There is no trivial solution to selecting optimal features that would provide useful input values to the classifier. The effectiveness of these feature extractors also depends upon scenes. For this paper, six operators for ambiguity and fuzzy geometric measures are selected.

### 2.1 Ambiguity Measures

Two measures of ambiguities used are second-order local entropy and edginess. They produce a measure of structural information that exists in a given image. The entropy of an image can be defined as a measure of the information (ambiguities) gain in a given image. The edginess measures the coarseness of texture based on the average amount of ambiguity present in a given image.

#### 2.1.1 Second-order Local Entropy

The calculation of the second-order local entropy contains a window that operates on two adjacent pixels. This window is then used to compute the co-occurrence matrix for incorporating the dependency of the spatial distribution of gray levels. In this case, the

horizontal co-occurrence matrix is used. Then, the probability of the co-occurrence matrix is calculated with

$$p_{ij} = \frac{c_{ij}}{\sum_{ij} c_{ij}}, \text{ where } 0 \leq p_{ij} \leq 1 \text{ [12].}$$

$$\mu(X) = -\frac{1}{2} \sum_{ij} p_{ij} \log(p_{ij})$$

The information gain is computed with a logarithmic function. As described in [8], this could be an exponential function. The co-occurrence matrix computation could also be modified with a combination of horizontal and vertical directions for a more accurate measure of the spatial distributions.

### 2.1.2 Edginess

This image property is a measure of edge information to detect edge intensities in an input image. Note that this is different from the gradient descent edge detectors. It calculates the edge ambiguity using a localized window to find the boundary between the current pixel and neighboring pixels [12].

In the equation

$$\delta(X) = [1 - I(X)]^\beta,$$

$I(X)$  stands for the ambiguity measure, or the index of fuzziness, and  $\beta$  is a positive constant. The spatial dependent membership function,  $\mu_x$ , must be computed first.

$$\mu_x(x_{mn}) = \frac{0.5}{1 + \frac{1}{N_1} \sum_{ij} |x_{mn} - x_{ij}|}$$

where  $N_1$  represents the dimensions of the window of  $i$  by  $j$ , i.e.  $N_1 = i*j$ . These are neighboring pixels of the point  $(m, n)$ . As shown in Figure 2.1, the linear index of fuzziness,  $I(X)$ , can be defined as follows:

$$I(X) = \frac{2}{n} \sum_i \min(\mu_x(x_i), 1 - \mu_x(x_i)).$$

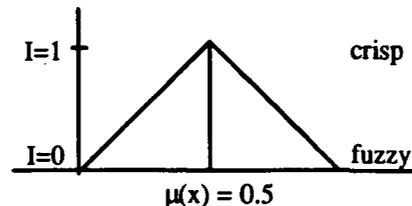


Figure 2.1: The linear index of fuzziness

Other measures of fuzziness, such as the quadratic index of fuzziness [6], fuzzy entropy [2], and index of non-fuzziness (crispness) [12], could also be used for the edginess measure.

## 2.2 Fuzzy Geometric Measures

Geometric measures define surfaces, shapes, solids, and boundaries of objects. Rosenfeld [13] and Rosenfeld and Haber [15] incorporated the fuzzy theoretic approach to the classical geometric measures and generalized some of the standard geometric properties of the relationships among regions to fuzzy sets [10]. Of these many measures, the primitive measures, such as area and perimeter, orientation measures, and shape measures are applied here.

The remaining methods that were applied, namely fuzzy geometrical properties, were extensions of the traditional geometrical measure concepts to operate in the fuzzy set framework. These measures examine various geometrical properties and relations such as area, perimeter, length, height, breadth, width, compactness, and elongatedness. There are many other topological concepts such as connectedness, major and minor axis, and adjacency, which could have been utilized in this study. These fuzzy measures are the basis for measuring spatial, gray, and region ambiguities.

### 2.2.1 Area

The area is an integral taken over the fuzzy image subset, i.e.  $\int \mu(x)$ . For a digital image, it is computed by summing the spatial brightness values of all image pixels. This spatial brightness value function is treated as the fuzzy membership function [11, 14].

$$\text{area}(\mu(x)) = \sum \mu(x)$$

### 2.2.2 Perimeter

The perimeter of an image is defined as the circumferential distance around the boundary. Using a faster method of computation, it can be computed as the sum of the product of the co-occurrence matrix and the difference of two adjacent pixels [11].

$$\text{perimeter}(\mu(X)) = \sum_i \sum_j c[i,j] |\mu(i) - \mu(j)|$$

where  $i=1, 2, \dots, L$  and  $j=1, 2, \dots, L$ .

### 2.2.3 Length

The length of an image is calculated by finding the longest extent in the column direction [11, 14].

$$\text{length}(\mu) = \max_m \left( \sum_n \mu_{mn} \right)$$

### 2.2.4 Height

The height of an image is another way of measuring its extent by summing the maximum membership values of each row [11, 14].

$$\text{height}(\mu) = \sum_n \max_m \mu_{mn}$$

### 2.2.5 Breadth

The breadth of an image measures the longest extent in the row direction [11, 14].

$$\text{breadth}(\mu) = \max_n \left( \sum_m \mu_{mn} \right)$$

### 2.2.6 Width

The width is calculated as the sum of maximum membership values of each column [11, 14].

$$\text{width}(\mu) = \sum_m \max_n \mu_{mn}$$

## 2.3 Orientations

The horizontal and vertical orientation of an image can be measured as follows [11]:

If  $\frac{\text{length}(\mu)}{\text{height}(\mu)} \leq 1$ , then vertically oriented.

If  $\frac{\text{breadth}(\mu)}{\text{width}(\mu)} \leq 1$ , then horizontally oriented.

## 2.4 Shape Measures

Shape measures can be computed using geometrical properties of a given image. These measures can also be defined independently of size measurements [16]. It basically represents the profile and physical structure of an image or image subsets. Two fuzzy measures are used: compactness and index of area coverage.

### 2.4.1 Compactness

The compactness measures the property of circularity [11].

$$\text{Comp}(\mu) = \frac{\text{area}(\mu)}{(\text{perimeter}(\mu))^2}$$

### 2.4.2 Index of Area Coverage

The index of area coverage (IOAC) is the fraction of the maximum area (that can be covered by the length and breadth of the image) actually covered by the image [11].

$$\text{IOAC}(\mu) = \frac{\text{area}(\mu)}{\text{length}(\mu) * \text{breadth}(\mu)}$$

## 3. SCENE ESTIMATION

As discussed in [12], the criterion of a good feature is that it should be invariant within class variation while emphasizing differences that are important in discriminating between patterns of different types. It is difficult to determine an optimal feature space comprising a set of image properties which would produce significant factors influential to classification decision. The approach taken for determining important features is to select image properties, namely ambiguity, size, orientation, and shape measures. Then, it translates all images to this pre-determined feature space.

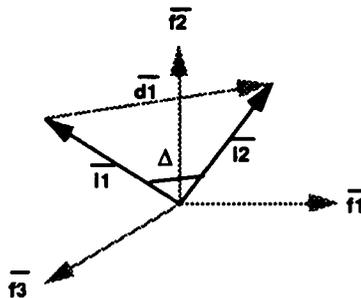


Figure 3.1: Fuzzy Image Feature Vector

Figure 3.1 depicts the sampled feature space having three features

$$\bar{i}_1 = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \quad \bar{i}_2 = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

and how the distance,  $|\bar{d}_1|$ , between two successive frames can be calculated with vector operation,  $|\bar{i}_1 - \bar{i}_2|$ . Because the goal is to analyze motion, this calculation of change of image constituents from frame to frame in a given time series gives the sampled mean and the sampled variance of all image features. By giving smaller weights to features having larger variance, the important features with small variance have more influence in the decision making process. It is discussed as a useful clustering technique to maximize the inter-set distance or minimize intra-set distance using a diagonal transformation such that features having larger variance are less reliable [12].

### 3.1 Distance Computation

Before the applied mathematical terms are discussed, the following nomenclatures need to be described.

M	Total number of frames or images
m	Last frame number where $m = M-1$
N	Total number of features or properties
i	Index to represent current image at $t$ where $i = 0, 1, \dots, m$
k	Index to represent the next image at $t+1$ where $k = 1, 2, \dots, m-1$

The sampled mean for the  $j^{\text{th}}$  feature element is given by,

$$\bar{f}_j = \frac{1}{M} \sum_{i=0}^m f_{ij} \quad \text{where } j = 1, 2, \dots, n.$$

Mnemonically, the index of feature element  $j$ , where  $j = 1, 2, \dots, n$ , can be represented in the following enumerated terms: edginess, entropy, compactness, ioac, l/h, and b/w, respectively (e.g.  $\bar{f}_{\text{entropy}}$ ). To standardize all sampled mean values to be 0.5, the following conversion is performed. This gives equal salience to all features for distance computation [3].

$$f_{ij}^{\text{norm}} = 0.5 \frac{f_{ij}}{\bar{f}_j}.$$

Consequently, this standardization makes all  $\bar{f}_j$  to be set to 0.5. And, the sampled variance for the  $j^{\text{th}}$  feature element is computed as

$$\sigma_j^2 = \frac{1}{(m-1)} \sum_{i=0}^m (f_{ij} - \bar{f}_j)^2 \quad \text{where } j = 1, 2, \dots, n.$$

The magnitude of the normalized distance between two successive frames  $i$  and  $k$  is [18],

$$D_{ik}^{\text{norm}} = \sqrt{\sum_{j=1}^n \frac{(f_{ij} - f_{kj})^2}{\sigma_j^2}} = \sum_{j=1}^n \left( \frac{|f_{ij} - f_{kj}|}{\sqrt{\sigma_j^2}} \right).$$

#### 4. EXPERIMENTS

Based on the above formulas, a schematic diagram (Figure 4.1) can be drawn to describe the process of feature selection and frame selection.

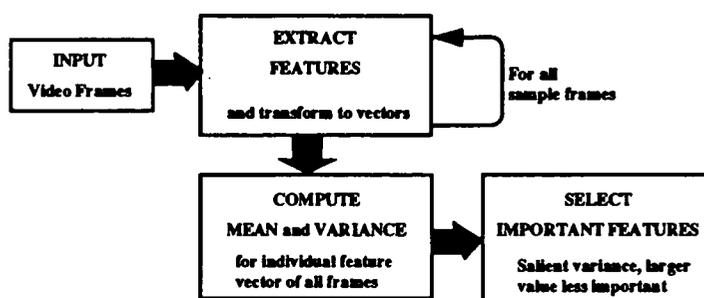


Figure 4.1: Schematic diagram of feature selection process

The distance between two frames in the aforementioned three feature space is computed to check the similarities. If this distance is larger than a predetermined threshold value, then the current video frame is considered to be significantly different from the previous frame, and therefore needs to be registered or stored as one of the abstract keys (Figure 4.2).

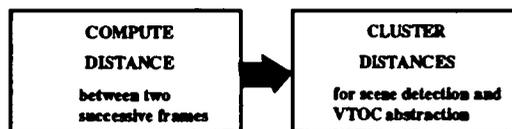


Figure 4.2: Schematic diagram of the frame selection process

#### 4.2 Input Data

Movie film projectors display 24 frames per second whereas NTSC standard television and video devices display 30 frames per second to achieve continuous and fluid full-motion images. The change of inter-frame information is gradual at such high frame rates. For storage conservation and computational efficiency, the simplest way to reduce or abstract video data is to sample it at lower frame rate.

In this paper, a time-suppressed frame rate of one per 5 seconds was assumed. A set of digitized video of previous space shuttle missions obtained from NASA/JSC was used (Figure 4.3). After a pre-processing step, each frame is stored in the CompuServ's Graphic Interchange File (GIF) format for portability.

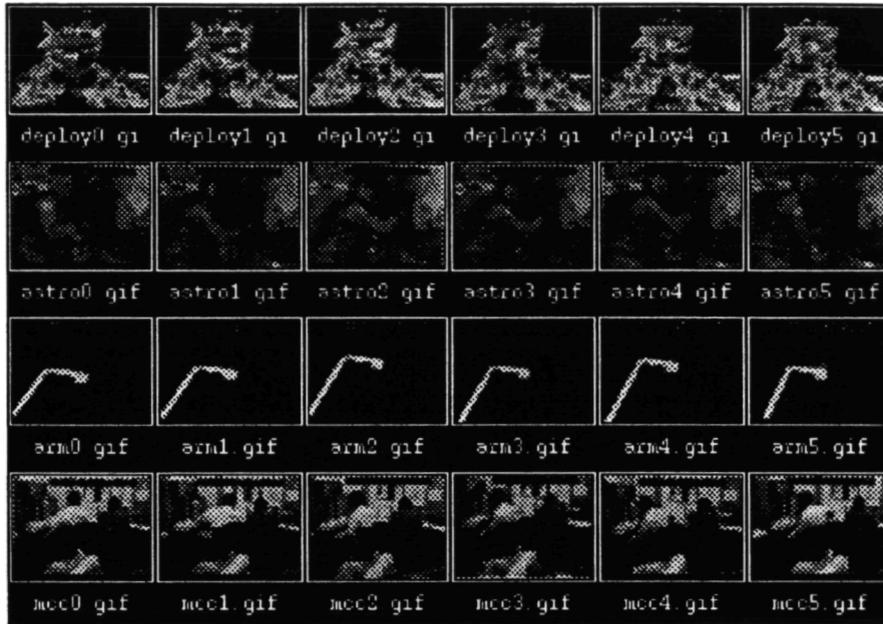


Figure 4.3: Experimental input data

With the fuzzy measures, the resulting distances between each two successive frames are shown in Figures 4.4 through 4.6. The abscissa represents the total number of frame distances in the sampled time series while the ordinate is the computed distance value between two successive images, i.e.  $|\bar{i}_j - \bar{i}_{j+1}|$ . For example, the abscissa index 0 represents  $|\bar{i}_0 - \bar{i}_1|$ , 1 represents  $|\bar{i}_1 - \bar{i}_2|$ , and so on. Each scene consists of six frames, therefore, there is a change of scene at every sixth index on the abscissa. The scene separation is denoted with vertical grid lines. Three sets of detection were experimented as follows:

- (1) Entropy, Compactness, L/H (Figure 4.4)
- (2) Edginess, IOAC, B/W (Figure 4.5), and
- (3) All of the above (Figure 4.6).

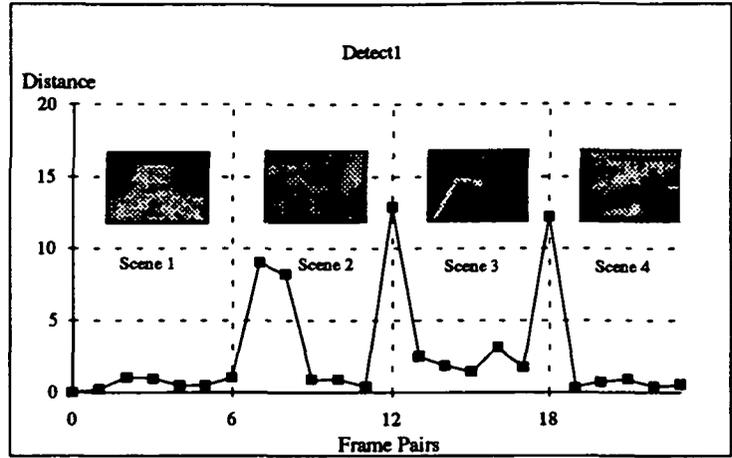


Figure 4.4: Detect 1 - Entropy, Compactness, L/H

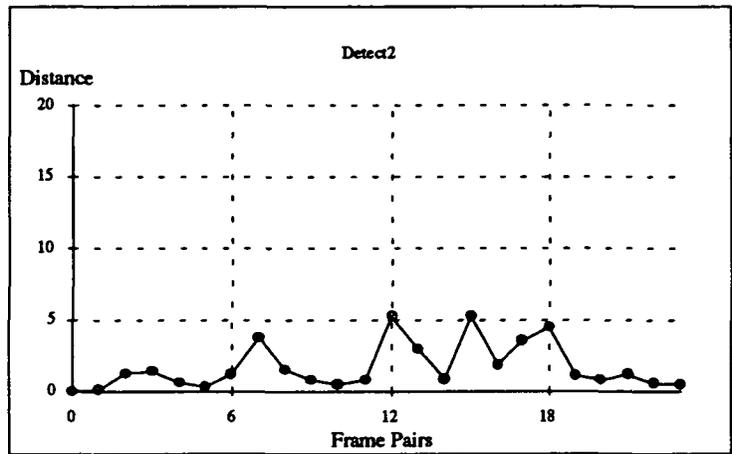


Figure 4.5: Detect 2 - Edginess, IOAC, B/W

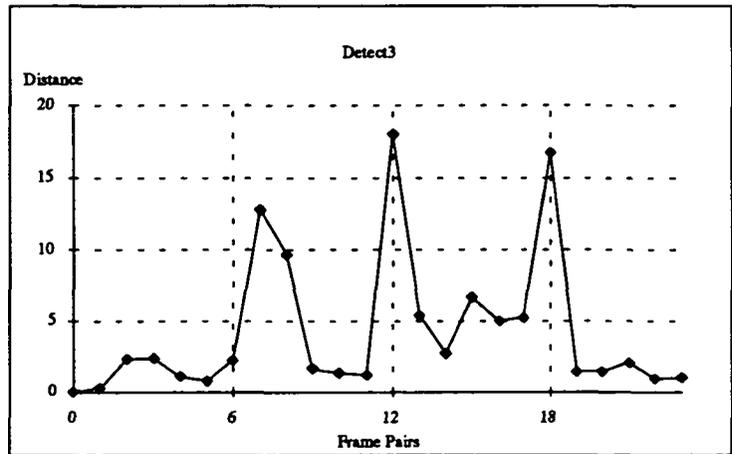


Figure 4.6: Detect 3 - All six features

It is to note that combining all features does not necessarily produce better results just because there are more features. It is not the quantity that is critical, but the discriminatory quality of features.

## 5. SUMMARY

The technique discussed here needs further improvements. It must have a classifier to correctly cluster the frames to the appropriate scenes. Both statistical and fuzzy approach pattern classifiers are being explored. Video frames that are to be classified are of temporal and dynamic data types, so non-linear classification methods need to be implemented. Scene classification is quite subjective in nature; therefore, the interactive tool developed here can be further extended to provide human interaction in setting problem-dependent criteria for this machine recognition task. Furthermore, the scenes that are detected may not necessarily be different from one another, but rather compose a video segment or document. A hierarchical abstraction scheme that allows for a higher level of abstraction will better suit the visual data management environment.

Finally, in the merging worlds of computers and media, new technologies mix traditional media such as video and publications with computer media as interactive, informational and entertainment software. This trend is rapidly growing at an unprecedented rate. Once digital video becomes a repository of common data on computers, the data needs to be accessed and manipulated just as documents are retrieved and managed by a DBMS. It might be useful to investigate new video inter-referencing strategies in correlating various context from the same event to derive knowledge points. Thus, this automatic abstraction of video index keys for non-linear, frame-accurate access would make information archival and retrieval applications more robust and efficient.

## 6. ACKNOWLEDGMENTS

The authors thank Dr. Robert N. Lea of NASA/JSC for his support in this work. This research was done as a master's thesis by Albert Leigh at the University of Houston-Clear Lake. It was partially supported by a NASA/JSC fellowship, "Graphics Technology in Space Application Program".

## 7. REFERENCES

1. Bezdeck, J. C. and Pal, S. K., Fuzzy Models for Pattern Recognition, IEEE Press, 1992
2. De Luca, A., and Termini, S., Information Control, vol. 20, pp. 301, 1972
3. Duda, R. and Hart, P., Pattern Classification and Scene Analysis, John Wiley & Sons, 1973
4. Gonzalez, R. C. and Wintz, P., Digital Image Processing, Addison-Wesley, 1988
5. Jain, A. K. Fundamentals of Digital Image Processing, Prentice-Hall, 1989
6. Kaufman, A., Introduction to the Theory of Fuzzy Subsets, Academic Press, 1975

7. Klir, G. J. and Folger, T. A., Fuzzy Sets, Uncertainty, and Information, Prentice-Hall, 1988
8. Pal, N. R. and Pal, S. K., "Object-Background Segmentation Using New Definitions of Entropy", IEE Proceedings, vol. 136, Pt. E, No. 4, July 1989
9. Pal, N. R. and Pal, S. K., "Entropic Thresholding", Signal Processing, vol 16, pp. 97-108, 1989
10. Pal, S. K., "Fuzzy Tools for the Management of Uncertainty in Pattern Recognition, Image Analysis, Vision and Expert systems", International Journal of Systems Science, vol. 22, No. 3, pp. 511-549, 1991
11. Pal, S. K., "Fuzziness, Image Information and Scene Analysis", Introduction to Fuzzy Logic Applications and Intelligent Systems, Edited by Yager R., and Zadeh, L., pp. 147-183, 1992
12. Pal, S. K., and Majumder, D. K., Fuzzy Mathematical Approach to Pattern Recognition, Halsted Press, 1986
13. Rosenfeld, A., "The Fuzzy Geometry of Image Subsets", Pattern Recognition Letters, vol. 2, pp. 311-317, September 1984
14. Rosenfeld, A., "Fuzzy Digital Topology", Information Control, vol. 40, no. 1, pp. 76-87, January 1979
15. Rosenfeld, A., and Haber, S., Pattern Recognition, vol. 18, pp 125, 1985
16. Rosenfeld, A. and Kak, A. C., Digital Picture Processing, vols. 1 and 2, Academic Press, Inc., 1982
17. Shannon, C. E., "A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, pp. 379-423, July 1948
18. Tou, J. and Gonzalez, R. C., Pattern Recognition Principles, Addison-Wesley, 1974
19. Zadeh, L. A., "Fuzzy Sets", Information and Control, vol. 8, pp. 338-352, 1965

**Automatic Rule Generation for High-Level Vision**

Frank Chung-Hoon Rhee and Raghu Krishnapuram  
 Department of Electrical and Computer Engineering  
 University of Missouri, Columbia, MO 65211

**ABSTRACT**

Many high-level vision systems use rule-based approaches to solve problems such as autonomous navigation and image understanding. The rules are usually elaborated by experts. However, this procedure may be rather tedious. In this paper, we propose a method to generate such rules automatically from training data. The proposed method is also capable of filtering out irrelevant features and criteria from the rules.

**1. Introduction**

High-level computer vision involves complex tasks such as image understanding and scene interpretation. In domains where the models of the objects in the image can be precisely defined, (such as the blocks world, or even the world of generalized cylinders) existing techniques for description and interpretation perform quite well. However, when this is not the case (such as the case of outdoor scenes or extra-terrestrial environments), traditional techniques do not work well. For this reason, we believe that the greatest contribution of fuzzy set theory to computer vision will be in the area of high-level vision. Unfortunately, very little work has been done in this highly promising area. Fuzzy set theoretic approaches to high-level vision have the following advantages over traditional techniques: i) they can easily deal with imprecise and vague properties, descriptions, and rules, ii) they degrade more gracefully when the input information is incomplete, iii) a given task can be achieved with a more compact set of rules, iv) the inferencing and the uncertainty (belief) maintenance can both be done in one consistent framework, v) they are sufficiently flexible to accommodate several types of rules other than just IF-THEN rules. Some examples of the types of rules that can be represented in a fuzzy framework are [1] possibility rules ("The more  $X$  is  $A$ , the more possible that  $B$  is the range for  $Y$ "), certainty rules ("The more  $X$  is  $A$ , the more certain  $Y$  lies in  $B$ "), gradual rules ("The more  $X$  is  $A$ , the more  $Y$  is  $B$ "), unless rules [2] ("if  $X$  is  $A$ , then  $Y$  is  $B$  unless  $Z$  is  $C$ ").

The determination of properties and attributes of image regions and spatial relationships among regions is critical for higher level vision processes involved in tasks such as autonomous navigation, medical image analysis and scene interpretation. Many high-level systems have been designed using a rule-based approach [3,4]. In these systems, common-sense knowledge about the world is represented in terms of rules, and the rules are then used in an inference mechanism to arrive at a meaningful interpretation of the contents of the image. In a rule-based system to interpret outdoor scenes, typical rules may be

**IF a REGION is RATHER THIN AND SOMEWHAT STRAIGHT  
 THEN it is a ROAD**

**IF a REGION is RATHER GREEN AND HIGHLY TEXTURED AND  
 IF the REGION is BELOW a SKY REGION  
 THEN it is TREES**

Attributes such as "THIN" and "NARROW", and properties such as "BRIGHT" and "TEXTURED" defy precise definitions, and they are best modeled by fuzzy sets. Similarly, spatial relationships such as "LEFT OF", "ABOVE" and "BELOW" are difficult to model using the all-or-nothing traditional techniques [5]. We may interpret the attributes, properties and relationships as "criteria". Therefore, we believe that a fuzzy approach to high-level vision will yield more realistic results.

In most rule-based systems, the rules are usually enumerated by experts, although they may also be generated by a learning process. Several techniques have been suggested in the literature to generate rules for control problems [6-9], some of which use neural net methods to model the control system [7-12]. These systems convert a given set of inputs to an output by fuzzifying the inputs, performing fuzzy logic, and then finally defuzzifying the result of the inference to generate a crisp output [13]. Some of the methods also "tune" the membership functions that define the levels (such as "LOW", "MEDIUM" and "HIGH") of the input variables [10]. While these methods have been shown to be very effective in solving control problems, they cannot be directly used in high-level vision

applications. For example, in control systems, the fuzzy rules have consequents which are usually a desired level of a control signal whereas in high-level vision, the consequent clauses are usually fuzzy labels. Also, it is desirable that membership functions for levels of fuzzy attributes such as "THIN", and "NARROW", and properties such as "BRIGHT" be related to how humans perceive such attributes or properties. Hence they have very little to do with the decision making or reasoning process in which they are employed. In many reasoning systems for high-level vision, confidence (or importance) factors are associated with every rule since the confidence in the labeling may depend on the confidence of the rule itself. In this paper, we propose a new method to generate rules for high-level vision applications automatically. The rules so obtained may be combined with the rules given by the experts to complete the rule base.

In Section 2, we describe several fuzzy aggregation operators which can be used in hierarchical (multi-layer) aggregation networks for multi-criteria decision making. In Section 3, we describe how these aggregation networks can be used to filter out irrelevant attributes, properties, and relationships and at the same time generate a compact set of fuzzy rules (with associated confidence factors) that describes the decision making process. In Section 4 we present some experimental results on automatic rule generation. Finally Section 5 contains the summary and conclusions.

## 2. Fuzzy Aggregation Operators

Fuzzy set theory provides a host of very attractive aggregation connectives for integrating membership values representing uncertain and subjective information [14]. These connectives can be categorized into the following three classes based on their aggregation behavior: i) union connectives, ii) intersection connectives, and iii) compensative connectives. Union connectives produce a high output whenever any one of the input values representing different features or criteria is high. Intersection connectives produce a high output only when all of the inputs have high values. Compensative connectives are used when one might be willing to sacrifice a little on one factor, provided the loss is compensated by gain in another factor. Compensative connectives can be further classified into mean operators and hybrid operators. Mean operators are monotonic operators that satisfy the condition:  $\min(a,b) \leq \text{mean}(a,b) \leq \max(a,b)$ . The generalized mean operator [15] as given below is one of such operator.

$$g_p(x_1, \dots, x_n; w_1, \dots, w_n) = \left( \sum_{i=1}^n w_i x_i^p \right)^{1/p}, \quad \text{where } \sum_{i=1}^n w_i = 1. \quad (1)$$

The  $w_i$ 's can be thought of as the relative importance factors for the different criteria. The generalized mean has several attractive properties. For example, the mean value always increases with an increase in  $p$  [15]. Thus, by varying the value of  $p$  between  $-\infty$  and  $+\infty$ , we can obtain all values between min and max. Therefore, in the extreme cases, this operator can be used as union or intersection. The  $\gamma$ -model devised by Zimmermann and Zysno [16] is an example of hybrid operators, and it is defined by

$$y = \left( \prod_{i=1}^n x_i^{\delta_i} \right)^{1-\gamma} \left( 1 - \prod_{i=1}^n (1-x_i)^{\delta_i} \right)^{\gamma}, \quad \text{where } \sum_{i=1}^n \delta_i = n \text{ and } 0 \leq \gamma \leq 1. \quad (2)$$

In general, hybrid operators are defined as the weighted arithmetic or geometric mean of a pair of fuzzy union and intersection operators as follows.

$$A \oplus_{\gamma} B = (1-\gamma)(A \cap B) + \gamma(A \cup B) \quad (3)$$

$$A \otimes_{\gamma} B = (A \cap B)^{(1-\gamma)} (A \cup B)^{\gamma} \quad (4)$$

The parameter  $\gamma$  in (3) and (4) controls the degree of compensation. The  $\gamma$ -model in (2) is a hybrid operator of the type in (4). The compensative connectives are very powerful and flexible in that by choosing correct parameters, one can not only control the nature (e.g. conjunctive, disjunctive, and compensative), but also the attitude (e.g. pessimistic and optimistic) of the aggregation.

One can formulate the problem of multicriteria decision making as follows. The support for a decision may depend on supports for (or degrees of satisfaction of) several different criteria, and the degree of satisfaction of each

criterion may in turn depend on degrees of satisfaction of other sub-criteria, and so on. Thus, the decision process can be viewed as a hierarchical network, where each node in the network "aggregates" the degree of satisfaction of a particular criterion from the observed support. The inputs to each node are the degrees of satisfaction of each of the sub-criteria, and the output is the aggregated degree of satisfaction of the criterion. Thus, the decision making problem reduces to i) selecting robust and useful criteria for the problem on hand, ii) finding ways to generate memberships (degrees of satisfaction of criteria) based on values of features (criteria) selected, and iii) determining the structure of the network and the nature of the connectives at each node of the network. This includes discarding irrelevant criteria to make the network simple and robust.

In our previous research, we have investigated the properties of several union and intersection operators, the generalized mean, and the  $\gamma$ -model [14,17]. We have shown that optimization procedures based on gradient descent and random search can be used to determine the proper type of aggregation connective and parameters at each node, given only an approximate structure of the network and given a set of training data that represent the inputs at the bottom-most level and the desired outputs at the top-most level [14,17]. In this paper, we extend this idea to the detection of irrelevant attributes and automatic rule generation.

### 3. Redundancy Analysis and Rule Generation

In the approach we propose, we first fuzzily partition the range of values that each criterion (property or an attribute or a relation) can take into several linguistic intervals such as LOW, MEDIUM and HIGH. The set of properties or an attribute or a relation which are used are the ones that may appear in the antecedent clause of a rule. As explained in Section 1, the membership function for each level needs to be determined according to how humans perceive such attributes, properties or relations. The membership values for an observed attribute, property or relationship value in each of the levels is calculated using such membership functions. (Methods to generate degrees of satisfaction of relationships such as "LEFT OF" may be found in [18]). The memberships are then aggregated in a fuzzy aggregation network of the type shown in Figure 1. The top nodes of the network represent the labels that may appear in the consequents of the rules. A suitable structure for the network, and suitable fuzzy aggregation operators for each node are chosen. The network is then trained with typical attribute, property or relationship data with the corresponding desired output values for the various labels to learn the aggregation connectives and connections that would best describe in input-output relationships. The learning may be implemented using a gradient descent approach similar to the backpropagation algorithm [14,17]. It is to be noted that there is a constraint on the weights.

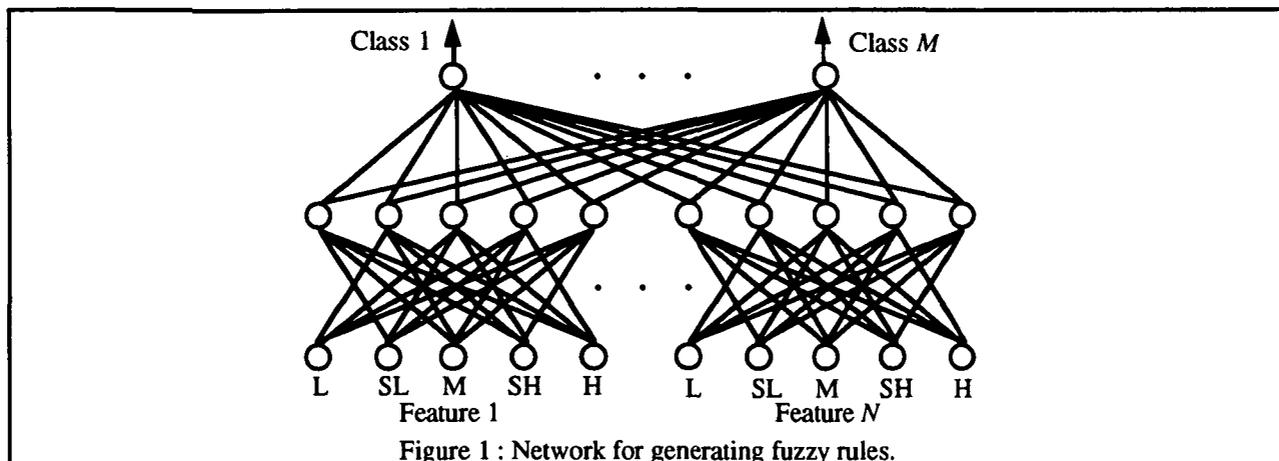


Figure 1 : Network for generating fuzzy rules.

Our experiments indicate that the choice of the network is not very critical. Also any compensative aggregation operator seems to yield good results. In all the results shown in this paper, we used the generalized mean operator as the aggregation operator. As indicated in Section 2, the generalized mean can closely approximate a union (intersection) operator for a large positive (negative) value of  $p$ . We start the training with the generalized mean aggregation function with  $p=1$ . If the training data is better described by a union (intersection) operator, then the value of  $p$  will keep increasing (decreasing) as the training proceeds, until the training is terminated when the error becomes acceptable. Also, the weights  $w_i$  in (1) may be interpreted as the relative importance factors for the different criteria. Initially we start the training with all the weights associated with a node being equal. As the training proceeds the weights automatically adjust so that the overall error decreases. Some of the weights eventually become

very small. Thus, the training procedure has the ability to detect certain types of redundancies in the network. In general, there are three types of redundancies (irrelevant criteria) that are encountered in decision making [17]. These correspond to uninformative, unreliable, and superfluous criteria.

**Uninformative Criteria:** These are criteria whose degrees of satisfaction are always approximately the same, regardless of the situation. Therefore, these criteria do not provide any information about the situation, thus contributing little to the decision-making process. For example, low texture content is a criterion that is always satisfied for both clear skies and roads, and hence it would be a uninformative criterion if one needs to distinguish between these two labels. Uninformative criteria do not contribute to the robustness of the decision making process, and therefore it is desirable that they be eliminated.

**Unreliable Criteria:** These correspond to criteria whose degrees of satisfaction do not affect the final decision. In other words, the final decision is the same for a wide range of degrees of satisfaction. For example, color would be an unreliable criterion for distinguishing a rose from a hibiscus because they both come in similar colors. Unreliable criteria do not contribute to the robustness of the decision making process, and therefore it is desirable that they be eliminated.

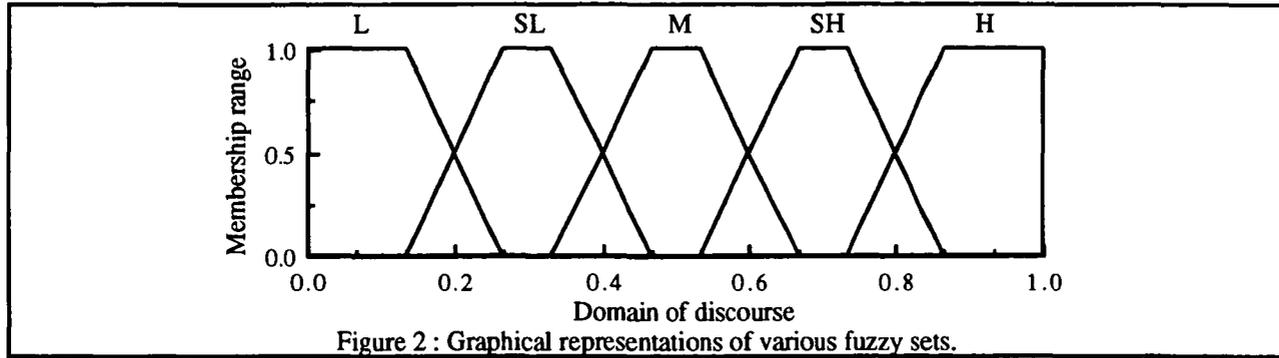
**Superfluous Criteria:** These are criteria which are strictly speaking not required to make the decision. The decisions made without considering such criteria may be as accurate or as reliable. For example, one may want to differentiate planar surfaces from spherical surfaces using Gaussian and mean curvatures, but the criteria are superfluous because either one of them is sufficient to distinguish between planar and spherical surfaces. However, redundancies of this type are not entirely without utility, since such redundancies make the decision making process more robust. If one criterion fails for some reason, we may still be able to arrive at the correct decision using the other. Hence such redundancies may be desirable to increase the robustness of the decision-making process.

**Redundancy Detection and Estimation of Confidence Factors:** A connection is considered redundant if the weight associated with it gradually approaches to zero (or a small threshold value) as the learning proceeds. A node (associated with a criterion) is considered redundant if all the connections from the output of this node to other nodes become redundant. Our simulations show that both in the case of uninformative criteria and unreliable criteria, the weights corresponding to all the output connections go to zero. Therefore such nodes (criteria) are eliminated from the structure. The examples in Section 4 illustrate this idea.

**Rule Generation:** The networks that finally result from this training process can be said to represent rules that may be used to make the decisions. If the final value of the parameter  $p$  at a given node is greater than one, the nature of the connective is disjunctive. If the value is less than one, it is conjunctive. Once the nature of the connective at each node is determined, we can easily construct the fuzzy rules that describe the input-output relations. In Section 4 we present some examples of this approach.

## 4. Experimental Results

In this section, we present some typical experimental results involving both synthetic and real data to show the effectiveness of the proposed automatic rule generation method. The method is shown to generate decision rules that best describe the decision criteria for the classes in each experiment. Figure 1 shows the general 3 layer aggregation network used to generate the rules. The input layer consists of  $nN$  number of input nodes where  $N$  is the number of fuzzy features or criteria (such as properties and relationships) and  $n$  is the number of linguistic levels used to partition each feature. For the hidden layer, there are  $nN$  hidden nodes where each node is connected to all but one (i.e., it is connected to  $n-1$ ) input nodes representing levels within each feature. The top layer fully connects the hidden layer. In the experimental results shown here, we used 5 fuzzy linguistic levels to represent each feature, therefore, each hidden node has 4 connections. Other types of network structures were also tried, however the one described above produced the best results. The target values in the training data were chosen to be 1.0 for the class from which the training data was extracted, and 0.0 for the remaining classes. The feature values were always normalized so that they fall in the range [0,1]. Figure 2 depicts the trapezoidal fuzzy sets used to model the intuitive notions of the five linguistic levels **LOW (L)**, **SOMEWHAT LOW (SL)**, **MEDIUM (M)**, **SOMEWHAT HIGH (SH)**, and **HIGH (H)**.



#### 4.1 The ellipse problem

Figure 3 shows the scatter plot of the "ellipse data" after mapping each sample feature into the interval [0,1]. There are 50 samples in each class. The membership values in each linguistic level for each sample is computed using the membership functions shown in Figure 2, and these with the corresponding desired targets are used as training data in the training algorithm described in Section 3. Figure 4 shows the reduced network after training. All connections with weights below a value of 0.01 were considered as redundant. Table 1 shows the final weights (which determine the confidence factors of the rules and criteria) and the  $p$  parameter values (which determine the conjunctive or disjunctive nature of the connective) for the specified nodes in Figure 4. Using the properties for the  $p$  values obtained, the following rules are generated, as discussed in Section 3.

$$\text{Class 1} = (\text{Feature 1 SL} \vee \text{Feature 1 M} \vee \text{Feature 1 SH}) \wedge (\text{Feature 2 SL} \vee \text{Feature 2 M} \vee \text{Feature 2 SH}). \quad (5)$$

In other words, the rule may be summarized as

**R<sub>1</sub> : IF Feature 1 is SL or M or SH and Feature 2 is SL or M or SH THEN the class is Class 1.**

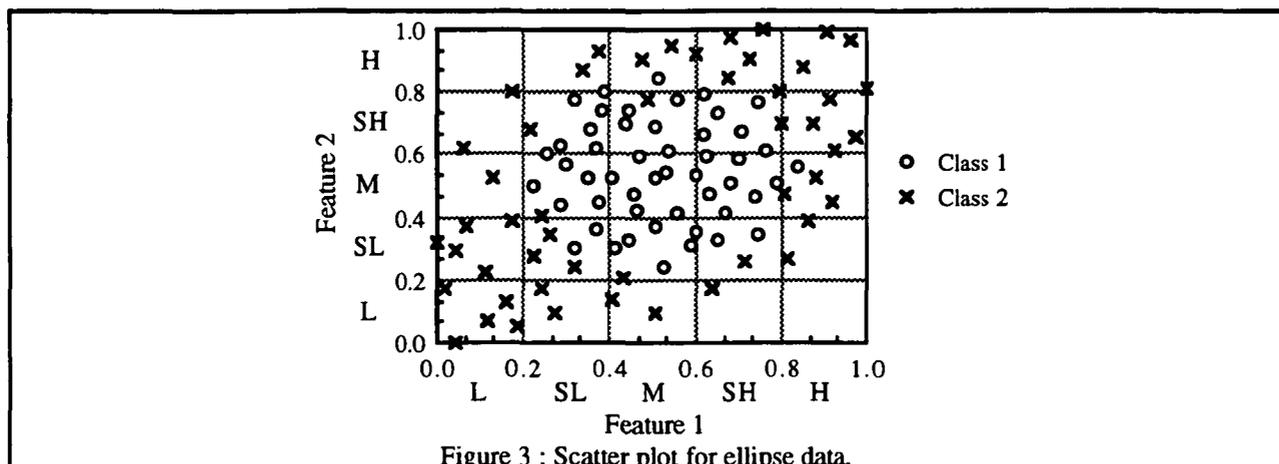
Similarly,

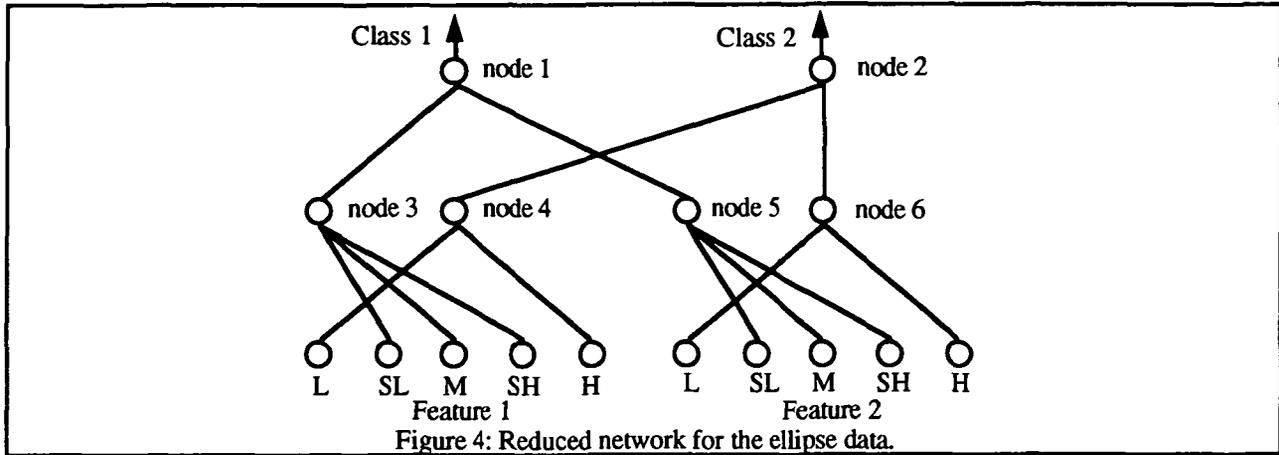
$$\text{Class 2} = (\text{Feature 1 L} \vee \text{Feature 1 H}) \vee (\text{Feature 2 L} \vee \text{Feature 2 H}) \quad (6)$$

and

**R<sub>2</sub> : IF Feature 1 is L or H or Feature 2 is L or H THEN the class is Class 2.**

These rules make sense since the expansion (5) fuzzily covers the 9 inner cells and the expansion of (6) fuzzily covers the outer 16 cells of the plot shown in Figure 3.





#### 4.2 The natural scene problem

Figure 5(a) shows a 256x256 image of a natural scene and Figure 5(b) shows the scatter plot of the training samples extracted from three different regions (vegetation, sky, and road) in the image. The two features used were the intensity and the position (row number) of the pixels. We used 40 samples from each class. Figure 5(c) shows the reduced network after training. Table 1 shows the final weights and  $p$  parameter values for the specified nodes in Figure 5(c). The following rules may be generated from the reduced network.

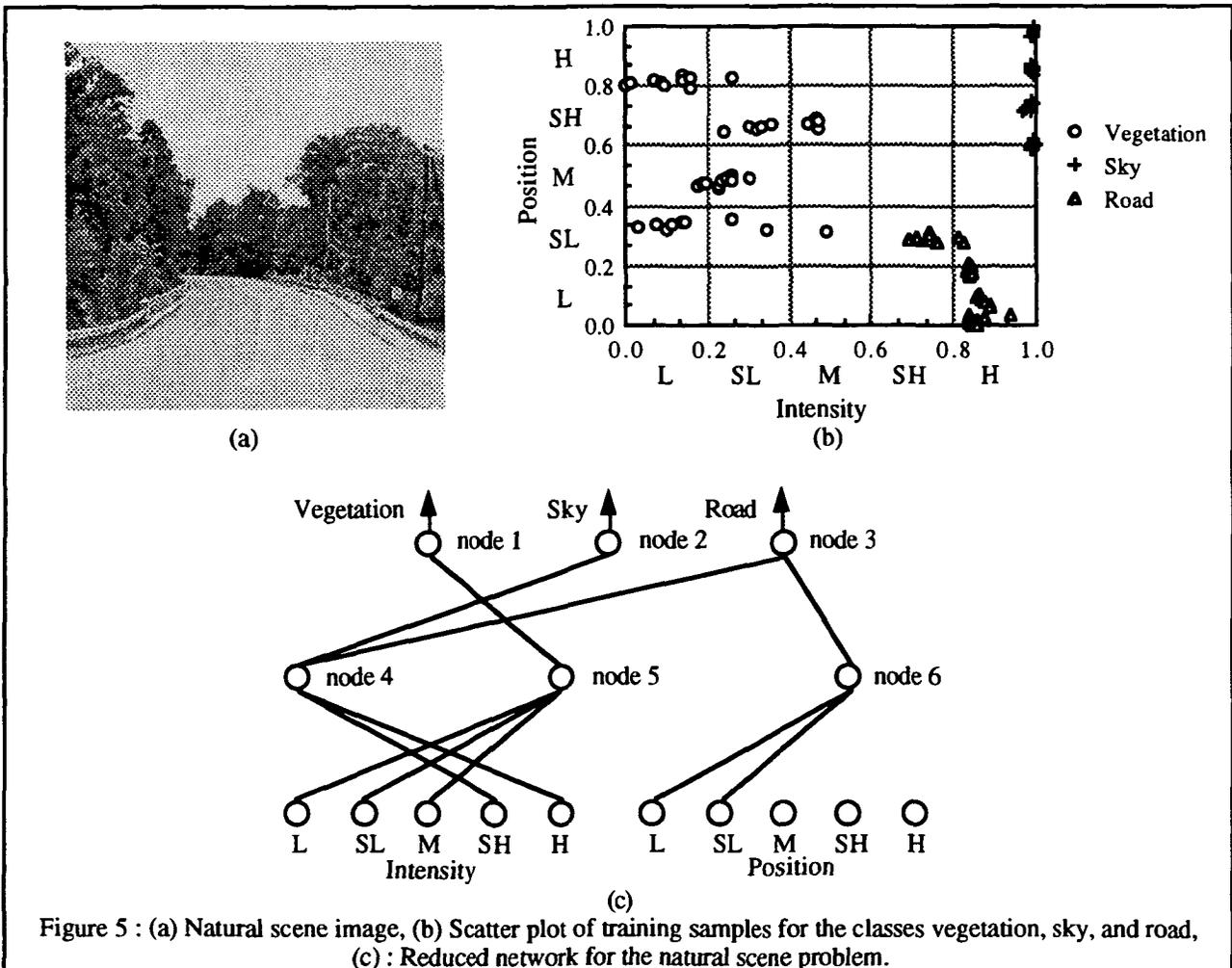


Figure 5 : (a) Natural scene image, (b) Scatter plot of training samples for the classes vegetation, sky, and road, (c) : Reduced network for the natural scene problem.

Table 1: Values of weights and parameter  $p$  for the ellipse and natural scene problems.

	ellipse problem		natural scene problem	
	weights	$p$	weights	$p$
node 1	0.55 0.45	-0.45	0.99	2.28
node 2	0.54 0.46	6.03	0.99	-0.30
node 3	0.19 0.52 0.28	6.15	0.23 0.77	-0.21
node 4	0.52 0.48	6.02	0.20 0.80	2.26
node 5	0.06 0.54 0.40	6.18	0.40 0.41 0.19	5.44
node 6	0.56 0.43	5.87	0.58 0.42	3.50

$$\text{Class Vegetation} = (\text{Intensity L} \vee \text{Intensity SL} \vee \text{Intensity M}). \quad (7)$$

**R<sub>VEG</sub>** : IF Intensity is L or SL or M  
THEN the class is Vegetation.

$$\text{Class Sky} = (\text{Intensity SH} \vee \text{Intensity H}) \quad (8)$$

**R<sub>SKY</sub>** : IF Intensity is SH or H  
THEN the class is Sky.

$$\text{Class Road} = (\text{Intensity SH} \vee \text{Intensity H}) \wedge (\text{Position L} \vee \text{Position SL}) \quad (9)$$

**R<sub>ROAD</sub>** : IF Intensity is SH or H and Position is L or SL  
THEN the class is Road.

In the rule for vegetation, the position feature becomes redundant (i. e., all position weights connected to vegetation drop towards zero). This is reasonable, since the intensity feature clearly separates vegetation from the other classes and the position feature is "unreliable" according to the definition in Section 3. Also, in the rule for sky, the intensity of the sky is more or less uniform and so the intensity feature can clearly distinguish the sky from the other classes. The position feature is again "unreliable". In the rule for road, both position and intensity features play a role. This makes sense since when considering the road, the position feature clearly separates it from the sky and the intensity feature can separate it from the vegetation.

## 5. Summary and Conclusions

In this paper, we introduced a new method for automatically generating rules for high level vision. The range of each feature is fuzzily partitioned into several linguistic intervals such as LOW, MEDIUM and HIGH. The membership function for each level is determined, and the membership values for an observed feature value in each of the linguistic levels is calculated using these membership functions. The memberships are then aggregated in a fuzzy aggregation network. The networks are trained with typical data to learn the aggregation connectives and connections that would give rise to the desired decisions. The learning process can also be made to discard redundant features. The networks that finally result from this training process can be said to represent rules that may be used to make the decisions. Riseman et al used similar rules for segmentation and labeling of outdoor scenes, but the weights used in the aggregation scheme were determined empirically [19]. The ability to generate rules that can be used in fuzzy logic

and rule-based systems directly from training data is a novel aspect of our approach. One of the issues that requires investigation is the choice of the number of linguistic levels and its effect on the decision making process.

## 6. References

1. D. Dubois and H. Prade, "Possibility Theory as a basis for preference propagation in automated reasoning", *Proceedings of the First IEEE Conference on Fuzzy Systems*, March 1992, San Diego, pp. 821-832.
2. D. Driankov, "Fuzzy logic with unless-rules", *Proceedings of the First IEEE Conference on Fuzzy Systems*, March 1992, San Diego, pp. 255-262.
3. T. O. Binford, "Survey of model-based image analysis systems", *International Journal of Robotics Research*, vol. 1, no. 1, Spring 1982, pp. 18-64.
4. D. M. McKeown, W. A. Harvey, and J. McDermott, "Rule-based interpretation of aerial imagery", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 5, 1985, pp. 570-585.
5. J. Freeman, "The modeling of spatial relations", *Computer Graphics and Image Processing*, vol. 4, pp. 156-171, 1975.
6. B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
7. H. Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks - Survey and Future Directions", *Proceedings of the International Conference on Fuzzy Theory and Neural Networks*, IIZUKA-90, pp. 13-17.
8. C. T. Lin, C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Computers*, vol. c-40, no. 12, Dec. 1991.
9. L. X. Wang, J. M. Mendel, "Fuzzy basis functions and orthogonal least squares learning," to appear in *IEEE Trans. Neural networks*, September 1992.
10. C. T. Lin, C. S. G. Lee, "Real-time Supervised Structure/Parameter Learning for fuzzy neural network," *Proceedings of the First IEEE International Conference on Fuzzy Systems*, San Diego, CA, March, 1992, pp. 1283-1291.
11. H. Nomura, I. Hayashi, and N. Wakami, "A learning method of fuzzy inference rules by descent method," *Proceedings of the First IEEE International Conference on Fuzzy Systems*, San Diego, CA, March, 1992, pp. 203-210.
12. D. G. Barkhart, P. P. Bonissone, "Automated fuzzy knowledge base generation and tuning," *Proceedings of the First IEEE International Conference on Fuzzy Systems*, San Diego, CA, March, 1992, pp. 179-188.
13. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller - parts I & II," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-20, no. 2, pp. 404-435, 1990.
14. R. Krishnapuram and J. Lee, "Fuzzy-connective-based hierarchical aggregation networks for decision making," *Fuzzy Sets Syst.*, vol. 46, no. 1, pp. 11-27, Feb. 1992.
15. H. Dyckhoff and W. Pedrycz, "Generalized means as a model of compensation connectives," *Fuzzy Sets Syst.*, vol. 14, no. 2, pp. 143-154, 1984.
16. H.-J. Zimmermann and P. Zysno, "Decisions and evaluations by hierarchical aggregation of information," *Fuzzy Sets Syst.*, vol. 10, no. 3, pp. 243-260, 1983.
17. R. Krishnapuram and J. Lee, "Fuzzy-set-based hierarchical networks for information fusion in computer vision," *The Journal of Neural Networks*, vol. 5, no. 2, pp. 335-350, March 1992.
18. R. Krishnapuram, J. M. Keller, and Y. Ma, "Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions", *Proceedings of NAFIPS*, Puerto Vallarta, Mexico, Dec. 1992.
19. E. M. Riseman and A. R. Hanson, "A Methodology for the Development of General Knowledge-Based Vision Systems", in *Vision Systems and Cooperative Computation*, M. A. Arbib (Editor), MIT Press, Cambridge Ma, 1988.

## ENCODING SPATIAL IMAGES - A FUZZY SET THEORY APPROACH

Leszek M. Sztanderas, Electrical Engineering Department  
University of Toledo, Toledo, OH 43606, USA  
lsztand@uoft02.utoledo.edu

## ABSTRACT

As the use of fuzzy set theory continues to grow, there is an increased need for methodologies and formalisms to manipulate obtained fuzzy subsets. Concepts involving relative position of fuzzy patterns are acknowledged as being of high importance in many areas.

In this paper, we present an approach based on the concept of dominance in fuzzy set theory for modelling relative positions among fuzzy subsets of a plane. In particular, we define the following spatial relations: to the left (right), in front of, behind, above, below, near, far from, and touching.

This concept has been implemented to define spatial relationships among fuzzy subsets of the image plane. Spatial relationships based on fuzzy set theory, coupled with a fuzzy segmentation should therefore yield realistic results in scene understanding.

## INTRODUCTION

One of the main difficulties in computer vision is the difference between how a human sees a scene and how a computer sees it. A human may see a large red building between two trees, but the computer "sees" only a two-dimensional array of pixel values.

To design a user interface for computer vision that can be used without extensive special training we have to translate from the computer's view to the human's. We must segment the image, properly label the objects in it, and then describe the objects both in terms of their absolute properties and in terms of their properties relative to each other.

This paper proposes to examine ways of defining and deriving the relative spatial properties of the objects in an arbitrary scene.

## A Need of Fuzzy Set Theory in Computer Vision

In computer vision, the standard approach to image analysis and

---

\* Part of this work was done when the author was with Department of Electrical & Computer Engineering, University of Missouri- Columbia, Columbia, MO 65211.

recognition is to segment the image into regions and to compute various properties of and relationships among these regions. However, the regions are not always "crisply" defined. It is sometimes more appropriate to regard them as fuzzy subsets of the image.

In the last several years, there has been increased attention given to the use of fuzzy set theory in image segmentation [1, 2, 3, 4].

When the objects in a scene are represented by crisp sets, the all-or-nothing definitions of the subsets actually add to the problem of generating such relational descriptions. It is our belief that definitions of spatial relationships based on fuzzy set theory, coupled with a fuzzy segmentation will yield realistic results.

For the purpose of this work we assume that we deal with an image of objects, that is, the scene has already been segmented and the objects have been labelled. The segmentation may be either crisp or fuzzy.

Using the above considerations the problem may be looked at in three different ways:

- i. Given a scene, describe (linguistically) the spatial relations between the objects in the scene,
- ii. Given a scene and a spatial description of an object, find that object in the scene,
- iii. Given the spatial relations between the objects, construct a scene, locating the objects so as to satisfy those spatial relations (this is the "layout" problem).

This work concentrates on the first two problems, although the resulting definitions of spatial relations will be useful for the "layout" problem.

## SPATIAL RELATIONS AMONG FUZZY SUBSETS

Spatial relationships between regions in an image play important role in scene understanding. Humans are able to quickly ascertain the relationship between two objects, for example "B is to the right of A", or "B is in front of A", but this has turned out to be a somewhat illusive task for automation [5, 6, 7].

When the objects in a scene are represented by crisp sets, the all-or-nothing definitions of the subsets actually add to the problem of generating such relational descriptions. It is our belief that definitions of spatial relationships based on fuzzy set theory, coupled with fuzzy segmentation will yield realistic results.

### The Idea of Projections

This work proposes an initial approach at defining spatial relationships among fuzzy subsets of the image plane.

The idea is to project the fuzzy subsets onto two orthogonal coordinate axes and to utilize fuzzy dominance relations to capture the approximate relationships.

Let  $A$  be a fuzzy subset of an image. Then  $A \subseteq U \times V$ , where  $U$  is the first spatial coordinate axis and  $V$  is the second one. In our case, both  $U$  and  $V$  are subsets of the reals (assumed to be the interval  $[0, 1]$  for convenience). Then  $\mu_A(x, y)$  is a fuzzy relation in  $U \times V$ . The projection of  $A$  onto  $U$ , denoted  $A_U$

is that fuzzy subset of  $U$  given by

$$\mu_{A_U}(x) = \sup_y (\mu_A(x, y))$$

for each  $x \in U$ .

A similar equation defines the projection of  $A$  onto  $V$ , that is

$$\mu_{A_V}(y) = \sup_x (\mu_A(x, y))$$

for each  $y \in V$ .

For a fuzzy subset  $C$  of  $U$ , the  $\alpha$ -level set  $C^\alpha$  is defined by

$$C^\alpha = \{ x \in U \mid \mu_C(x) \geq \alpha \}$$

for  $\alpha \in [0, 1]$ .

When  $\alpha = 0$ , the inequality is usually considered to be strict and the  $C^\alpha$  is called the support of  $C$ .

### Definitions of Spatial Relations for Fuzzy Objects

Once the two fuzzy subsets  $A$  and  $B$  are projected onto  $U$  and  $V$  axes, methods must be defined to access their relative position.

In this paragraph we introduce definitions for spatial relations.

**Definition 1** : We say that subset  $A$  is to the right of subset  $B$  if the projection of  $A$  onto the  $U$  axis dominates the projection of  $B$ , while the projections onto the  $V$  axis are (ideally) identical. In other words  $\mu_V(\alpha)$  should stay near zero for all  $\alpha$  (especially for small  $\alpha$ ).

Similar definitions are suggested for all other spatial relations [13, 14].

The definitions are for antisymmetric and transitive relations, that is TO THE LEFT (RIGHT) OF, IN FRONT OF (BEHIND), ABOVE (BELOW), INSIDE (OUTSIDE). They are strict partial order relations (i.e. reflexive, antisymmetric and transitive) and every one has a semantic inverse.

### Separation Measure

Let  $A_U, B_U, A_V, B_V$  be the projections of  $A$  and  $B$  onto  $U$  and  $V$ , respectively. Since these projections are fuzzy numbers, their  $\alpha$ -level sets are intervals, i.e.,

$$A_U^\alpha = [A_U^{\alpha l}, A_U^{\alpha r}], \text{ etc.}$$

For the projections of A and B onto the U axis, the  $\alpha$ -separation of A and B is defined by

$$S_U^\alpha = (\bar{A}_U^\alpha - \bar{B}_U^\alpha)^2 / (W_{A_U}^\alpha + W_{B_U}^\alpha)^2$$

where

$$\bar{A}_U^\alpha = (A_U^{\alpha l} + A_U^{\alpha r}) / 2,$$

$$\bar{B}_U^\alpha = (B_U^{\alpha l} + B_U^{\alpha r}) / 2,$$

$$W_{A_U}^\alpha = (A_U^{\alpha r} - A_U^{\alpha l}) / 2, \text{ and}$$

$$W_{B_U}^\alpha = (B_U^{\alpha r} - B_U^{\alpha l}) / 2.$$

Now,  $S_U^\alpha$  is the ratio of the square of difference between the midpoints of the  $\alpha$ -level sets and the square of the sum of the half-widths of these intervals. Similar equations are used for the projection of A and B onto the V axis.

Definition 2 : We say that  $A_U$  and  $B_U$  are  $\alpha$ -separated if  $S_U^\alpha > 1$ .

Definition 3 : We say that  $A_U$  and  $B_U$  are  $\alpha$ -just separated if  $S_U^\alpha = 1$ .

Definition 4 : We say that  $A_U$  and  $B_U$  are  $\alpha$ -overlapping if  $S_U^\alpha < 1$ .

Theorem 1 : i)  $A_U$  and  $B_U$  are  $\alpha$ -separated if and only if  $A_U^{\alpha r} < B_U^{\alpha l}$ .

ii)  $A_U$  and  $B_U$  are  $\alpha$ -just separated if and only if  $A_U^{\alpha r} = B_U^{\alpha l}$ .

iii)  $A_U$  and  $B_U$  are  $\alpha$ -overlapping if and only if  $A_U^{\alpha r} > B_U^{\alpha l}$ .

The proof of the theorem can be found in [9].

The value of these definitions and theorems is two-fold. First, they incorporate the fuzziness in the description of image regions, i.e., they use fuzzy subsets of the plane. Second, they deal with the ambiguity of defining spatial relationships in the plane. By this we mean that it is possible that parts of the two sets can overlap (small  $\alpha$ ) and yet be well separated for large  $\alpha$ .

The values of  $S_U^\alpha$  can get arbitrarily large as the widths of the level set intervals get small. In order to create a fuzzy membership function, we will map the interval  $[0, \infty)$  into  $[0, 1]$  by an "S-shaped function" [15] as follows. For a given  $\alpha$ , suppose  $A_U^\alpha = [0, 0.2]$  and  $B_U^\alpha = [0.8, 1]$ . (Recall that we have scaled the domain of the image into the unit square). Then  $S_U^\alpha = 16$ . This amount of separation (or more) will be considered complete, i.e.,  $\mu(S_U^\alpha) = 1$  if

$S_U^\alpha \geq 16$ . Also we will require that  $\mu(0) = 0$ ,  $\mu(1) = 0.5$  and  $\mu'(16) = 0$ . Such a function is defined in our case by:

$$\mu(S) = \begin{cases} 0.5 S^2 & 0 \leq S \leq 1 \\ -0.0022 S^2 + 0.0711 S + 0.4311 & 1 < S \leq 16 \\ 1 & S > 16 \end{cases}$$

### The Model for Spatial Relationships

The model for given spatial relationships can now be defined from the fuzzy subsets  $\mu_U$  and  $\mu_V$  of  $[0, 1]$ . For example, to model the relationship "A IS TO THE RIGHT OF B", we would like the projection of A onto the U axis to dominate that of B; whereas the projections should (ideally) be identical on the V axis. That is,  $\mu_V(\alpha)$  should stay near zero for all  $\alpha$  (especially for small  $\alpha$ ). Similar observations can be made for "ABOVE", and "BELOW".

Instead of dealing with two fuzzy subsets,  $\mu_U$  and  $\mu_V$  can be combined into a single set from which the relationship can be determined. Fuzzy set theory offers an infinite number of aggregation operators, which, given two pieces of evidence (values in  $[0, 1]$ ) can produce essentially any composite value between 0 and 1, depending on the type of connective and the parameters chosen. Union operators produce values greater than or equal to the maximum of the two numbers; intersection operators give a result less than or equal to the minimum; and generalized means fill the gap between the minimum and

maximum [11].

"TO THE RIGHT OF" should therefore be a combination of  $\mu_U$  and the complement of  $\mu_V$  since its large values signify that the level sets of A are "above or below" those of B.

For the experiments described in the next paragraph, we chose a generalized mean

$$m(\mu_U, \mu_V) = [W \mu_U^P + (1 - W) (1 - \mu_V)^P]^{1/P}$$

as the aggregation connective [16]. In this way, higher weight can be associated with the horizontal component with decreased compensation as the level sets diverge vertically. Note, that if  $P \rightarrow \infty$ , then we have [11]:

$$\lim_{P \rightarrow \infty} m(\mu_U, \mu_V) = \max(\mu_U, \mu_V).$$

Either the two fuzzy sets  $\mu_U$  and  $(1 - \mu_V)$  or the single aggregated set  $m(\mu_U, \mu_V)$  can be used to define the relation "A IS TO RIGHT OF B". If a single value for the degree to which the two sets satisfy the relation is desired, we can construct a fuzzy measure from the sets - such as the integral of the fuzzy number, or the output of an ordered weighted average (OWA) [12]. An alternate approach is to use the curves directly to define a linguistic assessment of the relation. Here, it is necessary to define fuzzy sets representing terms used in the relation, such as "to the right of", "somewhat to the right of", "barely to the right of", "very to the right of", etc. These sets could be defined by the designer of the system, or perhaps, by utilizing a group of humans to give relative comparisons of a set of examples. The actual curve is then matched to the closest term available to give the linguistic assessment. This process is known as linguistic approximation [13].

### Results of Sample Systems

All the definitions and theorems listed above were tested using simulated data on a computer workstation. Fuzzy subsets with two-sided drum like shaped membership functions on projections were used. The experiments were as follows. Let us consider an image containing two fuzzy subsets A and B whose

membership functions are identical gaussians, but with different mean locations. The set B will be fixed with mean (0.5, 0.5). Table 1 shows the fuzzy set  $\mu_U$  generated from eight choices of locations for the mean of A (assume that the V coordinate for the mean is 0.5). As can be seen, as the set A moves to the right, the fuzzy set  $\mu_U$  increases for all  $\alpha$ . Recall that the value  $\mu_U(\alpha) = 0.5$  represents the just separated condition. The seven  $\alpha$ -values are 0.011, 0.135, 0.258, 0.606, 0.796, 0.882, 0.923. They were chosen in order to get the following ranges from the mean of gaussian functions:  $-0.4\sigma$ ,  $-0.5\sigma$ ,  $-0.6745\sigma$ ,  $-\sigma$ ,  $-1.645\sigma$ ,  $-2\sigma$ ,  $-3\sigma$ , where  $\sigma$  is a standard deviation.

Mean of Projections of A	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$
0.525	.001	.002	.003	.008	.017	.031	.049
0.550	.014	.031	.046	.125	.275	.500	.516
0.575	.070	.158	.234	.508	.543	.579	.623
0.600	.222	.500	.514	.564	.622	.680	.731
0.625	.503	.536	.558	.631	.712	.788	.851
0.650	.532	.580	.609	.706	.806	.891	.950
0.675	.567	.628	.665	.783	.893	.968	.998
0.700	.604	.680	.724	.857	.962	1.00	1.00

Table 1. Membership functions generated from the projection of A onto U axis.

Since the projections onto V for these sets are the same as the projections onto U, the fuzzy sets from Table 1 can be used to simulate other placings of A relative to B, e.g., to the northeast or southeast. Table 2 shows four cases for the placement of the center of set A along with the aggregated fuzzy set generated from both projections. Generalized mean with  $W = 0.73$  and  $P = 2$  was used. The first case represents a set A which is east of B. Here, the combined values are larger than those for the U projections only. In fact, even the smallest  $\alpha$  (0.011) gives rise to a membership larger than 0.5 (the just separated crossover point). In case 2, the set A has moved to the north east of B. The movement north effectively decreases the membership in the fuzzy set "A is to the right of B". Cases 3 and 4 depict the situation where A is directly above B. As the centers move further apart, the membership drops dramatically.

	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$
(.6, .5)							
$\mu_U$	.222	.500	.514	.564	.622	.680	.731
$1 - \mu_V$	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$m(\mu_U, \mu_V)$	0.54	0.66	0.67	0.70	0.73	0.77	0.81
(.6, .6)							
$\mu_U$	.222	.500	.514	.564	.622	.680	.731
$1 - \mu_V$	.778	.500	.486	.436	.378	.320	.269
$m(\mu_U, \mu_V)$	0.43	0.50	0.51	0.53	0.57	0.61	0.65
(.5, .6)							
$\mu_U$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$1 - \mu_V$	.778	.500	.486	.436	.378	.320	.269
$m(\mu_U, \mu_V)$	0.39	0.25	0.24	0.22	0.19	0.16	0.13
(.5, .7)							
$\mu_U$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$1 - \mu_V$	.396	.320	.276	.143	.038	0.00	0.00
$m(\mu_U, \mu_V)$	0.20	0.16	0.14	0.07	0.02	0.00	0.00

Table 2. Combined membership function for the relation "A is to the right of B" ( $W = 0.75, P = 2$ ).

If we change either the weight  $W$  or the exponent  $P$ , we can alter the shape of the resultant fuzzy set. For more details see [9].

## Summary and Conclusions

A new approach, based on the concept of dominance in fuzzy set theory, for modelling spatial relationships among fuzzy subsets of an image has been proposed. Simulation results were presented to corroborate the theory and demonstrate the power of the approach for image description.

## REFERENCES

- [1]. Huntsberger T., Representation of Uncertainty in Low Level Vision, IEEE Transactions on Computers, Vol. 235, 2, 1986, p.145.
- [2]. Huntsberger T., Jacobs C., Cannon R., Iterative Fuzzy Image Segmentation, Pattern Recognition, Vol. 18, 1985, pp.131-138.
- [3]. Keller J. M., Carpenter C. L., Image Segmentation in the Presence of Uncertainty, International Journal of Intelligent Systems, Vol. 5, 1990 pp. 193-208.
- [4]. Pal S. K., Rosenfeld A., Image Enhancement and Thresholding by Optimization of Fuzzy Compactness, Pattern Recognition Letters, 7, 1988, pp.77-86.
- [5]. Freeman J., The Modelling of Spatial Relations, Computer Graphics and Image Processing, 4, 1975, pp. 156-171.
- [6]. Huttenlocher J., Constructing Spatial Images: A Strategy in Reasoning, Psychology Review, 75, 1968, pp. 550-560.
- [7]. Koczy L. T., On the Description of Relative Position of Fuzzy Patterns, Pattern Recognition Letters, 8, 1988, pp. 21-28.
- [8]. Sztandera L. M., Keller J. M., Spatial Relations Among Fuzzy Subsets of an Image, Proceedings of the 1st International Symposium on Uncertainty Modelling and Analysis, University of Maryland, Maryland, 1990, pp. 207-211.
- [9]. Sztandera L. M., Spatial Relations Among Fuzzy Subsets of an Image, Technical Report, University of Missouri-Columbia, Columbia, 1990.
- [10]. Zadeh L. A., Calculus of Fuzzy Restriction, in Zadeh L. A., Fu K. S., Tanaka K., Shimura M.(Eds.), Fuzzy Sets and Their Applications to Cognitive and Decision Process, Academic Press, New York, 1975, pp. 1-39.
- [11]. Klir G., Folger T., Fuzzy Sets, Uncertainty, and Information, Prentice Hall, 1988.
- [12]. Yager R. R., On Ordered Weighted Averaging Aggregation Operations in Multicriteria Decision Making, IEEE Transactions on Systems, Man and Cybernetics, Vol. 18, 1, 1988, pp. 183-190.
- [13]. Zadeh L. A., The Concept of a Linguistic Variable and Its Application in Approximate Reasoning, Information Sciences, 8, 1975, pp. 301-357.

512-67  
161812

N93-29528

## IMAGE SEGMENTATION USING FUZZY LVQ CLUSTERING NETWORKS

Eric Chen-Kuo Tsao, James C. Bezdek\* and Nikhil R. Pal  
Division of Computer Science  
The University of West Florida  
Pensacola, FL 32514

\*Research Supported by NSF Grant IRI-9003252

### ABSTRACT

In this note we formulate image segmentation as a clustering problem. Feature vectors extracted from a raw image are clustered into subregions, thereby segmenting the image. A fuzzy generalization of Kohonen learning vector quantization (LVQ) which integrates the Fuzzy c-Means (FCM) model with the learning rate and updating strategies of the LVQ is used for this task. This network, which segments images in an unsupervised manner, is thus related to the FCM optimization problem. Numerical examples on photographic and magnetic resonance images are given to illustrate this approach to image segmentation.

### 1. INTRODUCTION

Image segmentation divides an image into regions with uniform and homogeneous attributes such as gray tone or texture [1]. Roughly speaking, conventional segmentation algorithms can be divided into two classes: region-based schemes, wherein areas of images with homogeneous properties are found, which in turn gives region boundaries [2-4]; and edge-based schemes, where local discontinuities are detected first, and then connected to form longer, hopefully complete, boundaries [5]. Image segmentation should result in regions that cover semantically distinct visual entities and is a crucial step for subsequent recognition or interpretation tasks.

Several image segmentation methods based on Markov Random Fields (MRFs) have been proposed. The basic idea is to model spatial interaction of the image features by a MRF which is a probability distribution defined over a discrete random field. Hongo *et al.* [6] proposed a "multiple level multiple resolution MRF" to detect the edges which was an extension of the work of Geman and Geman [7]. This model incorporates *a priori* knowledge about global structures in images, but can be implemented in a local (and parallel) mode. Three algorithms (simulated annealing, iterative conditional modes, and maximization of posterior marginals) are compared in [8]; all use MRF models to include prior contextual information. Most of these approaches use an energy functional to guide image segmentation and numerical schemes for minimization of the energy functional. However, the search procedure for a global minimum (optimal solution) is usually time consuming. Moreover, edge-based segmentation schemes usually need a linking procedure to connect broken edges in order to make image subregions that have closed boundaries. Recently, several attempts to apply computational neural network architectures to image segmentation have been made. For example, edge detection has been formulated in the context of an energy-minimizing model by eliminating weak boundaries and small segments [9]; and also as a fuzzy feed-forward computational neural network problem [10]. A neural network system capable of detecting potential edges in various orientations that uses simulated and mean field annealing is discussed in [11].

In this note we propose using a new family of clustering algorithms called Fuzzy Learning vector Quantization (FLVQ) for image segmentation. FLVQ is a partial integration of Fuzzy c-Means (FCM) and Kohonen clustering networks (LVQs). The block diagram of the process is shown in Fig. 1. Unlabeled feature vectors (one for each pixel) are first extracted from an image. Then FLVQ clusters these feature vectors to get cluster centers. Each cluster center is regarded as a prototype (or vector quantizer) of some subregion of the image. Finally, each pixel feature vector is compared to the cluster centers, and is assigned a constant value corresponding to the closest cluster center. Note that the number of constant values is the same as the number of clusters.



Figure 1. FLVQ Image Segmentation: Overall Architecture.

The remainder of this paper is organized as follows. In the next section, we briefly review the FCM, LVQ and FLVQ algorithms. In Section 3, experimental segmentation results on photographic and Magnetic Resonance images are reported. Section 4 contains a discussion, conclusions, and some ideas for future research.

## 2. KOHONEN CLUSTERING NETWORKS

Many classical clustering algorithms can be found in the texts of Duda and Hart [12], Hartigan [13], and Jain and Dubes [14]. In [15] Lippman suggested that Kohonen's learning vector quantization (LVQ) [16] is closely related to the sequential Hard c-Means (HCM) algorithm. Fuzzy c-Means (FCM) is a well known generalization of HCM [17,18]. Since HCM/FCM are optimization procedures, whereas LVQ is not, integration of FCM and LVQ is one way to address several problems of LVQs while simultaneously attacking the general problem of how the two families are related. Huntsberger and Ajjimarangsee [19] first considered this approach, and their idea was extended in [20] to the FLVQ algorithms described below.

Let  $c$  be an integer,  $1 < c < n$ , and let  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  denote a set of  $n$  feature vectors in  $\mathcal{R}^p$ .  $X$  is numerical object data, the  $j$ -th object has vector  $\mathbf{x}_j$  as its numerical representation, and  $x_{jk}$  is the  $k$ -th characteristic (or feature) associated with object  $j$ . Given  $X$ , we say that  $c$  fuzzy subsets  $\{u_i : X \rightarrow [0,1]\}$  are a constrained fuzzy  $c$ -partition of  $X$  in case the  $cn$  values  $\{u_{ik} = u_i(\mathbf{x}_k), 1 \leq k \leq n, 1 \leq i \leq c\}$  satisfy three conditions:

$$0 \leq u_{ik} \leq 1 \text{ for all } i, k ; \quad (1a)$$

$$\sum_i u_{ik} = 1 \text{ for all } k ; \quad (1b)$$

$$0 < \sum_i u_{ik} < n \forall i. \quad (1c)$$

Here  $u_{ik}$  is interpreted as the *membership* of  $\mathbf{x}_k$  in the  $i$ -th partitioning subset (cluster) of  $X$ . If all of the  $u_{ik}$ 's are in  $\{1,0\}$ ,  $U = [u_{ik}]$  is a conventional (crisp, hard)  $c$ -partition of  $X$ . The most well known objective function for clustering in  $X$  is the classical within groups sum of squared errors function, defined as :

$$J_1(U, \mathbf{v} ; X) = \sum_i \sum_k u_{ik} \|\mathbf{x}_k - \mathbf{v}_i\|^2, \quad (2)$$

where  $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c)$  is a vector of (unknown) cluster centers (weights, prototypes, or vector quantizers),  $\mathbf{v}_i \in \mathcal{R}^p$  for  $1 \leq i \leq c$ , and  $U$  is a hard or conventional  $c$ -partition of  $X$ . Optimal partitions  $U^*$  of  $X$  are taken from pairs  $(U^*, \mathbf{v}^*)$  that are "local minimizers" of  $J_1$ . Dunn [18] first generalized (2) for  $m=2$ , and subsequently, Bezdek [17] generalized (2) to the infinite family written as:

$$J_m(U, \mathbf{v} ; X) = \sum_i \sum_k u_{ik}^m \|\mathbf{x}_k - \mathbf{v}_i\|_A^2, \quad (3)$$

where  $m \in [1, \infty)$  is a weighting exponent on each fuzzy membership,  $U$  is a fuzzy  $c$ -partition of  $X$ ,  $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c)$  are cluster centers in  $\mathcal{R}^p$ ,  $A =$  any positive definite  $(p \times p)$  matrix, and  $\|\mathbf{x}_k - \mathbf{v}_i\|_A = (\mathbf{x}_k - \mathbf{v}_i)^T A (\mathbf{x}_k - \mathbf{v}_i)$  is the distance (in the  $A$  norm) from  $\mathbf{x}_k$  to  $\mathbf{v}_i$ . Conditions that are necessary for

extrema of  $J_1$  and  $J_m$  follow : **Hard c-Means (HCM) Theorem [17]**  $(U, \mathbf{v})$  may minimize  $\sum u_{ik} (\|\mathbf{x}_k - \mathbf{v}_i\|_A)^2$  only if :

$$u_{ik} = \begin{cases} 1; & (\|\mathbf{x}_k - \mathbf{v}_i\|_A)^2 = \min_j \{(\|\mathbf{x}_k - \mathbf{v}_j\|_A)^2\} \\ 0; & \text{otherwise} \end{cases} \quad (4a)$$

$$\mathbf{v}_i = \sum u_{ik} \mathbf{x}_k / \sum u_{ik} \quad (4b)$$

In the context of image segmentation, equation (4a) will be used to assign each (pixel) vector  $\mathbf{x}_k$  to its closest prototype  $\mathbf{v}_i$ ; this is the essence of our segmentation scheme. Note that the HCM produces a partition  $U$  that contains hard clusters. The well known generalization of HCM is contained in the following: **Fuzzy c-Means (FCM) Theorem [17]** Assume  $\|\mathbf{x}_k - \mathbf{v}_j\|_A^2 > 0, \forall j, k$  at each iteration of (5):  $(U, \mathbf{v})$  may minimize  $\sum u_{ik}^m (\|\mathbf{x}_k - \mathbf{v}_i\|_A)^2$  for  $m > 1$  only if :

$$u_{ik} = \left( \sum (\|\mathbf{x}_k - \mathbf{v}_i\|_A / \|\mathbf{x}_k - \mathbf{v}_j\|_A)^{2/(m-1)} \right)^{-1} \quad (5a)$$

$$\mathbf{v}_i = \sum (u_{ik})^m \mathbf{x}_k / \sum (u_{ik})^m \quad (5b)$$

Conditions (5)  $\rightarrow$  (4) and  $J_m \rightarrow J_1$  as  $m \rightarrow 1$  from above. The FCM (HCM) algorithms are iterative procedures for approximately minimizing  $J_m$  ( $J_1$ ) by Picard iteration through (5) or (4), respectively. C-Means algorithms are non-sequential algorithms: updates on the weights  $\{\mathbf{v}_{i,t}\}$  are performed after each pass through  $X$ . Thus, iterate sequence  $\{\mathbf{v}_{i,t}\}$  is independent of the sequence of the data labels. The parameter ( $m$ ) essentially controls the "amount of fuzziness" in  $U$ . As  $m \rightarrow \infty$ ,  $u_{ik,t} \rightarrow 1/c$ ; when  $m \rightarrow^+ 1$ ,  $u_{ik,t} \rightarrow 1$  or  $0$ .

Kohonen clustering networks (LVQs) are unsupervised schemes which find the "best" set of prototypes (for hard clusters) in an iterative, sequential manner. The structure of LVQ consists of two layers: an input (fanout) layer, and an output (competitive) layer as shown in Fig. 2. The edges that connect the  $p$  input nodes to the  $c$  output nodes do not have "weights" attached to them, as, for example, in a feed forward network architecture. Instead, each output node has a prototype (vector quantizer) attached to it, and it is this set of network weight vectors that are adjusted during learning. A formal description of LVQ is given below. There are other versions of LVQ; this one is usually regarded as the "standard" form.

### The LVQ Clustering Algorithm [16]

LVQ1. Given unlabeled data set  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset \mathcal{R}^p$ . Fix  $c, T$ , and  $\epsilon > 0$ .

LVQ2. Initialize  $\mathbf{V}_0 = (\mathbf{v}_{1,0}, \dots, \mathbf{v}_{c,0}) \in \mathcal{R}^{cp}$ , and learning rate  $\alpha_0 \in (1, 0)$ .

LVQ3. For  $t = 1, 2, \dots, T$ ;

For  $k = 1, 2, \dots, n$ :

a. Find  $\|\mathbf{x}_k - \mathbf{v}_{i,t-1}\| = \min_{1 \leq j \leq c} \{\|\mathbf{x}_k - \mathbf{v}_{j,t-1}\|\}$ . (6)

b. Update the winner :  $\mathbf{v}_{i,t} = \mathbf{v}_{i,t-1} + \alpha_t (\mathbf{x}_k - \mathbf{v}_{i,t-1})$  (7)

Next  $k$ .

d. Apply the 1-NP (nearest prototype) rule to the data :

$$u_{LVQ_k} = \begin{cases} 1; & \|\mathbf{x}_k - \mathbf{v}_i\| \leq \|\mathbf{x}_k - \mathbf{v}_j\|, 1 \leq j \leq c, j \neq i \\ 0; & \text{otherwise} \end{cases}, 1 \leq i \leq c \text{ and } 1 \leq k \leq n. \quad (8)$$

e. Compute  $E_t = \|\mathbf{V}_t - \mathbf{V}_{t-1}\|_1 = \sum_{r=1}^c \|\mathbf{v}_{r,t} - \mathbf{v}_{r,t-1}\|_1 = \sum_{k=1}^n \sum_{r=1}^c |v_{rk,t} - v_{rk,t-1}|$ .

f. If  $E_t \leq \epsilon$  stop; Else adjust learning rate  $\alpha_t$ ;

Next t

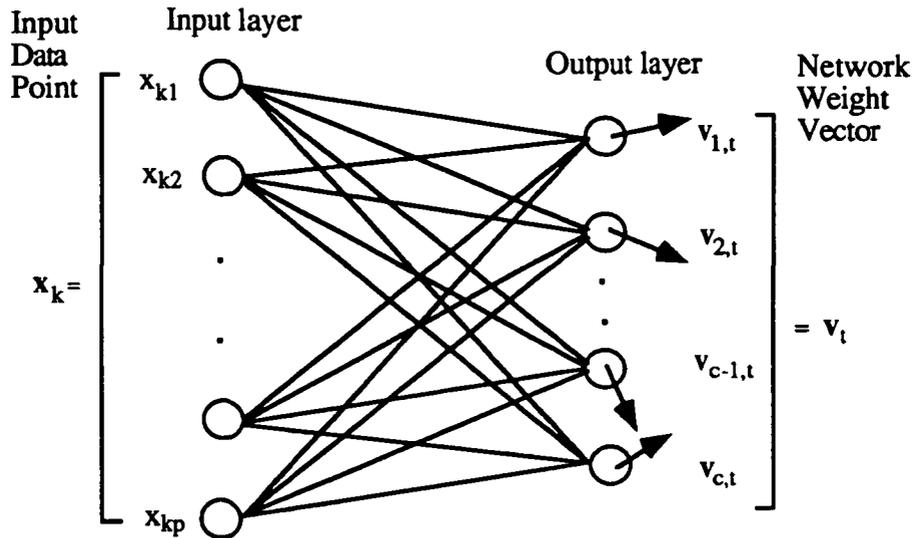


Figure 2. The structure of a Kohonen clustering network.

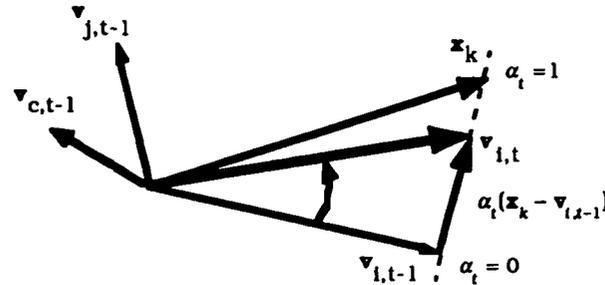
The numbers  $U_{LVQ} = [u_{LVQ_{jk}}]$  at (8) are a  $c \times n$  matrix that almost always (constraint (1c) may not be satisfied) define a hard  $c$ -partition of  $X$  using the 1-NP classifier assignment rule at (4). Our inclusion of computation of the hard 1-NP  $c$ -partition of  $X$  at the end of each pass through the data (step LVQ3.d) is **not** part of the LVQ algorithm - that is, the LVQ iterate sequence does not depend on cycling through  $U$ 's. Ordinarily this computation is done once, non-iteratively, outside and after termination of LVQ. Note that LVQ uses the Euclidean distance in step LVQ3.a. This choice corresponds roughly to the update rule shown in (7), since  $\nabla_{\mathbf{v}} (\|\mathbf{x} - \mathbf{v}\|_l^2) = -2l(\mathbf{x} - \mathbf{v}) = -2(\mathbf{x} - \mathbf{v})$ . The origin of this rule assumes that each  $\mathbf{x} \in \mathcal{R}^p$  is distributed according to a probability density function  $f(\mathbf{x})$ . LVQ's objective is to find a set of  $\mathbf{v}_i$ 's to minimize the expected value of the square of the discretization error :

$$E(\|\mathbf{x} - \mathbf{v}_i\|_l^2) = \int \dots \int_{\mathcal{R}^p} \|\mathbf{x} - \mathbf{v}_i\|_l^2 f(\mathbf{x}) d\mathbf{x} \quad (9)$$

In this expression  $\mathbf{v}_i$  is the winning prototype for each  $\mathbf{x}$ , and will of course vary as  $\mathbf{x}$  ranges over  $\mathcal{R}^p$ . A sample function of this optimization problem is  $e = \|\mathbf{x} - \mathbf{v}_i\|_l^2$ . An optimal set of  $\mathbf{v}_i$ 's can be approximated by applying local gradient descent to a finite set of samples drawn from  $f$ . The extant theory for this scheme is contained in [21], which states that LVQ converges in the sense that the prototypes  $\mathbf{V}_t = (\mathbf{v}_{1,t}, \mathbf{v}_{2,t}, \dots, \mathbf{v}_{c,t})$  generated by the LVQ iterate sequence converge, i.e.,  $\{\mathbf{V}_t\} \xrightarrow{t \rightarrow \infty} \hat{\mathbf{V}}$ , provided two conditions are met by the sequence  $\{\alpha_t\}$  of learning rates used in (7) :

$$\sum_{t=0}^{\infty} \alpha_t = \infty \quad ; \quad \text{and} \quad \sum_{t=0}^{\infty} \alpha_t^2 < \infty \quad . \quad (10)$$

One choice for the learning rates that satisfies these conditions is the harmonic sequence  $\alpha_t = 1/t$  for  $t \geq 1$ ;  $\alpha_0 \in (0,1)$ . Kohonen has shown that (under some assumptions) steepest descent optimization of the average expected error function (9) is possible, and leads to update rule (7). The update scheme at (7) has the simple geometric interpretation shown in Figure 3.



**Figure 3. Updating the winning LVQ Prototype.**

The winning prototype at iteration  $t$ ,  $v_{i,t-1}$ , is simply rotated towards the current data point by moving along the vector  $(x_k - v_{i,t-1})$  which connects it to  $x_k$ . The amount of shift depends on the value of a "learning rate" parameter  $\alpha_t$ , which varies from 0 to 1. As seen in Figure 3, there is no update if  $\alpha_t=0$ , and when  $\alpha_t=1$ ,  $v_{i,t}$  becomes  $x_k$  ( $v_{i,t}$  is just a convex combination of  $x_k$  and  $v_{i,t-1}$ ). This process continues until termination via LVQ3.f, when the terminal prototypes yield a "best" hard c-partition of  $X$  via (8).

**Comments on LVQ:** (1) Kohonen in [21] mentions that LVQ converges to a unique limit if and only if conditions (10) are satisfied. However, nothing was said about what sort or type of points the final weight vectors produced by LVQ are. Since LVQ does not model a well defined property of clusters (in fact, LVQ does not maintain a partition of the data at all), the fact that  $\{v_t\} \xrightarrow{t \rightarrow \infty} \hat{V}$

does not insure that the limit vector  $\hat{V}$  is a good set of prototypes in the sense of representation of clusters or clustering tendencies. (2) The termination strategy at LVQ3.e is based on small successive changes in the cluster centers. This method of algorithmic control offers the best set of centroids for compact representation (quantization) of the data in each cluster. However, LVQ seldom terminates in less than, say, 20,000 iterates unless  $\alpha_t \rightarrow 0$ : this forces it to stop because successive iterates are necessarily close. (3) LVQ often runs to its iterate limit, and sometimes passes the optimal (clustering) solution in terms of minimal apparent label error rate. This is called the "over-training" phenomenon in the neural network literature.

Huntsberger and Ajjimarangsee [19] combined the 1-NP rule at (4) with Self-Organizing Feature Maps (SOFMs) to develop clustering algorithms. Algorithm 1 in [19] is the SOFM algorithm with an additional layer of neurons that does not participate in weight updating. After the self-organizing network terminates, the additional layer, for each input, finds the weight vector (prototype) closest to it and assigns the input data point to that class. A second algorithm in their paper used the necessary conditions for FCM to assign a membership value in  $[0,1]$  to each data point for each of the  $c$  classes. Specifically, Huntsberger and Ajjimarangsee suggested fuzzification of SOFM by replacing the learning rates  $\{\alpha_{ik,t}\}$  usually found in rules such as (7) with fuzzy membership values  $\{u_{ik,t}\}$  computed with the FCM formula [17]:

$$\alpha_{ik,t} = u_{ik,t} = \left( \frac{D_{ikA,t}}{\sum_{j=1}^c D_{jkA,t}} \right)^{\frac{-2}{m-1}} \quad , \quad m > 1, \quad (11)$$

where  $D_{ik,t} = \|\mathbf{x}_k - \mathbf{v}_{i,t}\|_A$ . Numerical results reported in Huntsberger and Ajjimarangsee suggest that in many cases their algorithms and standard LVQ produce very similar answers. Their scheme was a partial integration of LVQ with FCM that showed some interesting results. However, it fell short of realizing a model for fuzzy LVQ clustering; and no properties regarding terminal points or convergence were established. Moreover, since the objective of LVQ is to find cluster centers (prototypes) in  $\mathcal{R}^P$ , the need for and use of the topological ordering idea of (images of) the weight vectors in display space is not well justified. Consequently, the approach taken in [19] seems to mix two objectives, feature mapping and clustering, and the overall methodology is difficult to interpret in either sense.

Integration of FCM with LVQ can be more fully realized by defining the learning rate for Kohonen updating as :

$$\alpha_{ik,t} = (u_{ik,t})^{m_t} = \left( \frac{\sum_{j=1}^c D_{jk,t}}{D_{jk,t}} \right)^{\frac{-2m_t}{m_t-1}}, \quad \text{where} \quad (12a)$$

$$m_t = m_0 + t[(m_f - m_0) / T] = m_0 + t\Delta m; \quad m_f, m_0 \geq 1; \quad t=1,2,\dots,T. \quad (12b)$$

$m_t$  replaces the (fixed) parameter  $m$  in (11). This results in three families of Fuzzy LVQ or FLVQ algorithms, the cases arising by different treatments of parameter  $m_t$ . In particular, for  $t \in \{1,2,\dots,T\}$ , we have three cases depending on choice of the initial ( $m_0$ ) and final ( $m_f$ ) values of  $m$ :

$$1. \quad m_0 > m_f \Rightarrow \{m_t\} \downarrow m_f \quad : \text{Descending FLVQ} \quad (13a)$$

$$2. \quad m_0 < m_f \Rightarrow \{m_t\} \uparrow m_f \quad : \text{Ascending FLVQ} \quad (13b)$$

$$3. \quad m_0 = m_f \Rightarrow m_t \equiv m_0 \equiv m \quad : \text{FLVQ} \equiv \text{FCM} \quad (13c)$$

Cases 1 and 3 are discussed at length in [20]. Equation (13c) asserts that when  $m_0 = m_f$ , FLVQ reverts to FCM; this results from defining the learning rates via (12a), and using them in the update rule for the prototypes shown in FLVQ3.b below. We provide a formal description of FLVQ :

### Fuzzy LVQ (FLVQ) [20]

FLVQ1. Given unlabeled data set  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ . Fix  $c, T, \|\cdot\|_A$  and  $\epsilon > 0$ .

FLVQ2. initialize  $\mathbf{v}_0 = (\mathbf{v}_{1,0}, \dots, \mathbf{v}_{c,0}) \in \mathcal{R}^{cp}$ . Choose  $m_0, m_f \geq 1$ .

FLVQ3. For  $t = 1, 2, \dots, T$ .

a. Compute all (cn) learning rates  $\{\alpha_{ik,t}\}$  with (12).

b. Update all (c) weight vectors  $\{\mathbf{v}_{i,t}\}$  with

$$\mathbf{v}_{i,t} = \mathbf{v}_{i,t-1} + \frac{\sum_{k=1}^n \alpha_{ik,t} (\mathbf{x}_k - \mathbf{v}_{i,t-1})}{\sum_{s=1}^n \alpha_{is,t}}$$

c. Compute  $E_t = \|\mathbf{v}_t - \mathbf{v}_{t-1}\| = \sum_{i=1}^c \|\mathbf{v}_{i,t} - \mathbf{v}_{i,t-1}\|$ .

d. If  $E_t \leq \epsilon$  stop; Else

Next  $t$ .

---

Observe that FLVQ is not a direct fuzzy generalization of LVQ because it does not revert to LVQ in case all of the  $u_{ik,t}$ 's are either 0 or 1 (the crisp case). Instead, if  $m_0 = m_f = 1$ , FCM reverts to HCM, and the HCM prototype update formula, which is driven by finding unique winners, as in LVQ, is a different formula than (7). Nonetheless, FLVQ is perhaps the closest possible link between LVQ and

c-Means type algorithms. For fixed  $c$ ,  $\{\mathbf{v}_{i,t}\}$  and  $m_t$ , the learning rates  $\alpha_{ik,t} = (u_{ik,t})^{m_t}$  at (12a) satisfy the following :

$$\alpha_{ik,t} = (u_{ik,t})^{m_t} = \left( \frac{\kappa}{D_{ik,t}} \right)^{\frac{2m_t}{m_t-1}} \quad (14)$$

where  $\kappa$  is a positive constant. Apparently the contribution of  $\mathbf{x}_k$  to the next update of the node weights is inversely proportional to their distances from it. The "winner" is the  $\mathbf{v}_{i,t-1}$  closest to  $\mathbf{x}_k$ , and it will be moved further along the line connecting  $\mathbf{v}_{i,t-1}$  to  $\mathbf{x}_k$  than any of the other weight vectors. Since  $\sum u_{ik,t} = 1 \Rightarrow \sum \alpha_{ik,t} \leq 1$ , this amounts to distributing partial updates across all  $c$  nodes for each  $\mathbf{x}_k \in X$ . This is in sharp contrast to LVQ, where only the winner is updated for each data point.

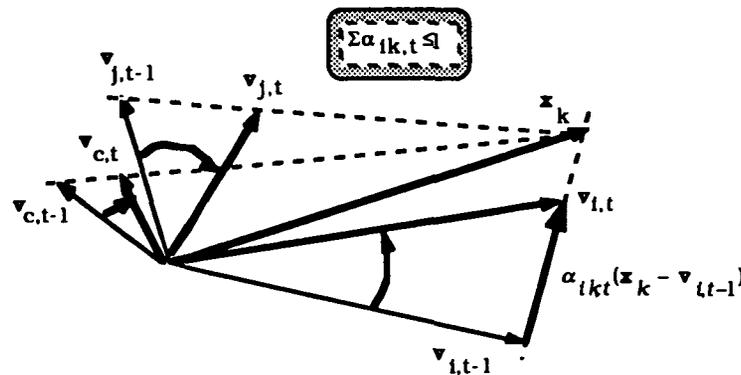


Figure 4. Updating Feature Space Prototypes in FLVQ Clustering Nets.

Figure 4 illustrates the update geometry of FLVQ; note that *every* node is (potentially) updated at every iteration, and the sum of the learning rates is always less than or equal to 1, an added constraint on the overall movement of the  $c$  prototypes at each  $t$ . In *descending* FLVQ (13a), for large values of  $m_t$  (near  $m_0$ ), all  $c$  nodes are updated with lower individual learning rates, and as  $m_t \rightarrow m_f$ , more and more of the update is given to the "winner" node. In other words, the lateral distribution of learning rates is a function of  $t$ , which in the descending case "sharpens" at the winner node (for each  $\mathbf{x}_k$ ) as  $m_t \rightarrow m_f$ .

**Comments on FLVQ :** (1) In contradistinction to Huntsberger and Ajjimarangsee's approach, there is no need to choose an update neighborhood. Neighborhood control is automatic, and depends entirely on the relative geometry of the data and their prototypes. (2) Reduction of the learning coefficient with distance (either topological or in  $\mathcal{R}^p$ ) from the winner node is not required. Instead, reduction is done automatically and adaptively by the learning rules. (3) The greater the mismatch to the winner (i.e., the higher the quantization error), the *smaller* the impact to the weight vectors associated with other nodes (recall (14)). (4) The learning process attempts to minimize a well-defined objective function (stepwise). This procedure depends on generation of a fuzzy  $c$ -partition of the data, so it is an iterative clustering model - indeed, stepwise, it is exactly fuzzy  $c$ -means [20]. (5) Our termination strategy is based on small successive changes in the cluster centers. This method of algorithmic control offers the best set of centroids for compact representation (quantization) of the data in each cluster.

### 3. EXPERIMENTAL RESULTS

In this section we discuss an application of *ascending FLVQ* to segmentation of intensity and Magnetic Resonance (MR) images. Success of a clustering technique as a tool for image segmentation depends largely on the choice of useful feature vectors. We first discuss the application of FLVQ to segmentation of light intensity images, and then to MR images. For a digital intensity image, every pixel is usually represented by a feature vector derived from pixel statistics like the mean, standard deviation, edginess and so on, computed over a small neighborhood (window) about the pixel under consideration. In this note we illustrate FLVQ using very simple feature vectors obtained by arranging pixels into an array. For a  $p \times p$  ( $p$  is odd) neighborhood, the  $p^2$  pixels will be arranged into a linear array in a systematic manner, starting from the top left corner of the window. For example, for a  $3 \times 3$  window the feature vector corresponding to pixel  $(i,j)$ , with clockwise traversal of pixels from location  $(i-1,j-1)$ , takes the following form :

$$\mathbf{x}_{ij} = [f(i-1,j-1), f(i-1,j), f(i-1,j+1), f(i,j+1), f(i+1,j+1), f(i+1,j), f(i+1,j-1), f(i,j-1), f(i,j)]^T$$

This kind of feature vector has the advantage that it accounts for spatial details of local gray levels and requires no computation for feature vector generation. This choice has disadvantages as well. For example, permutation of the same set of gray values over a window will generate different feature vectors, which may in turn lead to different results. This problem can be circumvented by sorting the gray values. In this investigation, we implemented the FLVQ algorithm with three different window sizes;  $1 \times 1$ ,  $3 \times 3$  and  $5 \times 5$ . The computing protocols are summarized in Table 1.

	norm	c	$m_0$	$\Delta m$	$t_{max}$	$\epsilon$	iterations
Fig. 5(b)	Euclidean	6	1.05	0.2	200	0.05	17
Fig. 5(c)	Euclidean	6	1.05	0.2	200	0.05	24
Fig. 5(d)	Euclidean	6	1.05	0.2	200	0.05	29
Fig. 6(c)	Euclidean	6	1.05	0.2	200	0.05	22

Table 1. Protocols for the Computational Experiments



Fig. 5(a) An intensity image

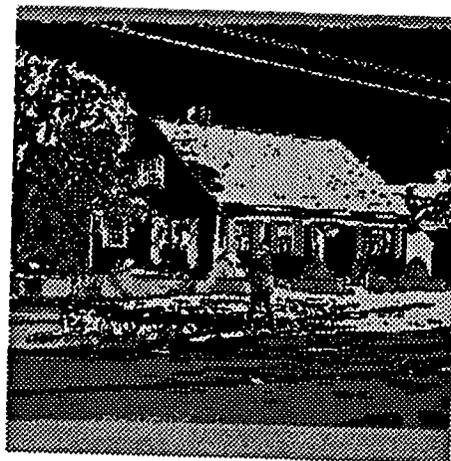


Fig. 5(b) Segmentation result using  $1 \times 1$

Figure 5(a) depicts the intensity image of a house. Figures 5 (b), (c) and (d) represent segmentation results produced by window of sizes  $1 \times 1$ ,  $3 \times 3$  and  $5 \times 5$ , respectively. Note that Figure 5(b) is too detailed, in a sense "noisy". This is so because the  $1 \times 1$  window does not take into account the spatial distribution of gray levels; in fact, in some sense, this is

equivalent to histogram thresholding. Comparison of Figure 5(c) with 5(b) reveals that the roof and the walls of the house are better segmented by the 3x3 window. On the other hand, Figure 5(d) contains more compact segmented regions; even the textured tree is segmented as compact homogeneous regions. This shows that too small a window may result in too many details, while too large a window may smooth out much relevant information. Probably a reasonably good compromise is a neighborhood of size 3x3.

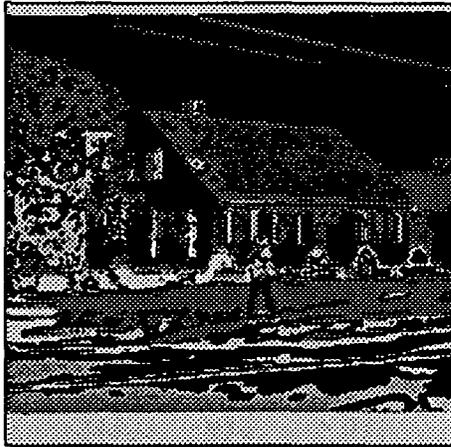


Fig. 5(c) Segmentation result using 3x3

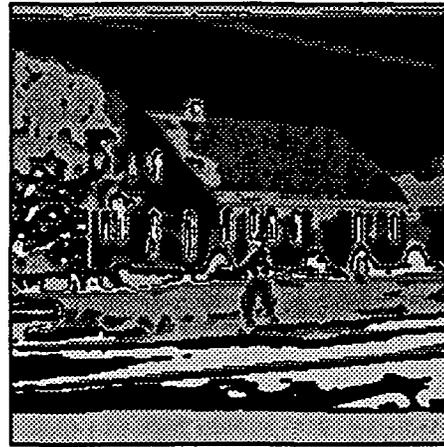


Fig. 5(d) Segmentation result using 5x5

If  $q$  images are correlated in the sense that they are perfectly registered because they are taken in different bands, pixel vectors of size  $q$  can be erected at each spatial site by simply aggregating the intensity across bands. This amounts to a multichannel version of the  $1 \times 1$  window. Magnetic Resonance Imagery, e.g. typically generates 3 bands, namely, T1 relaxation (spin lattice), T2 relaxation (transverse), and  $\rho$  (proton density). At pixel site  $(i,j)$ , MRI data can thus result in 3 dimensional pixel vectors, say  $\mathbf{x}_{ij} = (T1_{ij}, T2_{ij}, \rho_{ij})$ . This  $\mathbf{x}_{ij}$  can then be used a feature vector for segmentation of the MR image. Figures 6(a) and 6(b) show two bands ( $\rho$  and T2) of one physical slice of an human head. Fig. 6(c) depicts the segmentation obtained using FLVQ with the parameters shown in the last row of Table 1. It is well-known that comparison of image segmentation algorithms is not an easy task [8]. However, one of the most important criteria for performance evaluation is whether the algorithm can outline the desired or important components in the image. For instance, in Fig. 6(c), our segmentation delineates the white and gray matter tissue regions quite well.

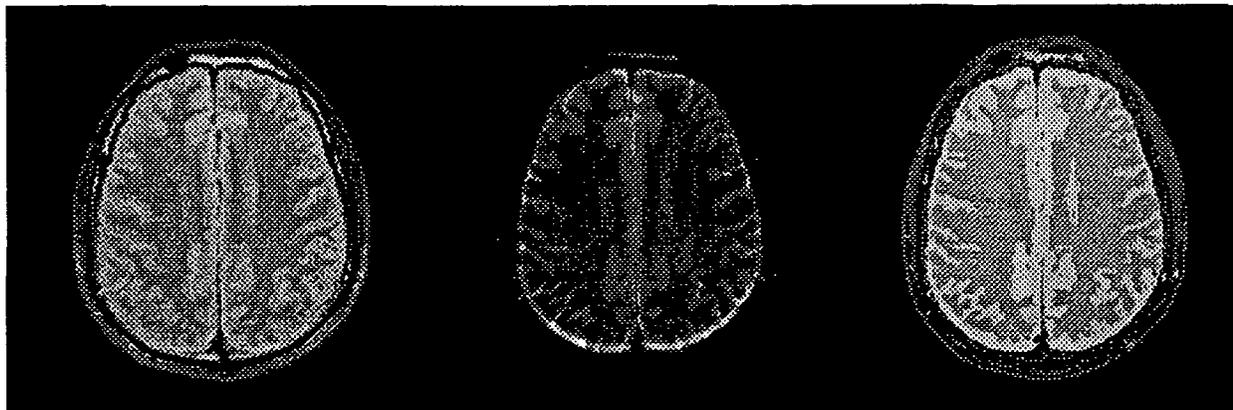


Fig. 6(a)  $\rho$  MR data

Fig. 6(b) T2 MR data

Fig. 6(c) FLVQ Segmentation

#### 4. CONCLUDING REMARKS

In this paper a family of Fuzzy generalization of LVQ (FLVQ) algorithms based on the integration of Fuzzy c-Means and Kohonen clustering networks have been used for image segmentation. FLVQ is non-sequential, unsupervised, and uses fuzzy membership values from FCM as learning rates. This yields automatic control of both the learning rate distribution and update neighborhood. Light intensity and MR images have been segmented using various feature extraction strategies; our results seem encouraging, but much remains to be done.

#### REFERENCES

- [1] R. M. Haralick and L. G. Shapiro, "Survey: image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol.29, pp.100-132, 1985.
- [2] B. Bhanu and B. A. Parvin, "Segmentation of natural scenes," *Pattern Recognition*, vol.20, pp.487-496, 1987.
- [3] J. R. Beveridge, J. Griffith, R. R. Kohler, A. R. Hanson, and E. M. Riseman, "Segmentation image using localized histogram and region merging," *International Journal of Computer Vision*, vol.2, pp.311-347, 1989.
- [4] T. Pavlidis and Y.-T. Liow, "Integrating region growing and edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.12, pp.225-233, 1990.
- [5] W. A. Perkins, "Area segmentation of image using edges points," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.2, pp.8-15, 1980.
- [6] S. Hongo, M. Kawato, T. Inui, and S. Miyake, "Contour extraction of image on parallel computer-local, parallel and stochastic algorithm which learns energy parameters," *Proceedings of the International Joint Conference on Neural Networks*, vol.1, pp. 161-168, 1990.
- [7] S. Geman and D. Geman, Stochastic relaxation, "Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.6, pp. 721-741, 1984.
- [8] R. C. Dubes, A. K. Jain, S. G. Nadabar and C. C. Chen, "MRF model-based algorithms for image segmentation," *Proceedings of the International Conference on Pattern Recognition*, pp. 808-814, 1990.
- [9] J. Shah, "Parameter estimation, multiscale representation and algorithms for energy-minimizing segmentation," *Proceedings of the International Conference on Pattern Recognition*, pp. 815-819, 1990.
- [10] D. Kerr and J. C. Bezdek, "Edge Detection Using Neural Networks," SPIE, 1992.
- [11] C. Cortes and J. A. Hertz, "A network system for image segmentation," *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 121-125, 1989.
- [12] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
- [13] J. Hartigan, *Clustering Algorithms*, Wiley, New York, 1975.
- [14] A. Jain and R. Dubes, *Algorithms that Cluster Data*, Prentice Hall, Englewood Cliffs, 1988.
- [15] R. Lippman, "An introduction to neural computing," *IEEE ASSP Magazine*, April, pp. 4-22, 1987.
- [16] T. Kohonen, *Self-Organization and Associative Memory*, 3rd edition, Springer-Verlag, Berlin, 1989.
- [17] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York, 1981.
- [18] J. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters," *J. Cybernetics*, vol. 3, pp. 32-57, 1974.
- [19] T. Huntsberger and P. Ajjimarangsee, "Parallel self-organizing feature maps for unsupervised pattern recognition," *Int'l. Jo. General Systems*, vol. 16, pp. 357-372, 1989.
- [20] J. Bezdek, E. C-K. Tsao and N. Pal, "Fuzzy Kohonen Clustering Networks," *Proceeding of the IEEE International Conference on Fuzzy Systems*, pp.1035 -1043, San Diego 1992.
- [21] T. Kohonen, Self-organizing maps : optimization approach, Artificial neural networks, elsevier Sc. Pub., (Eds. T. Kohonen, K. Makisara, O. Simula and J. Kangas), 981-990, 1991.

3/3-67  
1618L3

N93-29529

# A Neuro-Fuzzy Architecture for Real-Time Applications

P.A. Ramamoorthy and Song Huang

Department of Electrical & Computer Engineering,  
University of Cincinnati, M.L. #30  
Cincinnati, Ohio 45221-0030  
Tel: 513-556-4757, FAX: 513-556-7326  
Email: pramamoo@nest.ece.uc.edu

## Abstract

Neural networks and fuzzy expert systems perform the same task of functional mapping using entirely different approaches. Each approach has certain unique features. The ability to learn specific input-output mappings from large input/output data possibly corrupted by noise and the ability to adapt or continue learning are some important features of neural networks. Fuzzy expert systems are known for their ability to deal with fuzzy information and incomplete/imprecise data in a structured, logical way. Since both of these techniques implement the same task (that of functional mapping and we regard "inferencing" as one specific category under this class), a fusion of the two concepts that retains their unique features while overcoming their individual drawbacks will have excellent applications in the real world. In this paper, we arrive at a new architecture by fusing the two concepts. The architecture has the trainability/adaptability (based on input/output observations) property of the neural networks and the architectural features that are unique to fuzzy expert systems. It also does not require specific information such as fuzzy rules, defuzzification procedure used etc., though any such information can be integrated into the architecture. We show that this architecture can provide a performance better than is possible from a single two or three layer feedforward neural network. Further, we show that this new architecture can be used as an efficient vehicle for hardware implementation of complex fuzzy expert systems for real-time applications. A numerical example is provided to show the potential of this approach.

## 1 Introduction

In general, fuzzy logic uses linguistic variables which are not crisply defined and logical relations between these variables to define the relationship between system inputs and outputs. On the other hand, neural networks use simple linear and nonlinear building blocks, interconnections among these blocks and training or learning procedures to obtain the system input-to-output mapping from large input/output samples. Thus, even though the research on neural networks and fuzzy logic have progressed for all practical purposes on two independent paths, it can be seen that both the architectures serve as models for arbitrary nonlinear mapping ( $f: x \rightarrow y$ , where  $x$  represents the input vector,  $y$  the output vector and  $f$ , the nonlinear transformation). Hence, it is important to understand the similarities and differences between these two approaches and the strengths/drawbacks of each. More importantly, it would be desirable to arrive at a hybrid approach/architecture that inherits the unique strengths of each without their shortcomings. In this paper, we show how such an architecture can be arrived and what are its important features. First, we provide a background to the problem in section 2 and present the new architecture in section 3. In section 4, we present results of simulation using the new architecture.

## 2 Background

### 2.1 Multi-layer Feedforward Networks

A neural network can be considered as a system that maps an input  $u$ , a vector of size  $N$ , into an output  $y$ , a vector of size  $M$ , by the function  $f : u \rightarrow y$  [1]. The mapping is performed in the network or system by weighting each and every input, summing the results, subtracting a bias value and passing the result through a nonlinear function which may produce a binary or bipolar or continuous value (between -1 to 1) for each output. Thus, it can be noticed that a neural network is nothing but a non-linear network. The mapping function  $f$  is assumed to be unknown and is estimated from several numerical I/O samples  $(u_i, y_i)$  through the training procedure.

Above, we described a one-layer feed-forward model of a neural network. It is widely assumed that the Kolmogorov's theorem on functional approximation is a proof that a two-layer neural network is sufficient for approximating arbitrary non-linear systems given sufficient number of hidden nodes [2]. But, it is only an existence proof and does not tell us how to arrive at the network. In fact, there surfaced questions as to whether this theorem itself is applicable to the problem at hand [3], but we are not concerned about that issue here. From our perspective, a neural network is a non-linear system with interconnected neurons, which maps inputs into the outputs via the non-linear function  $f$ , and the function  $f$  is not given or known but estimated from a set of numerical I/O samples.

### 2.2 Cerebellar Model Articulation Controller Neural Networks

Another example of a feedforward neural network is that of the Cerebellar Model Articulation Controller (CMAC). An example of this network is shown in Figure 1 for a simple two input and one output system. This neural network was introduced by Albus [4, 5, 6] and seems to be getting renewed attention through the work of Miller, et al., [7, 8], Ersu, et al., [9] and Moody [10]. The nonlinear mapping is achieved in CMAC through nonlinear building blocks such as input sensors (a simple range detector — that is, each sensor produces an output of 1 if the input value falls in a certain range and 0 otherwise), AND gates (state space detectors) and OR gates (multiple field detectors) and linear weighting and summing blocks. It is claimed that CMAC can be an alternative for backpropagation networks<sup>1</sup> to achieve better performance [8]. Since backpropagation is basically a gradient descent technique applied to a multilayer nonlinear network, it needs a large computation time, converges slowly for large systems, and has an error surface which may contain local minima. The CMAC network contains a single linear feedforward network (that has to be trained) and hence does not require error propagation etc. and therefore can learn the mapping rather quickly. Miller, et al., recently modified the original CMAC architecture [11] where it is suggested that: 1). The input sensors implement local receptive fields with tapered sensitivity functions (that is the sensor output is 1.0 if the input is in the center of the receptive field, and the output decreases linearly towards 0.0 for inputs near the edges of the fields). 2). The state-space detectors can be considered as analog units (multiplication rather than logic AND gates) with the property that the unit output is 1.0 if all inputs are 1.0, while the unit output

---

<sup>1</sup>Backpropagation refers to an approach used to train multilayer networks and can be applied to any network. Hence it is not correct to call the multilayer perceptron network as a backpropagation network. We use it here as it has become a common practice.

decreases to 0.0 if any input decreases to 0.0. And 3). The multiple field detectors can be considered as simple summing units (rather than logic OR gates). The network output is then the sum of products of a certain number of non-zero multiple field detector outputs and the corresponding weights. It is indicated that the modified CMAC architecture has better properties than the original CMAC because the modified version provides continuous instead of piece-wise function approximations.

### 2.3 Fuzzy Logic and Fuzzy Expert Systems

The fuzzy systems can be also considered as implementing a mapping function  $f: u \rightarrow y$  [12 - 16]. The mapping is effected via:

1. Splitting the input(s) and output(s) total-range of values into a number of subsets or ranges which can possibly overlap.
2. Assigning membership functions corresponding to these sets for all range of values of the variables. Together these sets can be considered as fuzzy sets where the membership functions denote the degree of belongings of a particular input or output value to the various fuzzy sets of those inputs or outputs.
3. Defining Boolean relationships among the input fuzzy sets and output fuzzy sets. These Boolean expressions identify the output fuzzy sets under which the expected outputs might fall when the inputs fall under certain input fuzzy sets.
4. Procedure (also known as defuzzification) to find the final or crisp output(s) from the output fuzzy sets (that are selected or activated) and the various membership functions.

The steps involved in implementing a fuzzy expert system is shown in Figure 2. From the figure, it can be noted that the functional mapping is achieved in a fuzzy expert system through three well defined sub-blocks. We will look into this architecture in the next section.

## 3 New Neuro-Fuzzy Architecture

Let us examine more closely the steps that would be involved in hardware implementation of a fuzzy expert system. Let  $M$ ,  $N$  be the number of inputs to and outputs of the system,  $m_i$  ( $i = 1$  to  $M$ ), the number of fuzzy sets for the input  $i$  and  $n_j$  ( $j = 1$  to  $N$ ), the number of fuzzy sets for the output  $j$ . We will assume that the inputs and outputs are represented in a fixed-point weighted binary representation with  $B$  bits for each variable. The inputs can then be converted into an unweighted binary representation (with  $Q = 2^{**}B$  binary lines for each input and only one bit "on or 1" for any input at any given time) using  $M$  number of  $B$  to  $2^{**}B$  line decoders as shown in Figure 3. Now, given the exact input values, the first step in the implementation of a fuzzy expert system is to identify the fuzzy sets under which these input values fall. In a hardware implementation, this can be achieved by assigning one bit for each fuzzy set such that a particular bit gets turned on if and only if the input values fall under the range of that particular fuzzy set. We can call these bits as input fuzzy set pointers (IFSP) and there will be  $m = \sum m_i$  IFSPs. The logic for conversion from the input values to (unweighted binary representation) IFSPs is then simply a set of  $m_i$  OR gates for the  $i$ th input as shown in the figure.

Having identified the input fuzzy sets to which the given values of the inputs belong, the next task is the identification of the corresponding output fuzzy sets and this is accomplished through the fuzzy rules or fuzzy associative memories (FAMs). This step can be implemented in hardware by assigning bits to identify the various output fuzzy sets as we did for the case of input fuzzy sets. Thus, we will have  $n = \sum n_i$  output fuzzy set pointers (OFSPs) and the values of these binary variables will depend upon the values of the IFSPs. Since any binary variable in general can be represented in a two-level (or three level if we consider logical "negative" as one level <sup>2</sup>) sum-of-product expression involving the input binary variables, the fuzzy inferencing can be implemented in a two level logic as shown in the figure.

The defuzzification process makes use of the input values, corresponding input membership function values and the OFSPs, and the membership functions of the selected output fuzzy sets to produce the final outputs (see the third block of Figure 3). This block is more complex. However, it can be sub-divided into a number of sub-blocks or smaller sub-networks as shown in Figure 3B <sup>3</sup>. Here, we have  $n$  sub-networks with one output fuzzy set pointer acting as enable/disable signal for each network. They generate intermediate outputs which are combined (simple addition, for example) to produce the final outputs as shown in Figure 3B.

From the above discussions, we find that a hardware representation of a fuzzy expert system involves three separate blocks where each block has a unique function. The first two blocks and the sub-networks of the third block can in turn be 2 or 3 layer feedforward networks <sup>4</sup>. Thus, a fuzzy expert system can be considered as consisting of a number of multilayer feedforward networks with structured interconnections between them. Therefore, it is quite conceivable that fuzzy expert systems can provide a superior performance for functional mapping as compared to a single 2 or 3 layer network <sup>5</sup>. Kong and Kosko [17] illustrated this point through an example. Similar arguments can be made while comparing CMAC neural networks with fuzzy expert systems <sup>6</sup>.

The superior performance of fuzzy expert systems can be attributed to the use of additional information as compared to that for multilayer neural networks. In the case of neural networks, we use the input/output samples, a fixed architecture (or a time evolving architecture as in [19]) and a training procedure. In the case of fuzzy expert systems, additional information such as fuzzy sets, fuzzy rulebase etc. are used to obtain the mapping. Some of the information such as the number and the ranges of fuzzy sets, membership functions can be obtained from the problem at hand <sup>7</sup>. Thus, we can use such information, the derived architecture (shown in Figure 2) and training procedures to implement any functional mapping. This trainable architecture can be called a "Neuro-Fuzzy Architecture". The advantages of such an architecture are:

A structure consisting of smaller networks that can be trained easily and faster;

---

<sup>2</sup>The fuzzy rules do not use negation (input not falling under the range of a fuzzy set) and perhaps is a limitation of the classical fuzzy expert systems. This has to be researched further.

<sup>3</sup>There are many different possibilities and we discuss only one.

<sup>4</sup>This is due to the fact that any mapping can be achieved by multilayer feedforward networks.

<sup>5</sup>Cybenko has showed mathematically that a 3-layer network is sufficient for any functional mapping. But the paper dose not address the limits on the error of approximation.

<sup>6</sup>In another paper, we show that a CMAC network can be considered as a special case of a fuzzy expert system [18].

<sup>7</sup>The initial choice may not be optimal. However, it can be argued that the incorporation of some known information is better than incorporating none.

Incorporation of information about the problem <sup>8</sup>;

Adaptibility;

Identification of the fuzzy rules: Since there is a one-to-one correspondence between the blocks and the tasks in a fuzzy expert system implementation, we can train the second block if the fuzzy set regions are given and use that block to identify the fuzzy rulebase;

As an vehicle for hardware implementation: That is, rather than using a language based processing (with its associated MFLOPS or mega fuzzy logic operations per second ratings), we can use the new architecture as a digital or hard-wired implementation of a fuzzy expert system. Such an implementation will have a tremendous edge in real-time applications since the number of rules increases exponentially with a linear increase in the number of inputs. For example, if we assume that there are 5 fuzzy sets per input, the number of rules for a five input system and a ten input system will be 3125 and 9,765,625 respectively. Perhaps due to this problem, fuzzy expert systems considered in the literature are mostly two input systems or separability is assumed when there are more than two inputs;

Modeling of complex systems. By modeling the block corresponding to the rulebase by a 3 layer feedforward neural network and the defuzzification block by a number of neural networks, we will be able to model complex systems than is possible based on the presently used approaches.

## 4 Example

Here, we consider the problem of designing a controller to successfully back up a truck to a loading dock from any reasonable initial location. This problem was solved earlier by Nguyen and Widrow [20] using a two-layer neural network architecture with 26 nodes and later by Kong and Kosko [17] using a fuzzy expert system. The details of the problem are shown in Figure 4. For this example, we assume that the fuzzy sets (of inputs and outputs), the corresponding membership functions and the fuzzy rules are known (shown in Figures 5 and 6) but the defuzzification procedure is unknown. Thus, there is no need to train the first two blocks of Figure 3A, but only the third block (and the corresponding sub-networks) needs to be trained. The desired outputs corresponding to a set of inputs are obtained using the centroid defuzzification method (see [17] for details) and are used in the training. Two different approaches are used in the training: 1) Input  $x$ ,  $\phi$  values and the IFSPs as the inputs to the networks and  $\theta$  as the desired outputs and 2) Inputs  $x$ ,  $\phi$ , and the corresponding membership function values as the inputs and  $\theta$  as the desired outputs. It should be noted here that both approaches do not use the output membership function values. Since more accurate results are needed when the truck is in the center area or near center area, we selected more samples for  $x$ -position around 50, and less samples to the extremes. The training samples of  $\phi$  are chosen in the same fashion. This led to 34  $x$ -positions and 72  $\phi$  angles. Thus 2448 samples are used to train the controller. The  $y$ -positions are not used in training, thus simplifying the training process. There are 7 sub-networks corresponding to the seven output fuzzy sets. The whole set of training samples are divided into 7 smaller

---

<sup>8</sup>Concepts such as representing/designing a larger system by a number of smaller subsystems are not new in engineering.

groups according to their belongings to the output fuzzy sets. The largest group contained 826 training samples and the smallest one has 271 samples. Some samples are used in more than one sub-network due to the overlapping of the output fuzzy sets. This brought the total training samples for all the sub-networks to 3624.

The training samples are normalized to the range of -0.5 to 0.5. We selected 10 second-layer nodes for every sub-network. The backpropagation algorithm is used for the training. The number of iterations for training varied from few hundreds (for smaller sample groups) to few thousands (for larger groups). The training converged in both the cases with the average squared errors from 0.0005 (for the samples chosen near the center) to 0.0015 (for the extreme sets). A truck trajectory produced using the trained neural network corresponding to case 1) is shown in Figure 7A, and Figure 7B shows one trajectory corresponding to the case 2). It can be noted both methods produce smooth trajectories as compared to the one generated by a two layer neural network (as shown in Kong and Kosko's paper [17] and shown in Figure 8A) for this particular initial condition. Further, the trajectories by these networks are very similar to the ones produced by the original fuzzy expert system (the teacher) as can be seen comparing Figures 7A and 7B with Figure 8B.

## 5 Conclusions

Using functional mapping as a common framework, we showed how neural networks and fuzzy expert systems can be merged to arrive at a new Neuro-Fuzzy architecture. The architecture has the trainability/adaptability (based on input/output observations) property of the neural networks and the architectural features that are unique to fuzzy expert systems. It also does not require specific information such as fuzzy rules, defuzzification procedure used etc., though any such information can easily be integrated into the architecture. We showed that this architecture can provide a better performance than is possible from a single two or three layer feedforward neural network, and can be used as an efficient vehicle for hardware implementation of complex fuzzy expert systems for real-time applications. A numerical example is also provided to show the potential of this approach. Many variations of the architecture seem to be possible and further work needs to be done to exploit the potentials offered by the new architecture.

## 6 References

1. R. P. Lippmann, "An introduction to computing with neural nets", *IEEE ASSP Magazine*, pp. 4-22, April 1987.
2. G. Cybenko, "Approximation by superpositions of a sigmoidal function", Technical Report, University of Illinois, 1988.
3. F. Girosi and T. Poggio, "Representation Properties of Networks: Kolmogorov's Theorem," *Neural Computation*, Vol. 1, pp. 465-469, 1989.
4. J.S. Albus, "A theory of cerebellar functions", *Mathematical Biosciences*, Vol. 10, pp. 25-61, 1971.
5. J.S. Albus, "Theoretical and Experimental Aspects of a Cerebellar Model", PhD thesis, Univ. of Maryland, 1972.
6. J.S. Albus, "Data storage in the cerebellar model articulation controller", *J. of Dynamic Systems, Measurement and Control*, pp. 228-233, Sept. 1975.

7. W.T. Miller, "Non-linear learning controller for robotic manipulators", *Proc. SPIE, Intelligent Robots and Computer Vision*, Vol. 726, pp. 416-423, Oct. 1986.
8. W.T. Miller, F.H. Glanz, and L.G. Kraft, "CMAC: An associative neural network alternative to backpropagation", *Proc. IEEE*, pp. 1561-1567, Oct. 1990.
9. E.Ersu and H.Tolle, "Hierarchical learning control— an approach with neuron-like associative memories", *Proc. IEEE Conf. on Neural Infor. Proc. Systems*, Nov. 1988.
10. J.Moody, "Fast learning in multi-resolution hierarchies", in *Advances in Neural Information Processing*, edited by D.Touretzky, Morgan Kaufmann, 1989.
11. W.T. Miller, E.An, and F.H. Glanz, "The design of cmac neural networks for control", *Proc. 6th Yale Workshop on Adaptive and Learning Systems*, Aug. 1990.
12. L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Trans. Systems, Man and Cybernetics*, Vol. SMC-3, pp. 28-44, 1973.
13. L.A. Zadeh, "Commonsense Knowledge Representation Based on Fuzzy Logic," *IEEE Computer*, Vol. 10, pp. 61-65, 1983.
14. L.A. Zadeh, *Selected Paper by L.A. Zadeh*, edited by R.R. Yager et al., John Wiley and Sons, 1987.
15. L.A. Zadeh, "Fuzzy logic", *IEEE Computer*, pp. 83-93, April 1988.
16. B. Kosko, "Fuzzy Entropy and Conditioning," *Information Sciences*, Vol. 40, pp. 165-174, 1986.
17. S.Kong and B.Kosko, "Comparison of fuzzy and neural truck backer-upper control systems", *Proc. of IJCNN 1990*, Vol. III, pp. 349-358, June 1990.
18. P.A. Ramamoorthy and Song Huang, "Cerebellar Model Articulation Controller Neural Network — A Simple Fuzzy Expert System in Disguise?", to appear in the Conf. Proc. of Artificial Neural Networks in Engineering, Nov. 1991.
19. G. Govind and P.A. Ramamoorthy, "A New Neural Network Structure and Algorithm for Nonlinear System Identification", to be submitted to *IEEE Transactions on Systems, Man, and Cybernetics*, 1991.
20. D. Nguyen and B. Widrow, "The Truck Backer-Upper: An Example of Self-Learning in Neural Networks," *Proceedings of IJCNN 1989*, Vol. 2, pp. 357-363, June 1989.

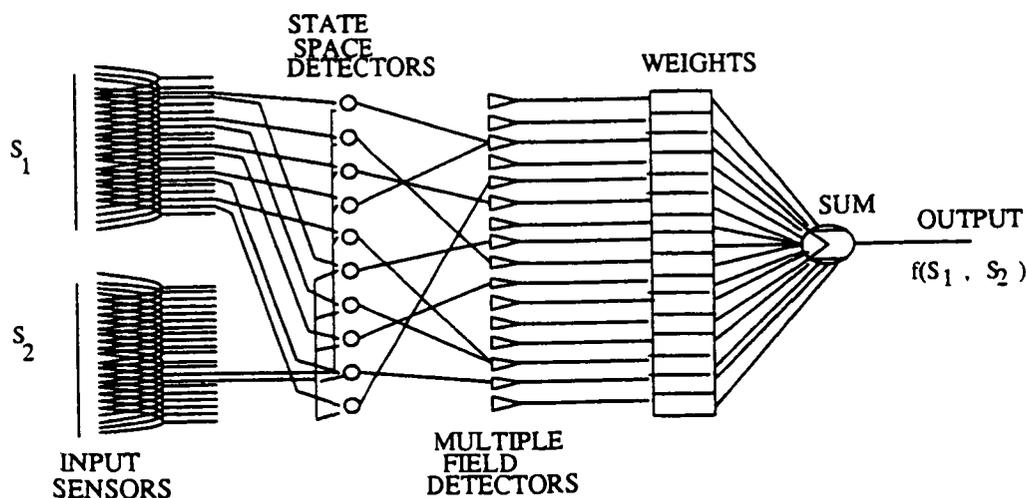


Figure 1. A simple CMAC system with two inputs and one output.

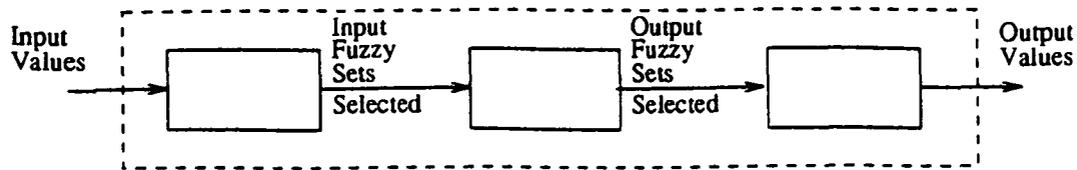
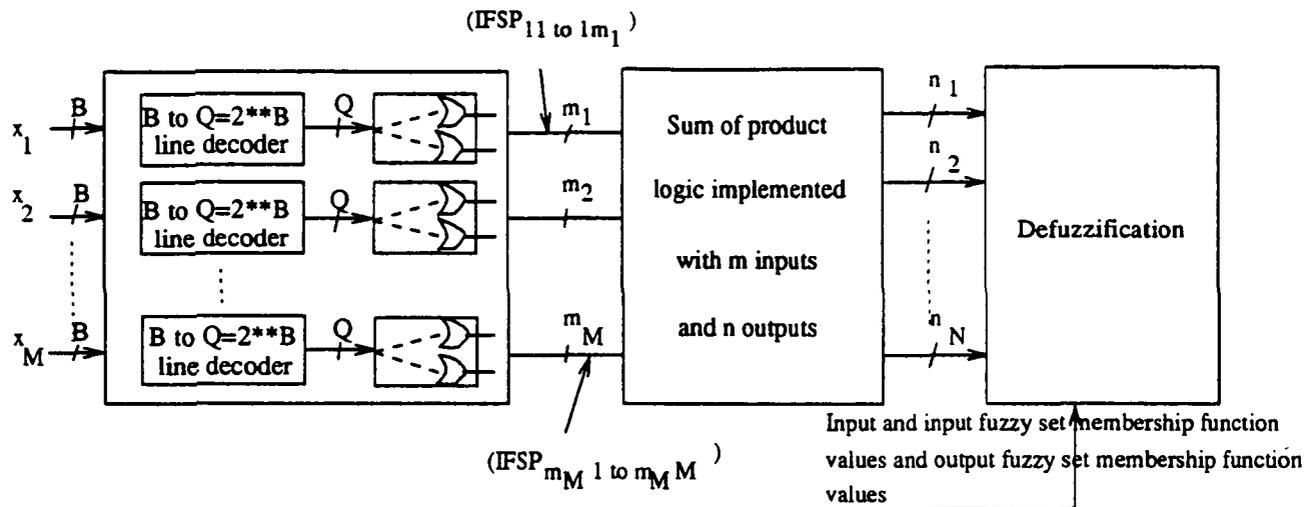
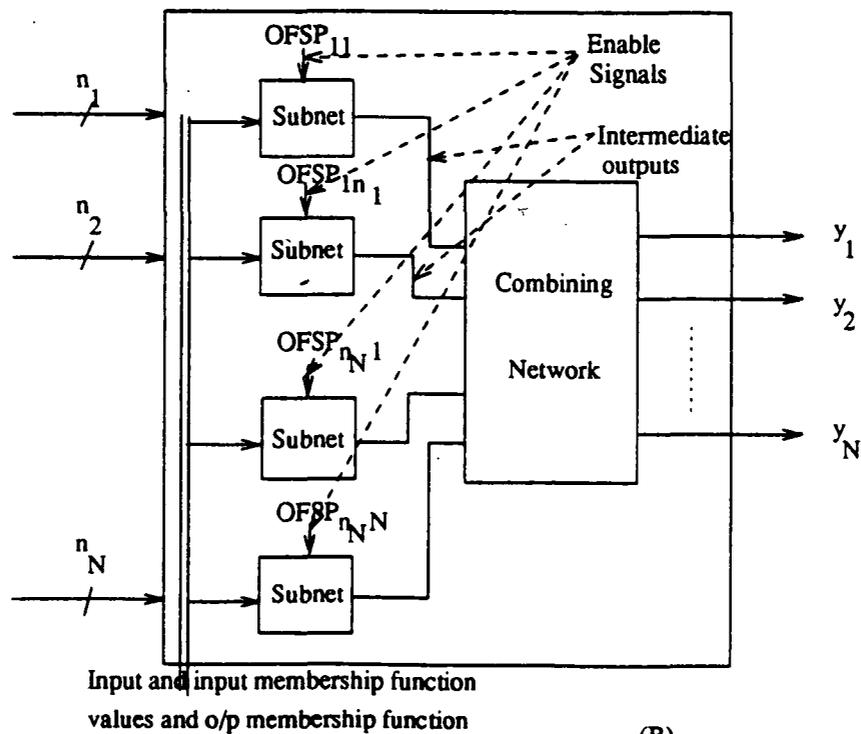


Figure 2. Block diagram of a fuzzy expert system.



(A)



(B)

Figure 3. (A) Block diagram of the Neuro-Fuzzy system

(B) Detailed representation of the third block in Figure (A).

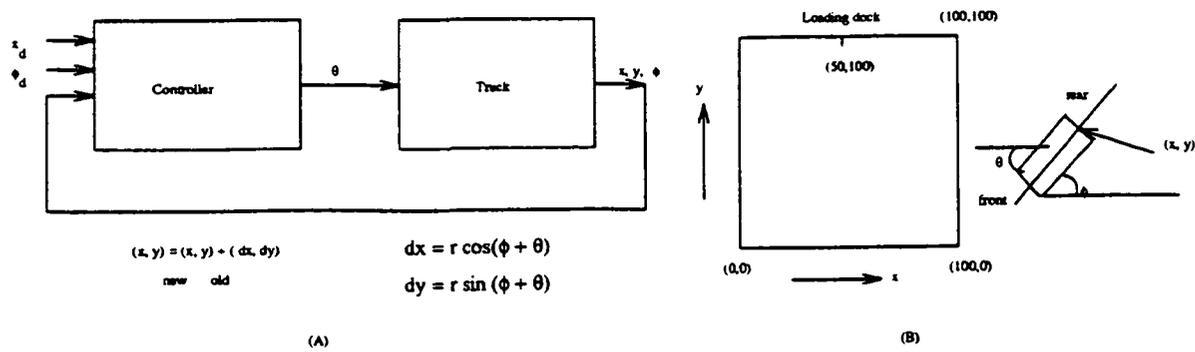


Figure 4. (A) Block diagram of a fuzzy expert system based controller to back up a truck. (B) Details of the loading zone and the truck.

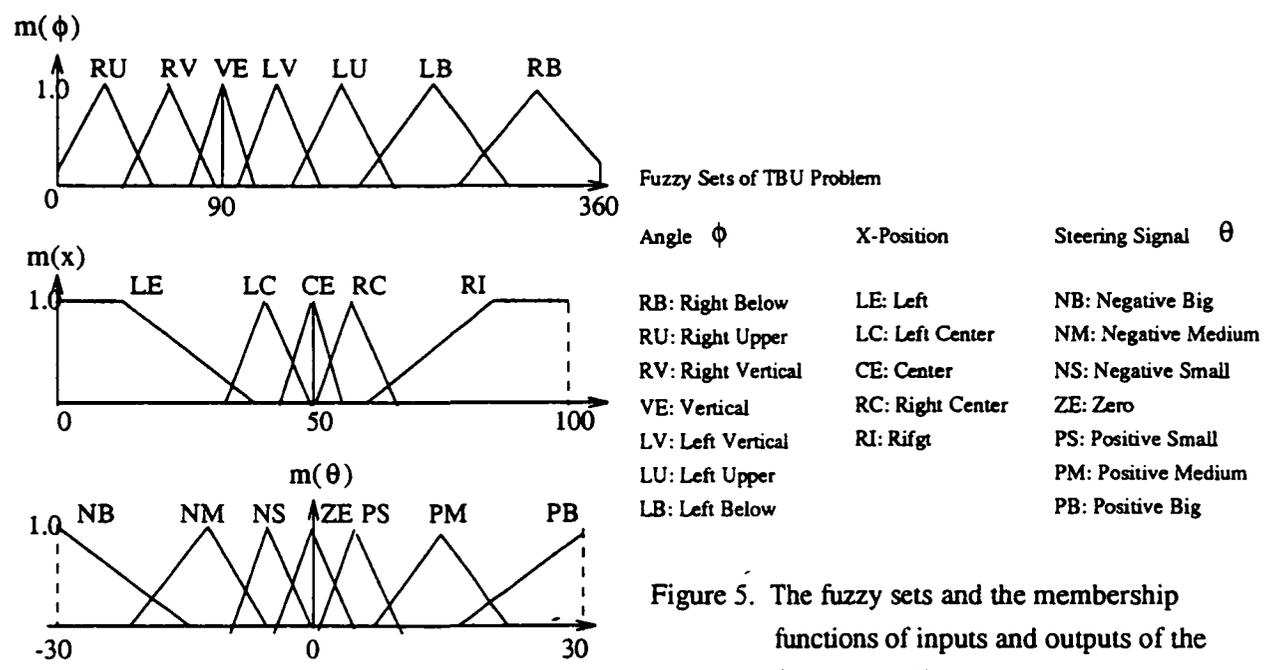


Figure 5. The fuzzy sets and the membership functions of inputs and outputs of the fuzzy controller.

x - position

	PB	PB	PM	NB
	ZE	PS	PM	PB
angle	NB	NM	PS	PM
( $\phi$ )	NB	NM	ZE	PM
	NB	NM	NS	PM
	NB	NB	NM	NS
	PB	PB	NM	NB

Figure 6. The rule base for the fuzzy controller.

Truck Backer-Upper Controller (FES-NN)  
 $x = 30, y = 20, \phi = 30$   
 Input Fuzzy Set Pointers used for training

Truck Backer-Upper Controller (FES-NN)  
 $x = 30, y = 20, \phi = 30$   
 Membership function values used for training.

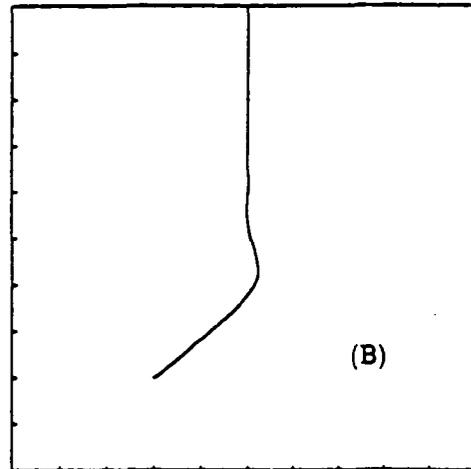
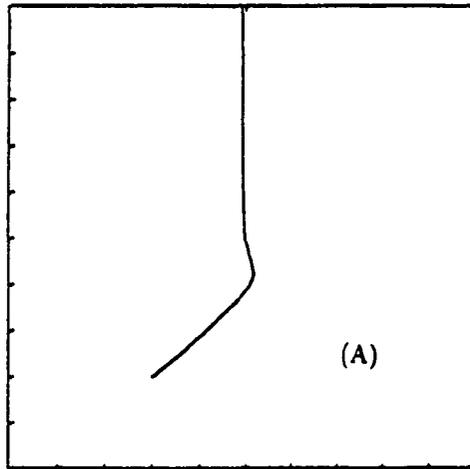
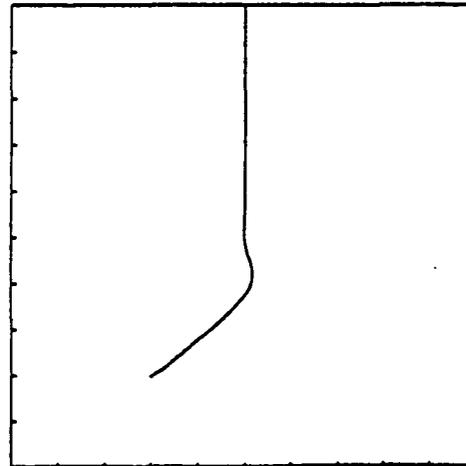
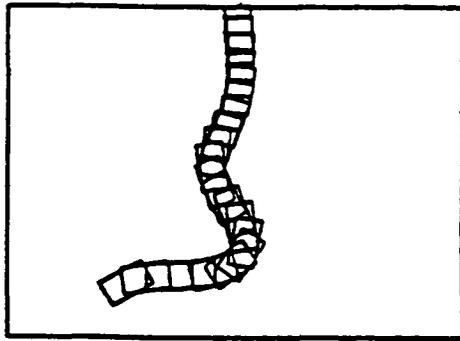


Figure 7 The trajectories of the truck using the Neuro-Fuzzy controller,  
 (A) Using Input Fuzzy Set Pointers in the training, and  
 (B) Using Membership function values.

Truck Backer-Upper Controller (FES)  
 $x = 30, y = 20, \phi = 30$



(A)  
 (B)

Figure 8 (A) A trajectory of the truck using a neural network controller.  
 (B) A trajectory of the truck using a fuzzy controller.

514-67  
161 814  
p. 9

N 93 - 29530

## A COMPOSITE SELF TUNING STRATEGY FOR FUZZY CONTROL OF DYNAMIC SYSTEMS

C-Y Shieh, Graduate Student  
Satish S. Nair, Assistant Professor  
Department of Mechanical and Aerospace Engineering  
University of Missouri - Columbia  
Columbia, MO 65211

**Abstract** - The feature of self learning makes fuzzy logic controllers [1,2] attractive in control applications. This paper proposes a strategy to tune the fuzzy logic controller on-line by tuning the data base as well as the rule base. The structure of the controller is outlined and preliminary results are presented using simulation studies.

### 1.0 Introduction

Fuzzy logic control is usually implemented using lookup tables that are derived off-line. Most of the commercial products currently employ this approach. Several researchers though have been studying approaches to incorporating learning into the fuzzy control architecture. Most of these algorithms are, however, heuristic and subjective and there is no systematic procedure to design and analyze self-tuning fuzzy controllers. Along these lines, a self-tuning strategy was presented by Wu et al. [3,4] to tune the data base for a nonlinear time varying system. They also report successful on-line implementation on an experimental setup. This paper extends the study using a controller with more degrees of freedom.

#### System Description

Figure 1 show a four-bar linkage system considered, which is representative of a common type of transmission system in several machines. The governing equations for the load is given by Eq.1,

$$M(\theta)\ddot{\theta} + V(\theta)\dot{\theta}^2 + G(\theta) = T(t) \quad (1)$$

where  $\theta$  is angular position of link 2,  $M$  and  $V$  are complex nonlinear functions of  $\theta$  representing the reflected inertia and the centrifugal and coriolis force terms respectively, and  $T$  is the torque applied by the motor. The system becomes a time invariant one when  $M(\theta)$ ,  $V(\theta)$  and  $G(\theta)$  are constants. The model nonlinearities in this case are primarily motor friction, both viscous and coulomb. Figure 2 shows that the variation of  $M$  as a function of the angular position of link 2 is significant. The control objective is to maintain the speed of link 2 constant.

## 2.0 Composite Algorithm

A 'velocity' type fuzzy logic controller (Figure 3) is used in this study. The error (e) and change of error ( $\Delta e$ ) are used as the control variables of the system and are defined as

$$\begin{aligned} e(k) &= s(k) - y(k) & \text{where } k &: \text{ present time} \\ \Delta e(k) &= e(k) - e(k-1) & k-1 &: \text{ previous sampling instant} \\ & & s(k) &: \text{ setpoint at instant } k \\ & & y(k) &: \text{ output of plant at instant } k \end{aligned}$$

Triangular and trapezoidal membership functions are used to interpret term sets of linguistic variables. Based on this interpretation, the term sets in the data base can be represented by functions of the position of the fuzzy sets heights as

$$E = E(0, s_e, m_e, b_e), \quad \Delta E = \Delta E(0, s_{\Delta e}, m_{\Delta e}, b_{\Delta e}), \quad \Delta U = \Delta U(0, s_{\Delta u}, m_{\Delta u}, b_{\Delta u})$$

The maximum overlap of membership functions of two adjacent fuzzy sets is 0.5 and three fuzzy sets do not overlap. This is found to be the optimal arrangement for 'completeness'[7].

The controller proposed consists of two parts,  $FLC_D$ , based on data base tuning, and  $FLC_R$ , based on rule base tuning. Contributions from both the FLCs are added to get the actuating signal (Figure 3). In the reported study, the data base is tuned first, and then the rule base.

### Data Base Tuning

A tuning factor  $\alpha$  is introduced to modify the support of every fuzzy set of the term set simultaneously, keeping the same completeness, as

$$F' = F [0, \alpha s, \alpha m, \alpha b]$$

where F can be any fuzzy term set of E,  $\Delta E$  and  $\Delta U$ , and F' is the modified fuzzy term set (Figure 4). Note that the rule base does not change in this case. This algorithm can be briefly stated as follows :

1. set all factors  $\alpha_i = 1$
2. select linguistic variable  $F_i$  to be tuned
3. start the control program and obtain  $ISE_0$
4. modify  $\alpha$  to  $\alpha-0.1$  and get the new membership functions
5. start control program to get  $ISE_i$
6. if  $0 \leq \alpha$  goto 4
7. get the minimum  $ISE_i$  and select the corresponding value of  $\alpha_i$  as optimal
8. go to 2 for the next linguistic variable  $F_i$  until all are complete
9. repeat (2) through (8) until  $\alpha_i(\text{new}) = \alpha_i(\text{old})$ .

### Tuning of the Rule Base

This part of the algorithm is implemented on-line after time  $> 4*t_r$ , where  $t_r$  is the system rise time.

The algorithm is structured as follows :

if  $\Delta ISE(t+\Delta t) > \Delta ISE(t)$  then

if  $\omega_2 > \omega_d$ , reduce predominant consequent term set by one level

if  $\omega_2 < \omega_d$ , increase the predominant consequent term set by one level

If  $\Delta ISE(t+\Delta t) \leq \Delta ISE(t)$ , no changes are made. ISE is the integral squared error calculated from  $4*t_T$  to  $t$ . The term 'predominant' refers to the antecedent sets with  $\mu(x) > 0.5$ . In our case, they are  $\mu_e(x) > 0.5$  and  $\mu_\theta(x) > 0.5$ . The contribution from  $FLC_T$  has arbitrarily been scaled at present to provide small correction inputs based on  $\theta$ , the position of link 2.

### 3.0 Results and Discussion

The original rule base and the data base of this fuzzy system is based on designer's knowledge which is heuristic and subjective. The sampling rate for the simulation control program is 150 Hz. The simulation is implemented using the Advanced Continuous Simulation Language (ACSL) and run on a CRAY computer. The membership functions are used directly rather than by lookup tables. Figures 5 and 6 depict the variation of output speed for the two control configurations. For the  $FLC_d$  only case, the error is  $ISE=0.185$ , while for the composite controller, i.e.,  $FLC_T$  and  $FLC_d$  the error was found to decrease to 0.156. The data base tuning was accomplished in three passes through the loop, i.e. steps 2 to 8 in the data base tuning algorithm above. The rule base tuning was performed only for 10 seconds (approximately 16 rotations of the four bar linkage). At present we only report that the architecture gives good results and has promising qualities. The controller should learn to reduce the error better after longer training periods.

A controller architecture is proposed which is capable of learning the periodic time varying dynamics of a nonlinear system and compensating for the repetitive dynamics. This compensation is provided by an additional input from the  $FLC_T$  part of the controller. In the system considered the periodic variation in load inertia results in a continuous fluctuation of load speed. Data base tuning by itself does not suffice since it does not capture the spatial variation effects. Appropriate rule base modification based primarily on the input angle  $\theta$  is found to be effective. It should be noted that the fuzzy logic controller allows for the inclusion of this information in a simple way as compared to the classical ones. MRAC controllers have also been designed for the system, but the complexity in its design is much more as compared to the fuzzy case [5]. The results presented are of a preliminary nature but seem to show definite trends as far as convergence and suitability of the proposed architecture. Real time implementation and experimental studies will be reported in forthcoming publications.

#### 4.0 Acknowledgments

This work was partially supported by the National Center for Supercomputing Applications under grant number CEE920013N and utilized the computer system CRAY Y-MP4/464 at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.

#### 5.0 References

- [1] T. J. Procyk and E. H. Mamdani, 1979, "A Linguistic Self-Organizing Process Controller," *Automatica*, vol. 15, pp. 15-30.
- [2] C. Batur and V. Kasparian, 1989, "A Real Time Fuzzy Self-Tuning Control," *American Control Conference*, pp. 1810-1815.
- [3] Wu, K, Outangoun, S., and Nair, S. S., March 1992, "Modeling and Experiments in Fuzzy Control," *First IEEE International Conference on Fuzzy Systems*, San Diego, CA , pp.725-731.
- [4] Wu, K., and Nair, S. S., 1991, "Self Organizing Strategies for Fuzzy Control," *North American Fuzzy Information Processing Society Annual Meeting (NAFIPS-91)*, Columbia, Missouri, pp 296-300 [Outstanding Paper Award, Fuzzy Logic].
- [5] Outangoun, S., Wu, K, and Nair, S. S., 1991, "Experimental Comparison of Fuzzy Logic and Adaptive Control Techniques," *North American Fuzzy Information Processing Society Annual Meeting*, Columbia, Missouri, pp 388-392.
- [6] K. Wu, 1991, "Self Organizing Fuzzy Control - Analytical and Experimental Investigation," MS Thesis, University of Missouri - Columbia.
- [7] O. Ishizuka, *et al*, 1990, "Optimum Control Based on Fuzzy Inference and Its Simulation," *International Conference on Fuzzy Logic & Neural Networks*, pp. 61-64.

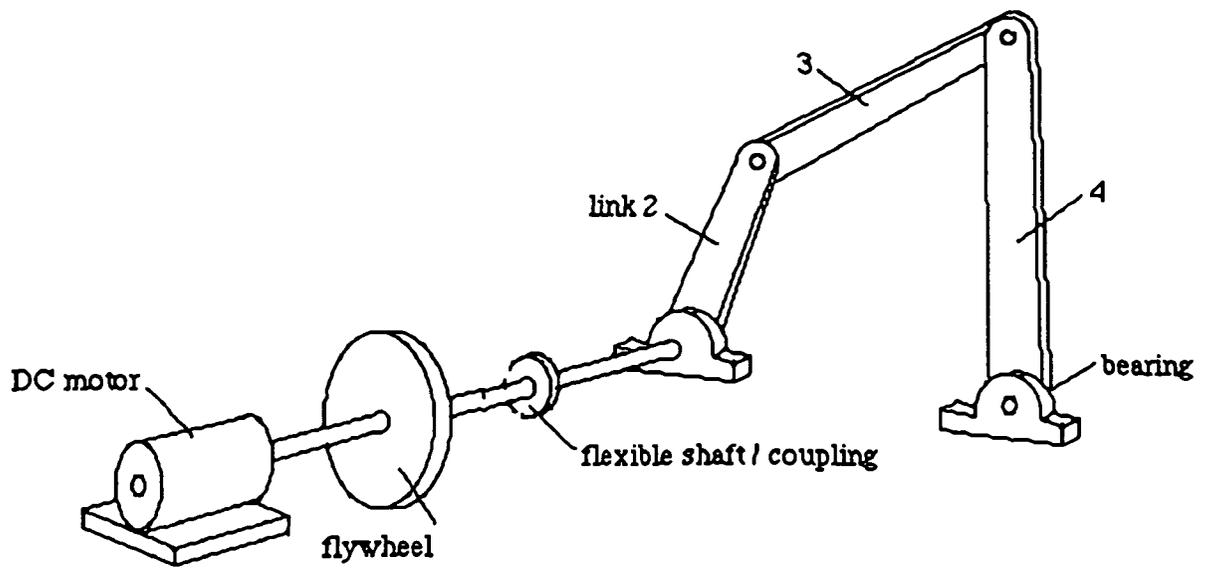


Figure 1 . Nonlinear periodically time varying system

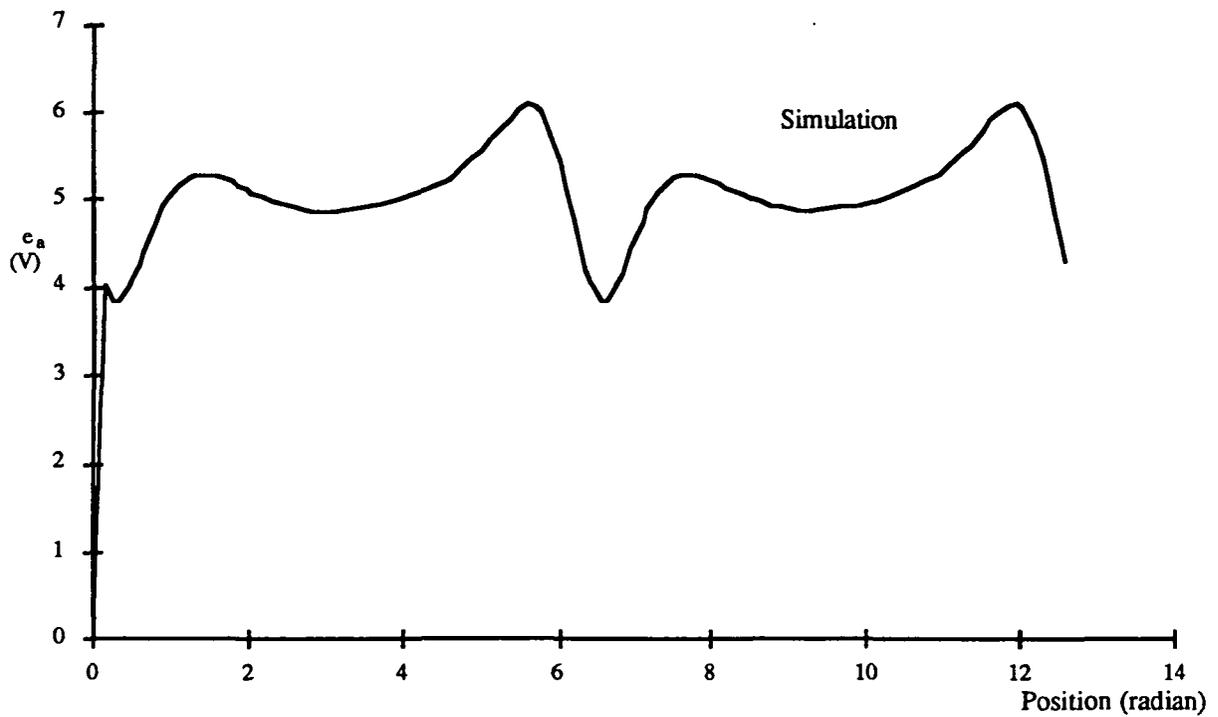


Figure 2. Load characteristics (a). Motor voltage required for constant speed

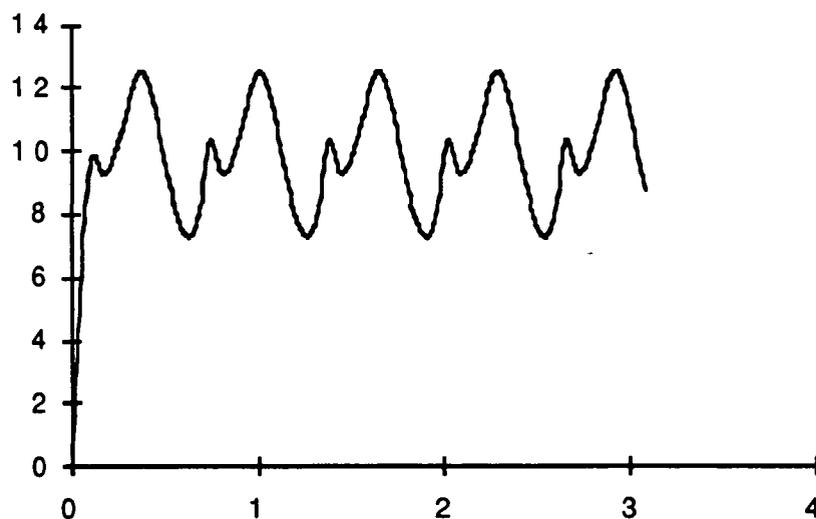


Figure 2. (b). Variation of speed at a constant motor voltage

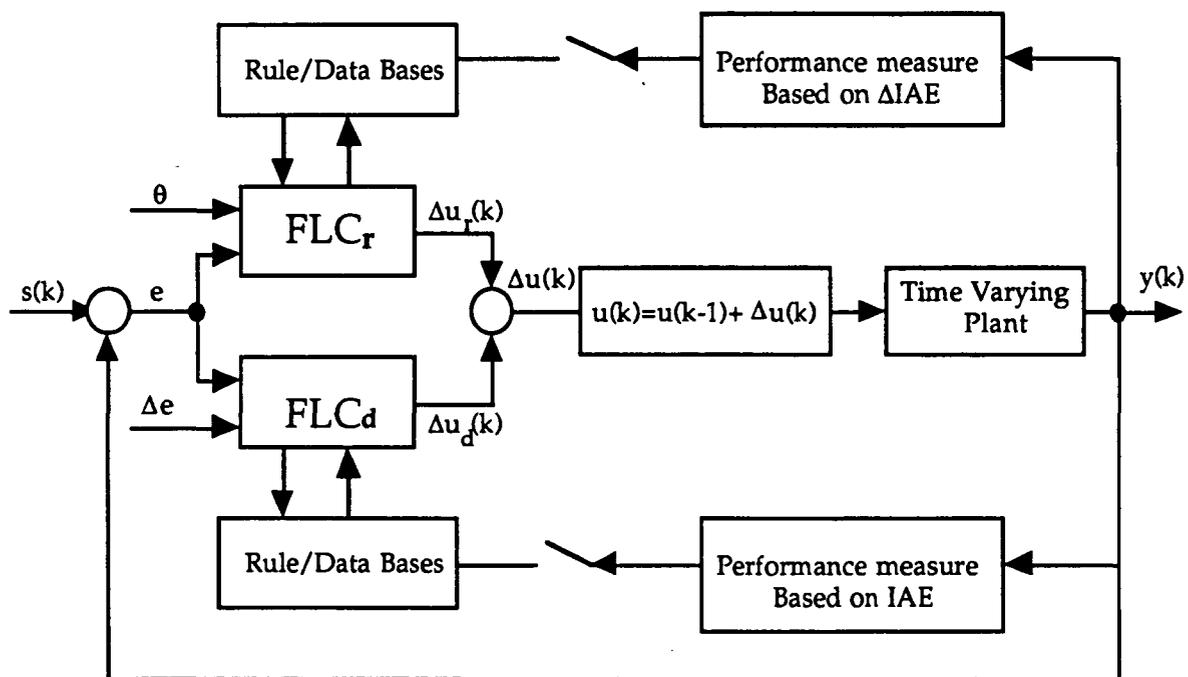


Figure 3. Composite fuzzy logic controller

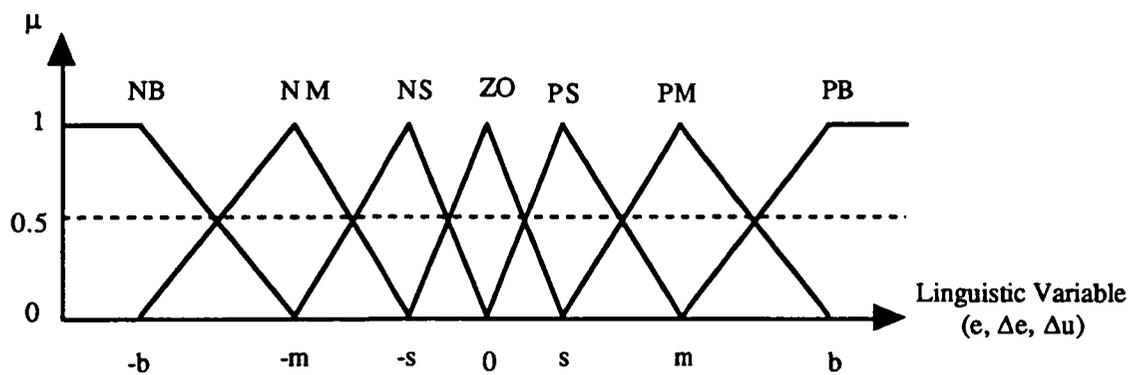


Figure 4. Membership functions for fuzzy term sets

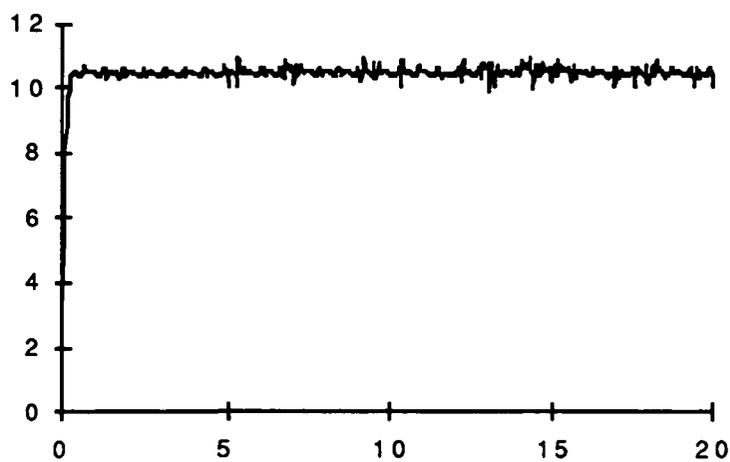


Figure 5. Fuzzy control with only data base tuning FLC

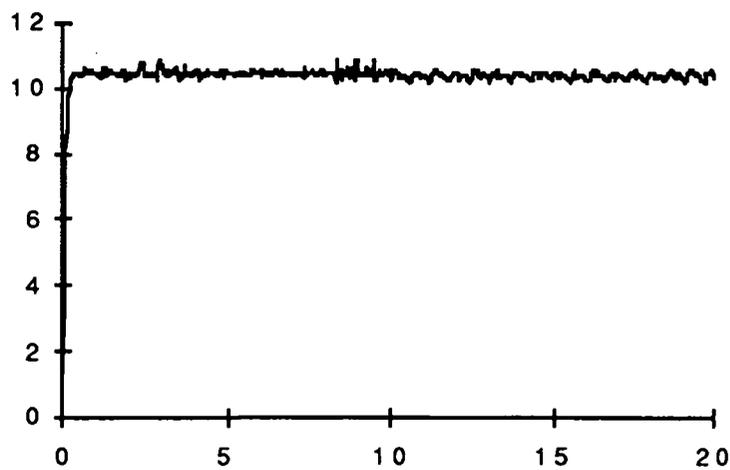


Figure 6. Fuzzy control with both data base and rule base tuning FLCs

A Self-Learning Rule Base for Command Following in Dynamical Systems<sup>1</sup>

P. 10

Wei K. Tsai, Hon-Mun Lee  
Department of Electrical and Computer Engineering  
University of California at Irvine  
Irvine, CA 92717  
and

Alexander Parlos  
Department of Nuclear Engineering  
Texas A&M University  
College Station, TX 77843

Abstract

In this paper, a self-learning Rule Base for command following in dynamical systems is presented. The learning is accomplished through reinforcement learning using an associative memory called SAM. The main advantage of SAM is that it is a function approximator with explicit storage of training samples. A learning algorithm patterned after the dynamic programming is proposed. Two unstable dynamical systems artificially created are used for testing and the Rule Base was used to generate a feedback control to improve command following ability of the otherwise uncontrolled systems. The numerical results are very encouraging. The controlled systems exhibit a more stable behavior and a better capability to follow reference commands. The rules resulting from the reinforcement learning are explicitly stored and they can be modified or augmented by human experts. Due to the overlapping storage scheme of SAM, the stored rules are similar to fuzzy rules.

---

<sup>1</sup>This research research is funded by the U.S. Department of Energy Idaho Operations Office under the Grant DE-FG07-89ER12893.

## I. INTRODUCTION

Expert systems (or knowledge-based systems) technology has many uses, see for example [1]. In this paper we will focus on automatic generation of the rule-base for controlling nonlinear dynamical systems often encountered in engineering endeavors. For nonlinear dynamical systems, there are two basic problems: estimation and control. Estimation refers to the the problem of reconstructing dynamic (as well as static) behaviors of partially unknown systems from input-output sample pairs. Control refers to the problem of generating desired system behaviors by exerting some control efforts to the system. In this paper, we will focus on one sub-problem of the control problem: command following. The problem is to generate feedback control rules to force the response of the controlled system to follow a reference command input. Our approach is to generate needed control by reinforcement learning [2] using an associative memory.

In this paper, we will focus only on the heart of an expert system, the Rule Base. A rule is of the form: "if  $x_1 = x_1^j, x_2 = x_2^j, \dots, x_d = x_d^j$ , then the control is  $u = u^j$ ." We propose a self-learning Rule Base in which learning is accomplished through reinforcement learning using an associative memory called SAM (Self-organizing Associative Memory). Self-learning is one of the main feature of the proposed Rule Base. Self-learning is especially useful for situations in which the dynamical system has a highly complex behavior such that even experienced engineers have difficulties in designing control laws. For such systems, additional rules are needed to supplement the experts' knowledge. An obvious way to gain additional knowledge is to experiment with an approximately realistic model of the system by feeding the model with various reference command inputs and feedback control efforts and observing the output behaviors. Such experimenting is known as *reinforcement learning* in the artificial neural network (ANN) literature.

The main advantage of using SAM is that it is an associative memory with explicit storage of training samples. Each training sample can be interpreted as a rule. The rules obtained through reinforcement learning are explicitly stored and they can be modified or augmented by human experts. Such modification or augmentation is useful when the model of the dynamical system is not entirely accurate and a human expert can modify a rule learned from the approximate model to incorporate dynamics not described by the model. Due to the fact that different rules can fire over overlapping regions, the rules base resembles a fuzzy rule base.

A learning algorithm patterned after the dynamic programming is also proposed. Two unstable dynamical systems artificially created are used for testing and the Rule Base was used to generate a feedback control to improve command following ability of the otherwise uncontrolled systems. The numerical results are very encouraging. The controlled systems exhibit a more stable behavior and a better capability to follow reference commands.

Since SAM is the essential part of the Rule Base, we now briefly describe the distinctive features of SAM and basic motivation behind the creation of SAM. SAM can be considered as a nonlinear function approximator. To obtain the best approximation,

techniques of the classical approximation theory, regression theory, and system identification theory, which include curve-fitting, Volterra and other basis function approximation methods, spline methods and others could be applied. In designing SAM, we use *local linear approximation* or *piecewise linear approximation* to represent the identification model. Similar to the classical spline technique, we employ linear interpolation to generate a recalled output for an input which is never seen before.

This paper is organized as follows. Section II motivates the selection of a model class for the dynamical systems under consideration. Section III provides a description of SAM. Section IV describes the reinforcement learning and section V presents simulation results. The paper concludes in section VI.

## II. THE NONLINEAR INPUT-OUTPUT MODEL

In this paper, we assume the nonlinear dynamical system is described in the following Nonlinear MIMO (Multi-Input and Multi-Output) Input-Output form:

$$y(k) = f(y(k-1), y(k-2), \dots, y(k-n), u(k), u(k-1), \dots, u(k-q)), \quad (1)$$

where  $y(k) \in \mathbb{R}^p$ ,  $u(k) \in \mathbb{R}^m$ ,  $k$  is a discrete time index, and  $f(\cdot)$  is a general vector-valued nonlinear function of multiple variables. The above system could represent either a genuine discrete-time system or a sampled continuous-time system.

The above input-output model is also known as the Nonlinear Auto-Regression with eXogenous inputs (NARX)[4]. The above model also includes dynamical systems with noise and disturbance, either at the input or at the output, or at both places. The overall input vector  $u(k)$  could be decomposed into three parts: the control input components, the disturbance input components (i.e., the un-intended inputs either due to noise or exogenous disturbances), and the measurement noise components.

## III. A BRIEF DESCRIPTION OF SAM

### A. The Overlapping Local Linear Approximation

The approximation method adopted for SAM is an *overlapping local linear approximation* (OLLA). Consider the generic scalar function approximation problem:

$$y = f(x) \quad , y \in \mathbb{R}^1, x \in \mathbb{R}^d, \quad (2)$$

For each  $x$  of interest, we assume that there exist a neighborhood of  $x$ ,  $N(x)$ , such that, for all  $\bar{x} \in N(x)$ ,  $f(\bar{x})$  is well approximated by a linear functional:

$$f(\bar{x}) = a^T \bar{x} + b, \quad (3)$$

where  $a$  is a  $d$ -dimensional weight vector and  $b$  is a scalar. The function can be viewed in the  $\mathbb{R}^{d+1}$  space as a linear hyperplane by defining the augmented state vector  $z = [1, x^T]^T$ ,

and the augmented weight vector  $w = [b, a^T]^T$ . The hyperplane is then described by the equation  $f(x) = w^T z$ .

To determine the local hyperplane, only  $d + 1$  linearly independent prototypes - a prototype is defined to be a vector of the form  $[\bar{x}^T, f(\bar{x})]^T$  - from the neighborhood  $N(x)$  will be needed. If there are exactly  $d + 1$  linearly independent prototypes available, one can solve the following linear equation to obtain the local parameters  $w$ :

$$\begin{bmatrix} 1 & \bar{x}_1^1 & \dots & \bar{x}_d^1 \\ 1 & \bar{x}_1^2 & \dots & \bar{x}_d^2 \\ & & \vdots & \\ 1 & \bar{x}_1^d & \dots & \bar{x}_d^d \end{bmatrix} \begin{bmatrix} b \\ a_1 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} f(\bar{x}^1) \\ f(\bar{x}^2) \\ \vdots \\ f(\bar{x}^{d+1}) \end{bmatrix} \quad (4)$$

Once the local  $w(x)$  is determined, the recalled value  $f(x)$  can be computed simply via the formula  $f(x) = a^T x + b$ .

Now suppose there are less than  $d + 1$  linearly independent prototypes available, i.e., there are less equations to determine uniquely the local weight  $w(x)$ . There are many options here, and we decided to use the minimum norm solution to (4). The minimum norm solution is equivalent to a least square minimization problem:

$$\begin{aligned} \min \quad & \|w\|^2 \\ \text{s.t.} \quad & Aw = \bar{f} \end{aligned} \quad (5)$$

where  $A$  is matrix in the left hand side of equation (4),  $\bar{f}$  is the vector in the right hand side of (4), and the  $\|\cdot\|$  is the usual Euclidean norm ( $\ell^2$  norm). The solution to (5) is well-known: a pseudo-inverse solution described by the following equation:

$$w = A^T(AA^T)^{-1}\bar{f}. \quad (6)$$

We now briefly describe the *storing* and *retrieval* mechanism of the OLLA method. In storing, a new sample  $[x^T, f(x)]^T$  will be stored in its entirety, if  $f(x)$  cannot be adequately linearly approximated by the already stored prototypes in  $N(x)$ . Let  $\tilde{f}(x)$  denote the value recalled from the present memory, i.e., with no more than  $d + 1$  prototypes stored in the memory in the neighborhood  $N(x)$ ,  $\tilde{f}(x)$  is computed based on (3) with the weights computed using either (4) or (6). The value  $\tilde{f}(x)$  is said to be *recalled* from the memory. The user of the linear SAM then chooses a tolerance  $\varepsilon_2$  such that if

$$|f(x) - \tilde{f}(x)| > \varepsilon_2, \quad (7)$$

then the sample  $[x^T, f(x)]^T$  is stored into the memory.

The reason that the approximation method described above is called an *overlapping* method is that, in a small neighborhood, the function could be approximated by several linear hyperplanes computed based on several overlapping (intersecting) sets of prototypes. This overlapping property is the main difference between the linear SAM approximation approach and the classical local linear parametric regression method [5].

## B. The Architecture of SAM

## C. The Storage and Retrieval Scheme of SAM

We now describe the detailed computation scheme to implement the storing and retrieval schemes aiming to minimize searching time for both storing (learning) and recall. There are many ways to implement these computation structures. The description here is most conveniently interpreted as a sequential algorithm. However, the algorithm can be easily parallelized given a proper hardware architecture.

We have developed three storing schemes: *tree scheme*, *mesh scheme*, and the *hybrid scheme*. In this paper, we will only described the mesh scheme. A simple mesh storing scheme is described as follows. In the following description, let the current training sample be  $x$ . Let  $\varepsilon_1 > 0$  be a user specified scalar such that a linear interpolation of  $x$  by a set of  $d + 1$  closest vectors to  $x$   $\{\bar{x}^i : i = 1, \dots, d + 1\}$  will be allowed only if

$$\|x - \bar{x}^i\| \leq \varepsilon_1. \quad (8)$$

The condition(8) will be referred to as the  $\varepsilon$ -neighborhood condition. Define the interpolation index:

$$I(x) = |f(x) - \bar{f}(x)|, \quad (9)$$

where  $\bar{f}(x)$  is the recalled value generated by SAM for  $x$ .  $\varepsilon_4$  is another user-specified parameter which is used by the algorithm to define a hypercube neighborhood. The only requirement is that the hypercube region defined by

$$\{\bar{x} : \bar{x}_i - \varepsilon_4 < x_i < \bar{x}_i + \varepsilon_4, \forall i = 1, \dots, d, \}, \quad (10)$$

contains the  $\varepsilon$ -neighborhood defined by (8). The mesh will be called the *SAM mesh*.

1. Initialization: Let the first training sample be  $x$ . Then let the entry node to the mesh represent the vector  $x$  and each node that will be added to the mesh represent a particular prototype. The node storing  $x$  will have  $2d$  pointers pointing to the set of mesh neighbors:

$$\begin{aligned} x^1 &= [x_1^H, x_2, \dots, x_d]^T, \\ x^2 &= [x_1^L, x_2, \dots, x_d]^T, \\ x^3 &= [x_1, x_2^H, x_3, \dots, x_d]^T, \\ x^4 &= [x_1, x_2^L, x_3, \dots, x_d]^T, \\ &\dots \\ x^{2d-1} &= [x_1, x_2, \dots, x_{d-1}, x_d^H]^T \\ x^{2d} &= [x_1, x_2, \dots, x_{d-1}, x_d^L]^T, \end{aligned} \quad (11)$$

where

$$x_i^L < x_i < x_i^H, \quad x_i^H - x_i^L \leq 2\varepsilon_4, \quad \forall i = 1, \dots, d, \quad (12)$$

are components of either *genuine* or *pseudo* prototypes – if no genuine prototype vectors satisfying (12) are found, then create artificial (pseudo) prototypes to make up the mesh and to mark boundaries of the mesh. A node storing a pseudo prototype  $x$  does not carry actual value of  $f(x)$ .

2. For the current training prototype  $x$ , compute the interpolation set and the interpolation index as follows: search in an  $\varepsilon$ -neighborhood of  $x$  to find a set of  $d + 1$  closest vectors to  $x$ , denoted by  $\tilde{N}(x)$ . Compute the interpolation index of  $x$  as in (9).
3. For the current training prototype  $x$ , check if  $f(x)$  can be well interpolated from previously stored prototypes according to (7). If this is so, the current training sample is discarded.
4. Else, extend the SAM mesh by adding  $x$  and  $f(x)$  to the SAM mesh.

The retrieval scheme for the mesh scheme is trivial: Suppose the cue vector is  $x$  and SAM is asked to supply an approximate  $f(x)$ .

1. Retrieve the  $\varepsilon$ -neighborhood set  $\tilde{N}(x)$  of  $x$ .
2. If there is a almost matching prototype, say  $\bar{x}$ , then return the value  $f(\bar{x})$ . Otherwise compute the recalled values based on (10).

#### IV. THE REINFORCEMENT LEARNING ALGORITHM

The reinforcement learning algorithm proposed is described below. First we describe the feedback structure. Let  $k$  be the discrete time index. Let  $u_f(k)$  be the reference command input at time  $k$ . For each  $k$  the algorithm iterates through the index  $i$  to generate a desirable incremental feedback control  $u_e^i(k)$ . The overall input  $u(k)$  is the difference between reference input  $u_f(k)$  and feedback control  $u_c(k)$ :  $u(k) = u_f(k) - u_c(k)$ . For training, the overall feedback control  $u_c(k)$  is decomposed into two parts: the current control  $u_c(k)$  recalled from SAM, and the  $i$ -th trial incremental control  $u_e^i(k)$ :  $u_c^i(k) = u_e^i(k) + u_c(k)$ . Let  $J^i(k) = (y^i(k) - u_f(k))^2$  be the error at time  $k$  using the  $i$ -th trial incremental control. At time index  $k$ , the following is done:

1. Set  $u_c(k) = SAM(y(k))$ .
2. Set  $i \leftarrow 1$ .
3. Generate a trial incremental control  $u_e^i(k)$ ;
4. Set  $u(k) = u_f(k) - (u_e^i(k) + u_c(k))$  and generate the output  $y^i(k + 1)$  with  $u(k)$ ;
5. If  $J^i(k) < J^{i-1}(k)$ , store the relation  $y^i(k) \rightarrow u_e^i(k) + u_c(k)$  into SAM; else set  $i \leftarrow i + 1$  and go to step 3.
6. Set  $k \leftarrow k + 1$

## V. NUMERICAL SIMULATION RESULTS

We tested two artificial SISO (Single-Input-Single-Output) systems. The error measure we use to gauge the overall performance of the predictor is a normalized:

$$E = \frac{\frac{1}{N} \sum_{i=1}^N |y_d(k) - y(k)|}{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_d(k))^2}} \quad (13)$$

where  $y_d(k)$  is the actual output at time  $k$ ,  $y(k)$  is predicted output at time  $k$ ,  $N$  is the total number of samples taken in time.

### A. Example 1:

The SISO nonlinear system is described by the following equation:

$$y(k) = u(k) + 0.2y(k-1) - 0.3(y(k-1)y(k-3))^{\frac{2}{3}} - 0.5(y(k-2)y(k-4))^2,$$

where  $y(k)$  is the output, and  $u(k)$  is the input. We made seven different tests. In the first test, we trained the system with two separate ramp inputs with slopes 0.01 and 0.009 and tested the system with a ramp input of slope 0.0095. The second test is similar to the first test except that there is a output white noise of magnitude .01. In the third test, we trained the system with two separate step inputs with magnitudes 1.0 and 0.9 and tested the system with a step input of magnitude .095. In the fourth test, we trained the system with two separate step inputs with magnitudes 1.2 and 1.3 and tested the system with a step input of magnitude 1.25. In the fifth test, we trained the system with two sinusoids inputs and tested the system with an input which is added to a sinusoid. The sixth test is similar to the first test except that there is a output white noise of magnitude .01. In the seventh test we trained the system with two ramp-with-step inputs with step magnitudes at 1.1 and 1.0, and we tested with a ramp-with-step input at 1.05.

From the Figures attached, it is clear that the controlled system has a better command-following capability than the uncontrolled. The only drawback is that the learning algorithm does not seem to perform well when there is an output noise.

### B. Example 2:

The nonlinear system is described by the following equations:

$$y(k) = \cos\left[\frac{1}{2}u(k)y(k-1)\right] + (y(k-1)y(k-3))^2.$$

This system is highly unstable, and in this example we demonstrate the stabilizing ability of the Rule Base feedback control. We trained the systems with two step inputs with magnitudes 0.55 and 0.45 and tested the system with a step input of magnitude of 0.5. The uncontrolled system exploded at around  $k = 20$  while the controlled system is marginally stable; it did not explode.

## VI. CONCLUSIONS

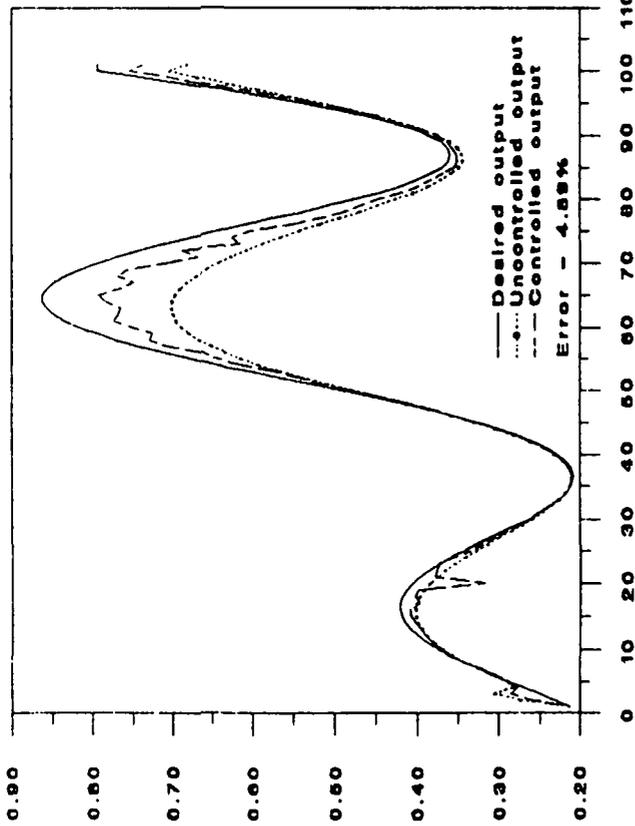
In this paper, a self-learning Rule Base for command following in dynamical systems is presented. The learning is accomplished through reinforcement learning using an associative memory. A learning algorithm patterned after the dynamic programming is also proposed. Two unstable dynamical systems artificially created are used for testing and the Rule Base was used to generate a feedback control to improve command following ability of the otherwise uncontrolled systems. The numerical results are very encouraging. The controlled systems exhibit a more stable behavior and a better capability to follow reference commands.

There are several directions of further research following this preliminary work. One is to improve the reinforcement learning algorithm so that the feedback controlled system responses will more closely follow the reference inputs. We intend to borrow insights from dynamic programming as the reinforcement learning algorithm proposed is very similar to the standard dynamic programming algorithm. Another is to test the self-learning Rule Base with realistic dynamical systems, especially systems with model uncertainty and output noise. For realistic systems, it would be interesting to investigate how a human expert can modify and add to the learned Rule Base so as to incorporate his own knowledge into the final Rule Base. A third direction is to apply the self-learning Rule Base to other control problems.

## REFERENCES

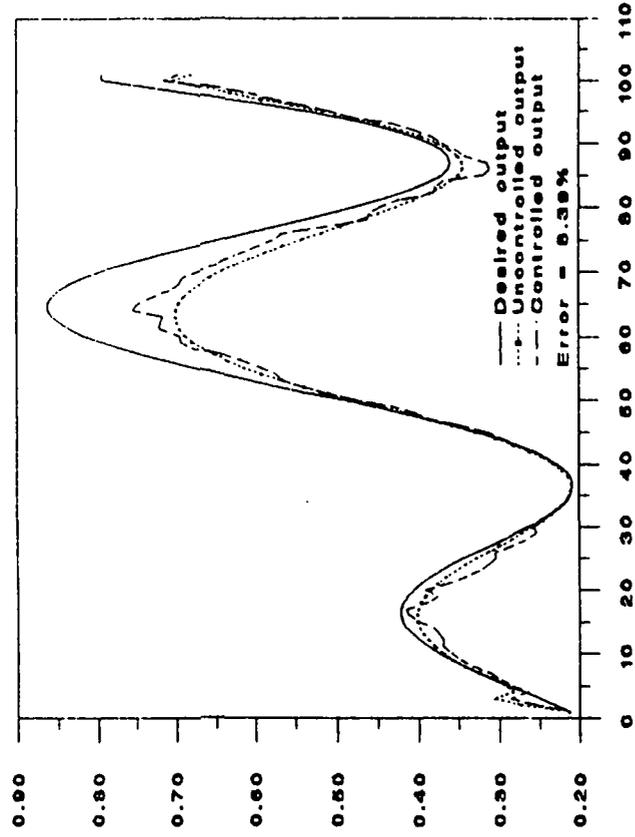
- [1] James P. Ignizio, *Introduction to Expert Systems: The development and Implementation of Rule-Based Expert Systems*, 1991, New York, N.Y.: McGraw-Hill, Inc.
- [2] Robert Hecht-Nielsen, *Neurocomputing*, 1991, Reading, MA: Addison-Wesley.
- [3] L. A. Zadeh, "A Theory of Approximate Reasoning," *Machine Intelligence*, Vol. 9, J. Hayes, D. Michie, and L. Mikulich (eds.), New York: Elsevier, 1979.
- [4] Chen, S., and S. A. Billings, "Representations of non-linear systems: the NARMAX model," *International Journal of Control*, 49, 3, 1013-1032, 1989.
- [5] Cleveland, W.S., "Robust Locally Weighted Regression and Smoothing Scatter Plots," *J. Amer. Statist. Assoc.*, Vol.-74, pp. 828-836, 1979.
- [6] Friedman, J.H., "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, Vol. 19, No. 1, pp. 1-141, March, 1991.

Composite testing without white noise



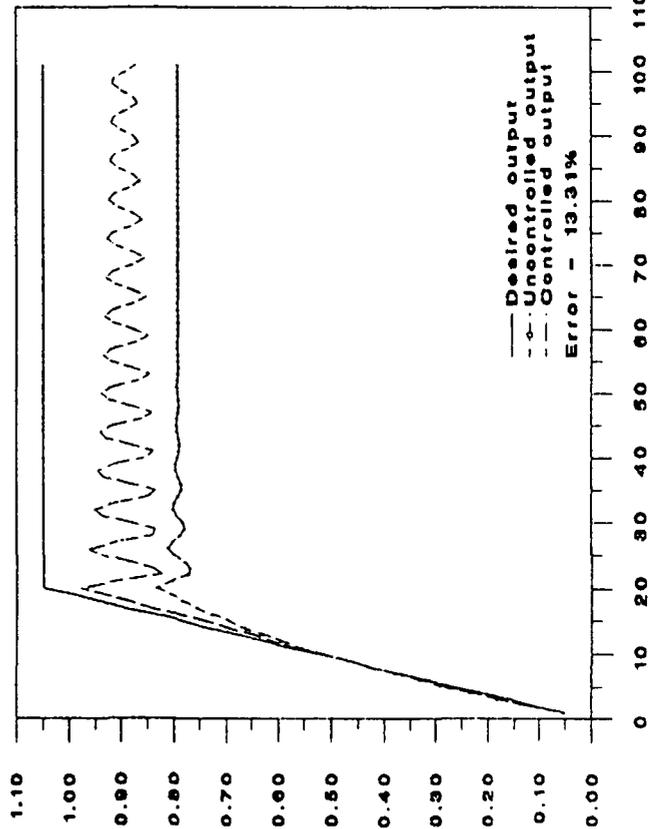
Example 1: test 5

Composite testing with white noise



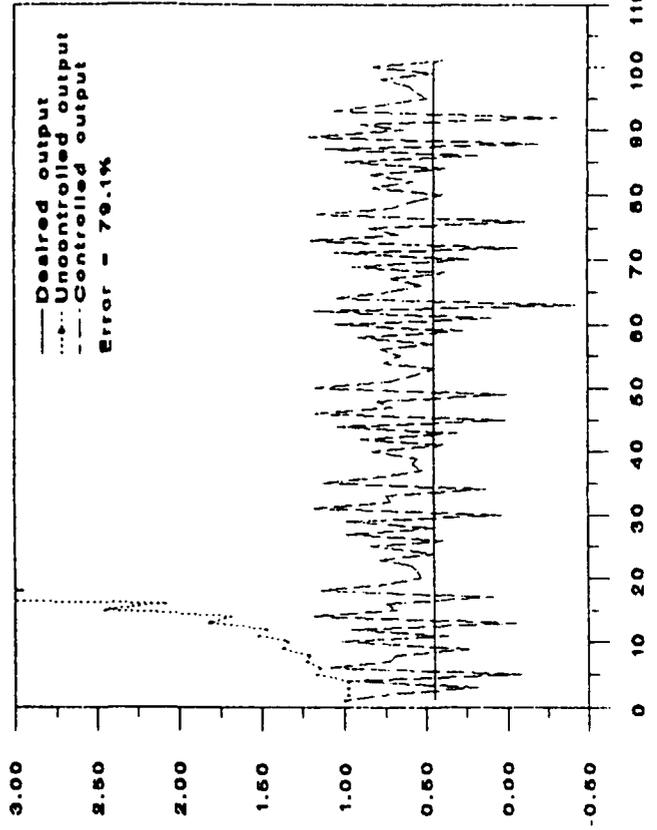
Example 1: test 6

Ramp and step testing.



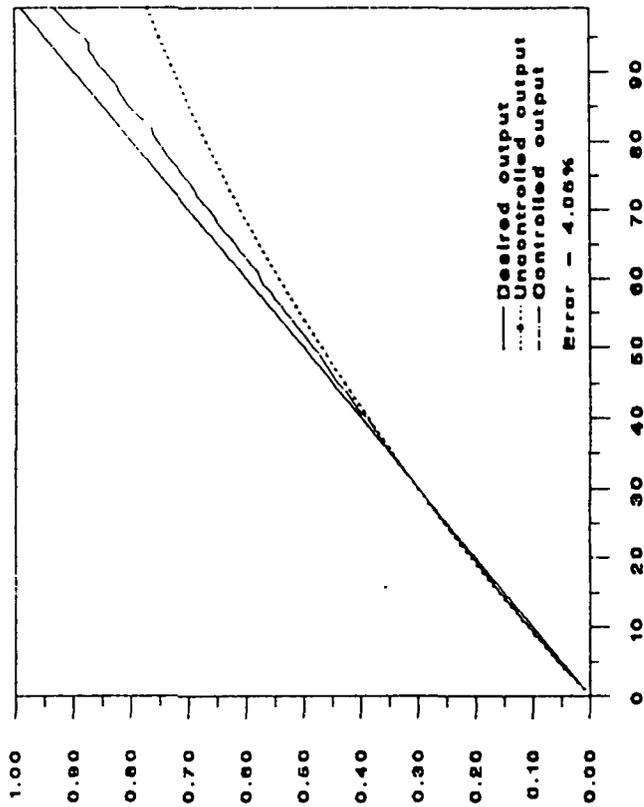
Example 1: test 7

Step testing for unstable non-linear system



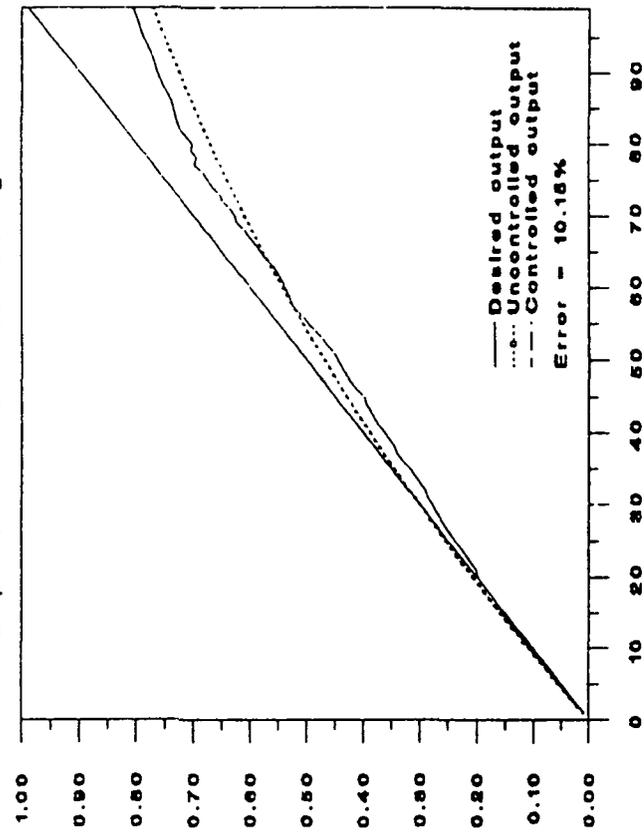
Example 2

Ramp testing



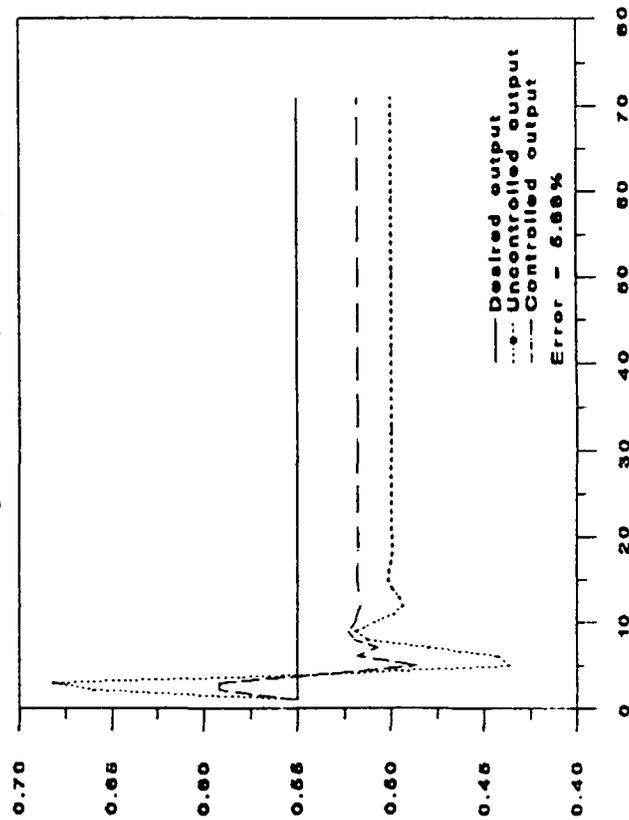
Example 1 : test 1

Ramp with white noise testing



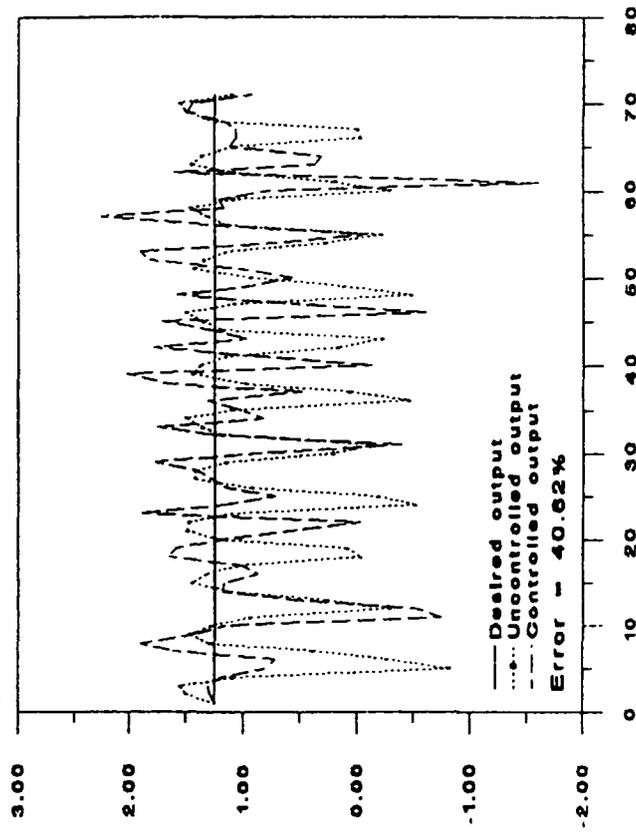
Example 1 : test 2

Low magnitude step testing



Example 1 : test 3

High magnitude step testing (oscillatory)



Example 1 : test 4

# Adaptive Defuzzification for Fuzzy Systems Modeling

Ronald R. Yager and Dimitar P. Filev  
Machine Intelligence Institute  
Iona College  
New Rochelle, NY 10801

## ABSTRACT:

*We propose a new parameterized method for the defuzzification process based on the simple M-SLIDE transformation. We develop a computationally efficient algorithm for learning the relevant parameter as well as providing a computationally simple scheme for doing the defuzzification step in the fuzzy logic controllers. The M-SLIDE method results in a particularly simple linear form of the algorithm for learning the parameter which can be used both off and on line.*

## 1. Introduction

Recently with the intensive development of fuzzy control [1, 2], the problem of selection of a crisp representation of a fuzzy set, defuzzification has become one of the most important issues in fuzzy logic. In [3, 4] it was shown that the commonly used defuzzification methods, Center of Area (COA) and Mean of Maxima (MOM) [1, 2], are only special cases of a more general defuzzification method, called Generalized Defuzzification via Basic Defuzzification Distribution (BADD). The BAD Distribution  $v_i, i=(1, n)$  of a fuzzy set  $D$  with membership function  $D(x_i) = w_i, w_i \in [0, 1]$ , is derived from its possibility distribution by use of the BADD transformation:

$$v_i = \frac{w_i^\alpha}{\sum_{j=1}^n w_j^\alpha}, \quad \alpha \geq 0 \quad (1)$$

The BADD transformation converts the possibility distribution  $w_i$  to a probability distribution  $v_i$ , in a manner that preserves the features of  $D, w_i > w_j \Rightarrow v_i \geq v_j$  and  $w_i = w_j \Rightarrow v_i = v_j$ . For  $\alpha = 1$  the BADD transformation converts proportionally the possibility distribution  $w_i, i=(1, n)$  to BAD distribution  $v_i, i=(1, n)$ . For  $\alpha > 1$  it discounts the elements of  $X$  with lower grade of membership  $w_i$ . Through parameter  $\alpha$  the BADD transformation relates the probability distribution  $v(x)$  to our confidence in the model [3, 4]. An increasing of  $\alpha$  is associated with a decrease of uncertainty, decreasing of entropy and an increase in confidence. The defuzzified value obtained via the BADD approach is defined as the expected value of  $X$  over the BAD distribution  $v_i, i=(1, n)$ :

$$d^{BADD} = \sum_{i=1}^n \frac{x_i w_i^\alpha}{\sum_{j=1}^n w_j^\alpha}, \quad \alpha \geq 0 \quad (2)$$

It is evident, that for fixed  $\alpha$ , the defuzzified value  $d^{BADD}$ , minimizes the mean square error,  $E\{(x - d^{BADD})^2\}$ . Thus the BADD defuzzified value is the optimal defuzzified value in the sense of minimizing the criterion

$$\sum_i (x_i - d^{\text{BADD}})^2 p_i \quad (3).$$

The main conclusion of this approach was that the best defuzzified value in the sense of above criterion can be obtained by adaptation of parameter  $\alpha$  by learning. Unfortunately the problem of learning the parameter  $\alpha$  from a given data set using directly expression (2) is a constrained nonlinear programming problem and its solution is difficult in real control applications. In this paper we solve the learning problem by the introduction of a new transformation of the possibility distribution  $w_i, i=(1, n)$  to the probability distribution  $v_i, i=(1, n)$ , called the Modified SemiLinear Defuzzification (M-SLIDE) transformation. The introduction of this new transformation results in a simple linear expression for the defuzzified value involving one parameter. An algorithm for learning the parameter is proposed.

## 2. M-SLIDE Defuzzification Technique

Let the probability distribution  $u_i, i = (1, n)$  be obtained by the proportional transformation (normalization) of  $w_i$ ,

$$u_i = c w_i = \frac{w_i}{\sum_{j=1}^n w_j}, \quad i=(1, n). \quad (4)$$

The following transformation of the probability distribution  $u_i, i=(1, n)$  to a probability distribution  $v_i, i=(1, n)$  is defined as the M-SLIDE transformation:

$$v_i = \begin{cases} \frac{1}{m} [1 - (1 - \beta) \sum_{j \in M} u_j] & \text{if } i \in M \\ (1 - \beta) u_i & \text{if } i \notin M \end{cases} \quad (5)$$

where  $m = \text{card}(M)$  is the cardinality of the set  $M$  of elements with maximal membership grades:  
 $M = \{i \mid w_i = \text{Max}_j[w_j]\}$

The derivation of the M-SLIDE transformation is expressed in detail in Yager & Filev [5]

The following theorem [5] shows some of the significant properties of the probability distribution obtained via the M-SLIDE transformation.

**Theorem 1:** Let  $w_i, i=(1, n)$  be the possibility distribution of a given fuzzy set and let  $v_i, i=(1, n)$  be obtained by application of transformations (4) followed by (5). Then it follows:

- i. distribution  $v_i, i=(1, n)$  is a probability distribution;
- ii.  $w_i = w_j \Rightarrow v_i = v_j, \forall i, j=(1, n)$  (identity);
- iii.  $w_i > w_j \Rightarrow v_i \geq v_j, \forall i, j=(1, n)$  (monotonicity)
- iv.  $\beta = 0 \Rightarrow v_i = \frac{w_i}{\sum_{j=1}^n w_j}, i=(1, n);$
- v.  $\beta = 1 \Rightarrow v_i = 0, i \notin M$  and  $v_i = \frac{1}{m}, i \in M.$

An immediate consequence of Theorem 1 is that the entropy of the M-SLIDE Distribution  $v_i$ , is maximal for  $\beta = 0$  and minimal for  $\beta = 1$ .

When using the M-SLIDE transformation to obtain the probability distribution  $v_i$  the expected value,  $d$ , with respect to the elements  $x_i$  of support set is

$$d = \sum_{i=1}^n v_i x_i = (1-\beta) \sum_{i \in M} u_i x_i + \frac{1}{m} [1 - (1-\beta) \sum_{i \in M} u_i] \sum_{j \in M} x_j$$

$$d = (1-\beta) \sum_{i \in M} u_i (x_i - d^{\text{MOM}}) + d^{\text{MOM}}$$

where  $d^{\text{MOM}}$  is the MOM defuzzified value,

$$d^{\text{MOM}} = \frac{1}{m} \sum_{j \in M} x_j.$$

It is evident that expected value  $d$  generalizes the MOM defuzzified value.

**Definition 1.** The process of selection of a deterministic value from the universe of discourse of a given fuzzy set by evaluation of the expected value  $d$  is called the Modified Semi Linear Defuzzification (M-SLIDE) Method. The defuzzified value, denoted  $d^{\text{MS}}$ , obtained by application of the M-SLIDE method is called the M-SLIDE value and is defined as

$$d^{\text{MS}} = (1-\beta) \sum_{i \in M} u_i (x_i - d^{\text{MOM}}) + d^{\text{MOM}}.$$

The next Theorem shows the relationship between the M-SLIDE method and the commonly used COA and MOM defuzzification methods.

**Theorem 2.** The M-SLIDE method reduces to the COA defuzzification method for  $\beta = 0$  and to the MOM defuzzification method for  $\beta = 1$ .

**Proof.** For  $\beta = 0$

$$d^{\text{MS}} = \sum_{i \in M} u_i x_i + \frac{1}{m} \sum_{j \in M} x_j = \sum_{i \in M} c w_i x_i + c w_{\text{max}} \sum_{j \in M} x_j$$

$$d^{\text{MS}} = \frac{1}{\sum_{j=1}^n w_j} \left[ \sum_{i \in M} w_i x_i + w_{\text{max}} \sum_{j \in M} x_j \right] = d^{\text{COA}}$$

where by  $d^{\text{COA}}$  we denote the defuzzified value obtained by the COA defuzzification method.

For  $\beta = 1$ ,  $d^{\text{MS}} = d^{\text{MOM}}$ .

**Theorem 3.** The following expressions of the M-SLIDE defuzzified value,  $d^{\text{MS}}$ , are equivalent:

$$d^{\text{MS}} = (1-\beta) \sum_{i \in M} u_i (x_i - d^{\text{MOM}}) + d^{\text{MOM}}$$

$$d^{\text{MS}} = \beta \sum_{i \in M} u_i (d^{\text{MOM}} - x_i) + d^{\text{COA}}$$

$$d^{\text{MS}} = \beta d^{\text{MOM}} + (1-\beta) d^{\text{COA}}$$

$$d^{\text{MS}} = \beta (d^{\text{MOM}} - d^{\text{COA}}) + d^{\text{COA}}$$

**Proof.**

$$d^{\text{MS}} = (1-\beta) \sum_{i \in M} u_i (x_i - d^{\text{MOM}}) + d^{\text{MOM}}$$

$$\begin{aligned}
&= \beta \sum_{i \in M} u_i (d^{\text{MOM}} - x_i) + \sum_{i \in M} u_i (x_i - d^{\text{MOM}}) + d^{\text{MOM}} \\
&= \beta \sum_{i \in M} u_i (d^{\text{MOM}} - x_i) + \sum_{i \in M} u_i x_i - \sum_{i \in M} u_i d^{\text{MOM}} + d^{\text{MOM}} \\
&= \beta \sum_{i \in M} u_i (d^{\text{MOM}} - x_i) + \sum_{i \in M} u_i x_i - (1 - m u_{\text{max}}) d^{\text{MOM}} + d^{\text{MOM}} \\
d^{\text{MS}} &= \beta \sum_{i \in M} u_i (d^{\text{MOM}} - x_i) + d^{\text{COA}} \\
&= \beta \sum_{i \in M} u_i d^{\text{MOM}} - \beta \sum_{i \in M} u_i x_i + d^{\text{COA}} \\
&= \beta (1 - m u_{\text{max}}) d^{\text{MOM}} - \beta \sum_{i \in M} u_i x_i + d^{\text{COA}} \\
&= \beta d^{\text{MOM}} - \beta m u_{\text{max}} \frac{1}{m} \sum_{i \in M} x_i - \beta \sum_{i \in M} u_i x_i + d^{\text{COA}} \\
&= \beta d^{\text{MOM}} - \beta d^{\text{COA}} + d^{\text{COA}} \\
d^{\text{MS}} &= \beta d^{\text{MOM}} + (1 - \beta) d^{\text{COA}} = \beta (d^{\text{MOM}} - d^{\text{COA}}) + d^{\text{COA}}
\end{aligned}$$

Theorem 3 provides convenient forms for the M-SLIDE defuzzified value as a linear function of the parameter  $\beta$ . In the next section we will use these forms for estimation of the parameter  $\beta$  in a learning procedure, capable of working on line.

### 3. Algorithm for Learning the M-SLIDE Parameter

In this section we solve the problem of learning the parameter  $\beta$  of the M-SLIDE method from a given sequence of fuzzy sets and desired defuzzified values. Furthermore we demonstrate that the M-SLIDE method can be used as an approximation of the Generalized Defuzzification Method via the BAD Distribution [3].

Assume we are given a collection of fuzzy sets  $U_k$  and the desired defuzzified values  $d_k$ ,  $k = (1, K)$ . We denote by  $d_k^{\text{MOM}}$  and  $d_k^{\text{COA}}$  the defuzzified values of the fuzzy sets  $U_k$  under MOM and COA defuzzification methods. The problem of learning of the parameter  $\beta$  is equivalent to the recursive solution of the set of linear equations:  $\beta * (d_k^{\text{MOM}} - d_k^{\text{COA}}) + d_k^{\text{COA}} = d_k$ ,  $k = (1, K)$ .

For simplification we denote:  $c_k = d_k^{\text{MOM}} - d_k^{\text{COA}}$  and  $y_k = d_k - d_k^{\text{COA}}$  and rewrite the set of equations that has to be solved in the form:  $c_k \beta = y_k$  for  $k = (1, K)$ .

In general there is no guarantee that this set of equations can be exactly satisfied for some value of  $\beta$  and also that  $c_k$  doesn't vanish for some  $k$ . For this reason we seek a **least squares** solution of the set of equations under the assumption of noisy observation data. The solution of this classical mathematical problem can be obtained by the application of a number of different techniques. In this paper we shall use an algorithm that is a deterministic version of the well known Kalman filter [6] which is usually used to solve the same kind least squares of errors

estimation problem for the case of dynamic systems.

The unknown parameter  $\beta$  that has to be estimated is regarded as a state vector of a hypothetical autonomous scalar dynamic system driven by the equations

$$\beta_{k+1} = \beta_k \text{ and } y_k = c_k \beta_k + \xi_k$$

where the term  $\xi_k$  denotes Gaussian white noise with covariance  $r_k$ . Then the recursive Kalman filter that gives the best estimate of the state vector  $\beta_k$  of this system has the form [6]:

$$\hat{\beta}_{k/k} = \hat{\beta}_{k/k-1} + g_k (y_k - c_k \hat{\beta}_{k/k-1}) \quad \text{i}$$

$$\hat{\beta}_{k+1/k} = \hat{\beta}_{k/k} \quad \text{ii}$$

$$P_{k/k-1} = P_{k-1/k-1} \quad \text{iii}$$

$$g_k = P_{k/k-1} c_k \frac{1}{c_k^2 P_{k/k-1} + r_k} \quad \text{iv}$$

$$P_{k/k} = P_{k/k-1} - g_k c_k P_{k/k-1} \quad \text{v}$$

Roughly speaking the Kalman filter calculates at every step the best estimate of the state vector as a sum of the prediction of  $\beta$  at step  $k$  from its value at step  $k-1$ ,  $\hat{\beta}_{k/k-1}$ , and a correction term proportional to the difference between current output value  $y_k$  and predicted output  $c_k \hat{\beta}_{k/k-1}$ . Equation iv calculates the varying gain,  $g_k$ , of the filter. The evolution of error covariance is given by equation v. Because of the static nature of the autonomous system  $\hat{\beta}_{k+1/k} = \hat{\beta}_{k/k} = \beta_k$  and  $P_{k/k-1} = P_{k-1/k-1} = P_{k-1}$  this significantly simplifies the algorithm to

$$\beta_k = \beta_{k-1} + g_k (y_k - c_k \beta_{k-1}) \quad \text{(vi)}$$

$$g_k = P_{k-1} c_k \frac{1}{c_k^2 P_{k-1} + r_k} \quad \text{(vii)}$$

$$P_k = P_{k-1} - g_k c_k P_{k-1} \quad \text{(viii)}$$

by combining vi and vii a more compact form of the algorithm is obtained

$$\beta_k = \beta_{k-1} + P_{k-1} c_k \frac{1}{c_k^2 P_{k-1} + r_k} (y_k - c_k \beta_{k-1}) \quad \text{(ix)}$$

$$P_k = P_{k-1} - P_{k-1}^2 c_k^2 \frac{1}{c_k^2 P_{k-1} + r_k} \quad \text{(x)}$$

Because usually we have no idea about the magnitude of the additive noise  $\xi_k$  we shall consider  $r_k = 1$ . Then equation (x) is further simplified and we receive the following final form of the Kalman filter algorithm for recursive least square solution of the original set of equations :

$$\beta_k = \beta_{k-1} + \frac{P_{k-1} c_k}{c_k^2 P_{k-1} + 1} (y_k - c_k \beta_{k-1}) \quad \text{xi}$$

$$P_k = \frac{P_{k-1}}{c_k^2 P_{k-1} + 1} \quad \text{xii}$$

Regarding the initial conditions, it can be argued [7] that a reasonable assumption is to consider  $\beta_0 = 0$  and nonnegative  $p_0$ .

The algorithm gives an unconstrained solution for  $\beta$ . Because of the requirement of  $\beta$  belonging to the unit interval, we shall restrict the solution  $\beta_k$  by applying a threshold to give the value  $\beta_k^*$  where

$$\beta_k^* = \begin{cases} 1 & \text{if } \beta_{k-1} + \Delta_k > 1 \\ 0 & \text{if } \beta_{k-1} + \Delta_k < 0 \\ \beta_{k-1} + \Delta_k & \text{otherwise} \end{cases}$$

where  $\Delta_k$  denotes the second term in the right part of xi,

$$\Delta_k = \frac{p_{k-1} c_k}{c_k^2 p_{k-1} + 1} (y_k - c_k \hat{\beta}_{k-1}).$$

The thresholding effect can be replaced by the following nonlinear expression:

$$\beta_k^* = 1 - 0.5 [1 - 0.5 (\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|) + |1 - 0.5 (\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|)|]$$

The algorithm for learning the M-SLIDE parameter, based on Kalman filter, can now be summarized in the following.

**Algorithm for learning the parameter  $\beta$  (M-SLIDE Learning Algorithm)**

1. Set  $\beta_0 = 0$ ;  $p_0 > 0$ .
2. Read a sample pair  $U_k, d_k$ .
3. Calculate: i.  $d_k^{\text{MOM}}$ ; ii.  $d_k^{\text{COA}}$ ; iii.  $c_k = d_k^{\text{MOM}} - d_k^{\text{COA}}$ ; iv.  $y_k = d_k - d_k^{\text{COA}}$
4. Update  $\beta_k, p_k$ :  $\beta_k = \beta_{k-1} + \frac{p_{k-1} c_k}{c_k^2 p_{k-1} + 1} (y_k - c_k \beta_{k-1})$  and  $p_k = \frac{p_{k-1}}{c_k^2 p_{k-1} + 1}$
5. Calculate  $\beta_k^*$  :

$$\beta_k^* = 1 - 0.5 [1 - 0.5 (\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|) + |1 - 0.5 (\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|)|]$$

6. Update the current estimate of the parameter  $\beta$ :  $\beta = \beta_k^*$ .

We note that since the estimate of the parameter  $\beta$  is determined sequentially there is no need to resolve the whole set of equations when a new pair of data pair  $(U_{k+1}, d_{k+1})$  becomes available for learning. The addition of a new data pair can be incorporated by just an additional iteration of the algorithm. This property of the algorithm allows it to be used for either off-line or on-line learning of the parameter  $\beta$ .

In the case when the *desired* defuzzified values, the  $d_k$ 's, are the defuzzified values obtained from the defuzzification method using the BADD distribution, the Algorithm can be used to get an associated M-SLIDE parameter  $\beta$  corresponding to a BADD transformation parameter  $\alpha$ .

The next example presents an application of the M-SLIDE learning algorithm.

**Example.** Assume our data consists of 10 fuzzy sets:

- $U_1 = \{0/3, 0.6/4, 1/5, .8/6, 0.9/7, 0/8\}$ ;  $U_2 = \{0/5, 0.9/7, 1/9, 1/11, 0.2/12, 0/13\}$ ;  
 $U_3 = \{0/2, 0.4/3, 0.8/4, 1/5, 0.5/6, 0/7\}$ ;  $U_4 = \{0/4, 1/5, 0.9/6, 1/7, 0.9/8, 0/9\}$ ;  
 $U_5 = \{0/6, 0.3/7, 1/8, 0.6/9, 1/10, 0/11\}$ ;  $U_6 = \{0/3, 0.2/4, 0.9/7, 1/9, 1/10, 0/12\}$ ;  
 $U_7 = \{0/1, 0.9/4, 0.5/5, 1/7, 0.4/8, 0/10\}$ ;  $U_8 = \{0/3, 0.5/7, 0.9/10, 1/11, 0.4/14, 0/16\}$ ;

$U_9 = \{0/5, 0.2/6, 1/7, 1/9, 0.1/10, 0/11\}$ ;  $U_{10} = \{0/4, 1/7, 0.5/8, 1/9, 0.7/10, 0/11\}$ .

We used the BADD defuzzification method to generate the *ideal* defuzzified values,  $d_k$ , associated with each of these fuzzy sets. In this way we formed six different data sets, each consisting of 10 pairs  $(U_k, d_k)$ . In each data set the  $d_k$ 's were generated by a different BADD parameter  $\alpha$ .

For each data set, using the M-SLIDE learning algorithm, we obtained the optimal estimate for the parameter  $\beta$ . The following tables show the results of the experimentation with our algorithm. In the tables below we note that  $d_k$  is the ideal value and  $d_k^c$  is the calculated defuzzification value using the M-SLIDE defuzzification procedure with the optimal estimated  $\beta$  parameter for that data set.

**DATA SET # 1 OPTIMAL ESTIMATED  $\beta = 0.00022$**

k	1	2	3	4	5	6	7	8	9	10
$d_k^c$	5.60	9.26	4.59	6.47	8.79	8.42	5.82	10.39	7.91	8.43
$d_k$	5.60	9.26	4.59	6.47	8.79	8.42	5.82	10.39	7.91	8.43

**DATA SET # 2 OPTIMAL ESTIMATED  $\beta = 0.10758$**

k	1	2	3	4	5	6	7	8	9	10
$d_k^c$	5.54	9.34	4.64	6.42	8.82	8.54	5.95	10.46	7.92	8.39
$d_k$	5.71	9.21	4.70	6.42	8.98	8.82	5.76	10.46	7.99	8.28

**DATA SET # 3 OPTIMAL ESTIMATED  $\beta = 0.22539$**

k	1	2	3	4	5	6	7	8	9	10
$d_k^c$	5.47	9.43	4.68	6.37	8.84	8.66	6.09	10.53	7.93	8.34
$d_k$	5.72	9.32	4.77	6.37	9.00	8.93	5.88	10.58	8.00	8.15

**DATA SET # 4 OPTIMAL ESTIMATED  $\beta = 0.66891$**

k	1	2	3	4	5	6	7	8	9	10
$d_k^c$	5.20	9.75	4.87	6.16	8.93	9.14	6.61	10.80	7.97	8.14
$d_k$	5.36	9.72	4.97	6.17	9.00	9.27	6.49	10.83	8.00	8.00

**DATA SET # 5 OPTIMAL ESTIMATED  $\beta = 0.92394$**

k	1	2	3	4	5	6	7	8	9	10
$d_k^c$	5.05	9.94	4.97	6.04	8.98	9.42	6.91	10.95	7.99	8.03
$d_k$	5.08	9.94	5.00	6.04	9.00	9.45	6.88	10.96	8.00	8.00

**DATA SET # 6    OPTIMAL ESTIMATED  $\beta = 0.97293$**

<b>k</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b><math>d_k^c</math></b>	5.02	9.98	4.99	6.01	8.99	9.47	6.97	10.98	8.00	8.01
<b><math>d_k</math></b>	5.03	9.98	5.00	6.01	9.00	9.48	6.96	10.99	8.00	8.00

It can be seen from the above example that the M-SLIDE learning algorithm learns values of the parameter  $\beta$  that allow a very good matching of the data set.

#### **4. References**

[1]. Lee, C. C., "Fuzzy logic in control systems: fuzzy logic controller, Part 1," IEEE Trans. on Systems, Man and Cybernetics 20, 404-418, 1990.

[2]. Lee, C. C., "Fuzzy logic in control systems: fuzzy logic controller, Part II," IEEE Trans. on Systems, Man, and Cybernetics 20, 419-435, 1990.

[3]. Filev, D. and Yager, R. R., "A generalized defuzzification method under BAD distributions," International Journal of Intelligent Systems 6, 687-697.

[4]. Yager, R. R. and Filev, D. P., "On the issue of defuzzification and selection based on a fuzzy set," Technical Report# MII-1201, Machine Intelligence Institute, Iona College, 1991.

[5]. Yager, R. R. and Filev, D. P., "SLIDE: A simple adaptive defuzzification method," IEEE Transactions on Fuzzy Systems, (To Appear).

[6]. Aoki, M., Optimization of Stochastic Systems, Academic Press: New York, 1967.

[7]. Deutsch, R., Estimation Theory, Prentice Hall: Englewood Cliffs, NJ, 1965.

# Design Issues of a Reinforcement-based Self-Learning Fuzzy Controller for Petrochemical Process Control

John Yen, Haojin Wang and Walter C. Daugherty

Center for Fuzzy Logic and Intelligent Systems Research  
Department of Computer Science  
Texas A&M University  
College Station, TX 77843-3112

## Abstract

Fuzzy logic controllers have some often-cited advantages over conventional techniques such as PID control, including easier implementation, accommodation to natural language, and the ability to cover a wider range of operating conditions. One major obstacle that hinders the broader application of fuzzy logic controllers is the lack of a systematic way to develop and modify their rules; as a result the creation and modification of fuzzy rules often depends on trial and error or pure experimentation. One of the proposed approaches to address this issue is a self-learning fuzzy logic controller (SFLC) that uses reinforcement learning techniques to learn the desirability of states and to adjust the consequent part of its fuzzy control rules accordingly. Due to the different dynamics of the controlled processes, the performance of a self-learning fuzzy controller is highly contingent on its design. The design issue has not received sufficient attention. The issues related to the design of a SFLC for application to a petrochemical process are discussed and its performance is compared with that of a PID and a self-tuning fuzzy logic controller.

## 1 Introduction

Conventional model-based control has the advantage of stability and proved optimality within the given range of operating conditions. For this reason, Proportional-Integral-Derivative (PID) control has been a major practical control technology for a long time. However, there are some serious limitations with this approach in dealing with ill-defined, non-linear and dynamic processes. One of the problems is its lack of adaptivity to the operating environment. When the operating conditions are out of the prescribed range, human intervention is needed to manually tune and adjust the operating parameters.

In the past few years, a great deal of interest has been generated in applying fuzzy logic and approximate reasoning to industrial process control and these efforts have resulted in various techniques of fuzzy control. The basic idea of fuzzy control is to transform human expert knowledge about controlling the process into fuzzy *if-then* rules and use approximate reasoning to deal with uncertainty and to derive the control actions. The advantage of fuzzy

control is that it can capture the imprecise and uncertain aspects of human reasoning and as a result the fuzzy controller can deal with those dynamic, ill-defined or non-linear systems more efficiently than conventional approaches. Since E. Mamdani [6] applied the basic concepts of fuzzy logic, coined by Lotfi Zadeh [7], to control in early 1970s, and especially during the past decade, there has been a great deal of research activity in this field, and many techniques and architectures have been proposed or developed.

Despite the advantages of the fuzzy logic controller mentioned above, there are some problems associated with the fuzzy controller that hinders its wider application. One of the issues is its lack of a systematic way to develop and modify its rules, and as a result the creation and modification of fuzzy rules often depend on trial and error or pure experimentation. Several approaches have been proposed to address this issue. One of the proposed approaches is to use a self-learning mechanism to learn the desirability of rules and modify the consequent part of the fuzzy rules accordingly.

The issue of self-learning is to let the system itself learn the proper control actions through a given number of trials. Several techniques have been developed or proposed to accomplish the goal of self-learning during recent years. Barto et al [2] proposed a learning mechanism composed of two neuron-like elements called the adaptive critic element (ACE) and the associative search element (ASE). Lee [5] integrated this idea into a fuzzy control system and applied it to the well-known pole-balancing problem. Chen [3] used a similar approach with slight modification, and applied it to three similar dynamic process. One important aspect of SFCL that has not been properly addressed is that the performance of a self-learning fuzzy controller is application dependent and the different dynamics of the controlled process requires different treatment in the design of a SFCL. In the following sections, the issues related to the design of a SFCL in general and for a particular petrochemical process are discussed.

## 2 Description of the Control Process and SFCL

The control process for this research is a simple gas-fired water heater, since it is widely used in the petrochemical industry and an accurate simulation model was available. The inlet water at a certain temperature and feed rate enter a stirred tank heated by a gas burner. At a certain point downstream the outlet water temperature is measured by a sensor. The resultant time delay is known as dead time. The controller calculates the temperature difference between the current value and the desired (or "setpoint") value, i.e., the error, and adjusts the valve controlling the gas supply accordingly. The initial temperature reading of the water tank is assumed to be at room temperature level. For a more detailed description of the control process, see [4]. The control task is first to heat the tank to the desired set point and then to keep the temperature at the desired level in the presence of sensor noise and changing operating conditions.

The self-learning fuzzy controller we developed is based on the approach proposed by Sutton, Barto and Lee and it is intended for application to industrial processes in general and to petrochemical processes in particular. The controller has two major components, namely, a fuzzy control component and a learning component. The fuzzy control component consists of a rule base which has a set of fuzzy rules and a fuzzy inference mechanism that uses the fuzzy rules and applies fuzzification and defuzzification operators to the input and output to obtain the actual control action. The learning component contains two neuron-like elements. They are the adaptive critic element (ACE) and the associative search element

(ASE) respectively. Initially, the consequent part of every control rules is initialized to an arbitrary fuzzy control value. When a rule fires with non-zero firing strength, the two neuron-like elements learn the desirability of the previous control action and adjust the weights of the fired rules. The consequent part is adjusted according to its weight. The control action is the result of applying a defuzzification operator to the control commands inferred by the fired rules. For more detailed information on the implementation of this type of controller, see Barto [2] and Lee [5].

The SFLC in this research has two input variables to the controller, namely, the error (difference between the current temperature reading and the set point) and the change of error (difference between the current temperature and the one at dead time steps back). The output variable is the amount of change to the valve that controls the gas supplied to the burner.

### 3 Design Issues

The performance of a SFLC is highly application dependent and one of the determining factors in the design of a SFLC is the dynamics of the control process. The design issues considered in this research are the choice of a training set, the choice of feedback, and the learning parameters.

#### 3.1 The choice of Training Cases

A SFLC first needs to go through a learning session to learn the proper control action through a certain number of trials. After learning, the controller is put into actual operation where the learned rules are applied. The issue regarding training cases is the choice of cases presented to the controller during the learning session.

The dynamics of the control process directly influences the choice of training cases. For those highly dynamic processes, what the control system encounters during the learning session tends to cover a wide range of operating conditions and consequently this results in a better knowledge base for the control system. With a broader knowledge base, the control system can perform well under the various operating conditions. Therefore, the choice of training cases is not a real issue. However for those processes which are less dynamic, it is likely that the knowledge acquired during the learning session is not sufficient to cover a wide range of operating conditions if the training cases are generated in the same fashion. Then, the choice of test cases becomes very important, because what the SFLC learns will, to a large extent, determine how it performs in the operating environment.

The central idea of the choice of training cases is to design the training cases in such a way that all operating conditions that we anticipate the control system might encounter should be covered in the learning session. One approach to accomplish this is to map the state space of the control system into a two-dimensional space like the one in Figure 1 and then design the training cases in such a way that they are complementary to each other and that taken together, they can cover most of the state space. In this figure, each numbered square corresponds to a state the control system can be in. NB, NS, ZE, PS, and PB are fuzzy sets used for the SFLC and they are abbreviations of negative big, negative small, zero, positive small and positive big respectively. For more than two state variables, we can use a similar approach to map the rule base into a hyperspace. A state space like this can be used to design the training cases for a SFLC. In this figure, each curve is the trajectory

of the fired rules for one trial of a training instance. Together, they form a region that defines a desirable performance curve for the control system. If the control system is in a state that is within the desirable performance curve, it is expected to perform well because it has the knowledge regarding this particular state in its knowledge base. However, if the control system falls out of the desirable performance curve, the performance may be poor. A simple example will illustrate this point well. If the desirable performance curve is the one shown in Figure 1, and if the initial state for the control system happens to be the one in the lower right corner of the state space where the error and the change of error both are positive big, the knowledge acquired during the learning session will not be sufficient for the control system to handle this case effectively. The goal of designing training cases is to have the desirable performance curve cover as many states as possible in the state space.

We now use the process for this research to illustrate the above idea. For the petrochemical process under consideration, the dynamics differs from the inverted pendulum that is often used to demonstrate the concept of reinforcement-based self-learning. The inverted pendulum is highly dynamic and the choice of training cases is easier because each training case tends to cover a large portion of the state space. Therefore, by randomly generating the training cases (i.e., the arbitrary initial angles and positions), the system is able to learn an appropriate response for most of the states in the state space. However, this is not the case for industrial processes in general and petrochemical processes in particular for the following two reasons:

- The slow-response nature of the process dynamics may result in smaller portion of the state space being covered during the learning session;
- With a proper choice of feedback, training a SFLC for an industrial process may not require a large number of training cases to reach the goal state.

Consequently, only a limited number of operating conditions are encountered in the learning session and the desired performance curve covers only a small portion of the state space. The choice of the training cases is thus the issue of ensuring that a large portion of the state space is covered. The suggested approach to this problem is to design multiple training cases which are complementary to each other so that taken together they can cover a large portion of the state space.

### 3.2 The choice of feedback

The basic idea of reinforcement learning is to use feedback from the environment to generate reinforcement signal that helps distinguish desirable states from undesirable states. The choice of feedback directly impacts the performance of a SFLC by influencing the quality of learning and the length of the learning cycle.

Reinforcement learning is implemented through the two neuron-like elements ACE and ASE. The ACE receives feedback from the environment and its main function is to provide a critique of the control action that took place at dead time steps back and in doing so, it generates an internal reinforcement signal to the ASE. The rationale is that when the process is moving from a less desirable state to a more desirable state, it should receive a positive reinforcement signal and when it moves in the opposite direction, it should receive a negative reinforcement signal. The choice of feedback directly impacts the quality and quantity of the internal reinforcement the ACE generates, and in turn it impacts the weight associated with each rule and eventually affects the control action the SFLC generates.

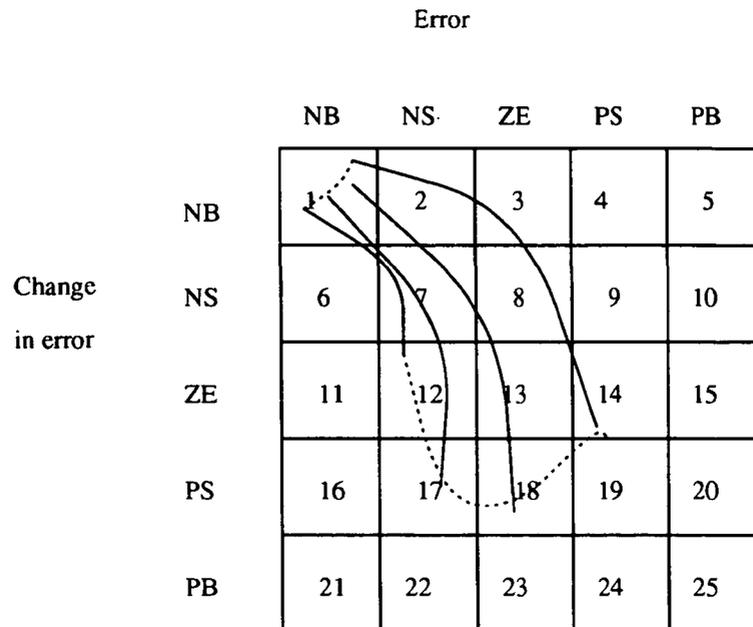


Figure 1: Trajectory of trained rule in state space

The choice of feedback is closely related to the dynamics of the control process. A difference in process dynamics may need a different choice of feedback to best suit the purpose of reinforcement. The method currently employed for the generation of feedback in the concept-demonstration control systems developed by Barto [2] and Lee [5] is to give a -1 as feedback once the control process falls into a "failure" state. All states that are outside a desirable region of the pole's angles are deemed failure states. This method does not suit petrochemical processes well for three reasons. First, there is a delay between a control action and the resultant response and the rules fired immediately before a failure state may not be the real "culprit." Secondly, for a less dynamic process, the feedback may be infrequent if only failure states cause feedback. Finally, the initial state of a training trial can be a state outside the desirable region (i.e., the initial temperature is not in  $[T-\alpha, T+\alpha]$  where  $T$  is the set point and  $\alpha$  is the threshold that specifies the desirable region) and if we give negative feedback to all the states outside the desirable region, then the process will never reach the the goal state. Taking into consideration the difference in the process dynamics, we discuss some general issues regarding the choice of feedback for reinforcement learning and then propose some design guidelines for addressing these issues.

First, we should ensure that no strong negative feedback is given while the control process is on its way to the goal state even though the intermediate states are failure states. This is one of the major differences between petrochemical processes and the processes used to demonstrate the concept of SFLC, like the inverted pendulum. On the other hand, a negative feedback should be generated once the control process falls into a failure state that is not on the desired performance curve that leads the process from the initial state to the goal state. This requires that we define the desired performance curve and distinguish those

failure states that are on the performance curve from those that aren't.

Next, the amount of domain-specific information used should be limited to a minimal level. The fundamental assumption for the self-learning fuzzy logic controller is that it should learn its own control action in the absence of knowledge about input and output relations.

Another factor is regarding how early or frequently feedback is generated. In order to shorten the learning cycle, ideally the feedback should be generated as frequently and as early as possible. However, there is a trade-off between a short learning cycle and the amount of domain specific information required. It is usually the case that to increase the frequency of feedback often requires more domain-specific information. A balance between the two can be struck depending on the availability of domain-specific information and the requirement for the length of the learning session. For instance, if the learning system is given the desired performance curve (which is highly specific to the particular process being considered), a feedback can be generated every cycle based on the distance between the current state and the corresponding state on the desired performance curve. However, such an approach is not feasible if the desired performance curve is not readily available. Under such a circumstance, a mechanism that generates feedback less frequently but relies on less domain-specific information should be used.

Having discussed these issues, we now outline some design guidelines for addressing them. First, the feedback can be expressed as a function of the factors it depends on. It can be expressed as  $Feedback(S)$ , where  $S$  stands for state. We generalize the notion of state-based feedback to the notion of performance-based feedback. A state-based feedback, as demonstrated by Barto, Sutton and Lee using the inverted pendulum problem, generates a feedback signal entirely based on the current state of the system. Therefore, a *performance-based feedback* incorporates the initial and goal states into the function for generating the feedback, in addition to the current state. It can be expressed as  $Feedback(S, I, G)$  where  $I$  stands for initial state and  $G$  for the goal state. The advantage of this approach is its flexibility. A state can be given different feedback depending on whether it is on the desired performance curve, which is determined by the initial operating conditions and the goal state. For instance, in Figure 2, the state  $s$  is the same state for cases  $a$  and  $b$ . Because the initial states are different for the two cases, the state  $s$  is on the desired performance curve in  $a$  but it is not in case  $b$ . By incorporating the initial state into the feedback function, we are able to give different feedback for the same state  $s$  under different circumstances. A variation of this method is to use global performance history instead of a single failure state to generate the feedback. It can be expressed as  $Feedback(I, G, \bar{S})$  where  $I$  and  $G$  are same as above and  $\bar{S}$  is the global performance history, e.g., the average of all errors. A method similar to this is employed in Y. Y. Chen's[3] system.

The second design guideline for generating feedback is to incorporate the performance objectives such as reaching time or overshoot to generate feedback. The performance objectives serve as constraints to the control process. Once the control process fails any of the performance objectives, feedback is generated. Thus, the feedback can be expressed as  $Feedback(S, O_1, \dots, O_n)$  where  $O_i$  represents  $i$ th performance objective. The method we employed for this research is a combination of using the global performance history and incorporating a performance objective into feedback function.

The third design guideline is to use general knowledge about the control process such as the dynamics of the process to generate feedback. This is control process dependent and detailed implementation hinges on the actual control process in question.

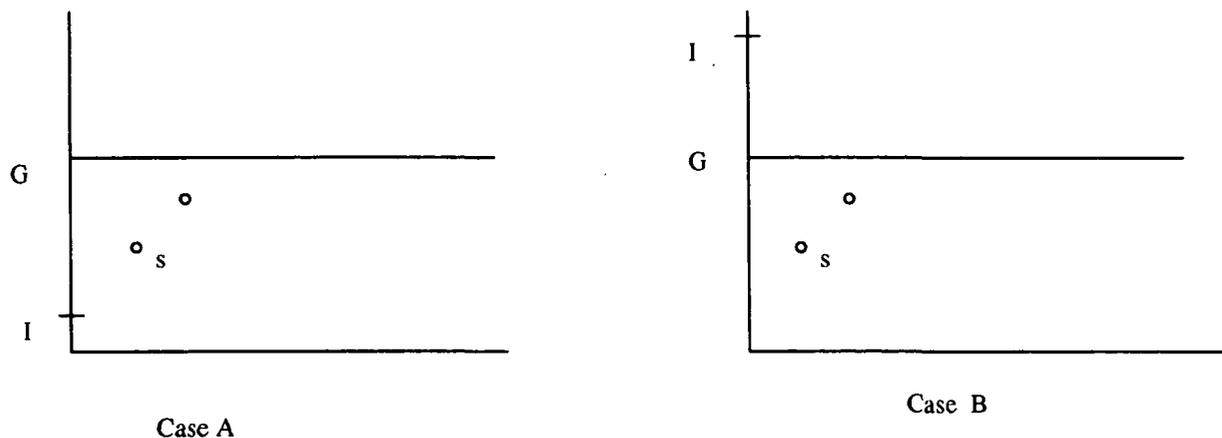


Figure 2: The two different training cases

### 3.3 Learning Parameters

There are many parameters for the learning rules. The values of those parameters are usually determined in a more or less trial and error fashion. It is a research issue as how to determine the parameter values systematically. Following are some observations about the relations between the parameter values and process dynamics.

#### *Rule trace decay parameter*

This determines how fast rule traces decay. Rule trace is the history of a rule's firing strength and frequency. This parameter is highly dependent on the dynamics of the control process. We observed that the more dynamic a process, the faster the decay and the less dynamic a process, the slower the decay. Intuitively, a large amount of information is likely to be required to compensate for the higher rate of decay for very dynamic processes.

#### *Sigmoid gain parameter*

This determines to what extent the weight of a rule is transferred into the consequent part of a fuzzy rule. This parameter is also highly related to the dynamics of the control process. It appears that the less dynamic the process is, the greater the sigmoid gain should be.

## 4 Empirical Experiments and Discussion

### 4.1 Description of Learning Rules

For the ACE, the learning rules are as follows:  
Internal reinforcement is defined as

$$\dot{r} = r(t) + \gamma p(t) - p(t-1)$$

where  $r(t)$  is feedback and  $p(t)$  is total desirability of all states at time  $t$ :

$$p(t) = \sum_{i=1}^n v_i(t)x_i(t)$$

where  $v_i$  is the desirability of the  $i$ th state and  $x_i$  is the firing strength of the  $i$ th rule. In turn,  $v_i$  is defined by

$$v_i(t) = v_i(t-1) + \beta r(t) \bar{x}_i(t),$$

where  $\bar{x}_i$  is the local memory trace defined by

$$\bar{x}_i(t) = \lambda \bar{x}_i(t-1) + (1-\lambda)x_i(t).$$

For the ASE, the learning rules are as follows:  
The weight of each fuzzy rule is determined by

$$w_i(t) = w_i(t-1) + \bar{a}(t-1) \bar{r}(t) \epsilon_i(t-d),$$

where  $\epsilon_i$  is the rule trace,  $d$  is the dead time delay and  $\bar{a}(t)$  is the dynamic positive learning rate. The rule trace  $\epsilon_i$  is given by

$$\epsilon_i(t) = \delta \epsilon_i(t-1) + (1-\delta)y(t)x_i(t),$$

where  $\delta$  is rule trace decay parameter, and

$$\bar{a}(t) = \frac{\alpha k}{k+t},$$

where  $\alpha$  is initial value and  $k$  is a weight freeze parameter. The consequent of each fuzzy rule is determined by a sigmoid function:

$$y_i = f(w_i(t), \text{noise}(t)),$$

where the dynamic sigmoid function  $f$  is defined by

$$f(x, t) = \frac{x}{T(x)+x} \text{ for } x > 0$$

$$f(x, t) = \epsilon \text{ for } x = 0$$

$$f(x, t) = \frac{x}{T(x)-x} \text{ for } x < 0$$

where  $T(x)$  is the tuning parameter defined by

$$T(x) = \kappa \max(w_i(t))$$

where  $\kappa$  is the sigmoid gain parameter.

We incorporate both the performance objective, the initial state and performance history into the function to generate feedback:

$r = 0$  if the system neither fails a performance objective nor falls into a failure state;

$r = -\frac{1}{|a-b|} (\frac{1}{N} \sum_{k=1}^N |E(T)|)$  where  $a$  is the initial temperature which is the initial state for the process,  $b$  is the performance objective overshoot requirement and  $E(T)$  is the average learning period;

$r = -1$  if  $c \notin [\min\{i, s-o\}, s+o]$  for  $i < s$  or  $c \notin [\max\{i, s+o\}, s-o]$  for  $i > s$ .  $c, i, s, o$  represent the current state, initial state, set point and overshoot limit respectively.

Y. Y. Chen [3] used a method similar to this in form but the interpretation of  $a$  and  $b$  is different.

In the present research, the following parameter values were used:  $\alpha = 0.05, \beta = 0.5, \delta = 0.93, \epsilon = 0.01, \gamma = 0.95, \kappa = 0.25, \lambda = 0.9,$

For more detailed information on the derivation of these learning rules, see Barto[2], Barto[1] and Lee[5].

## 4.2 Simulation Results

The work presented in this paper is the continuation of a previous project for developing a self-tuning fuzzy controller [4]. We simulated our system on a IBM PS/2 and compared its performance with that of three other control strategies: PID, the previously developed self-tuning fuzzy controller and a bare-bones fuzzy controller (without any learning or tuning). The SFLC is trained for 200 time steps, a set point of 200, and feed rate of 10 gallons per minute.

The general performance of the four regimes is shown in Figure 3. This is the performance of the controllers without any variation in operating parameters. We can see the self-learning scheme shows a performance very similar to that of the PID and the self-tuning scheme has a faster reaching time but slightly more overshoot.

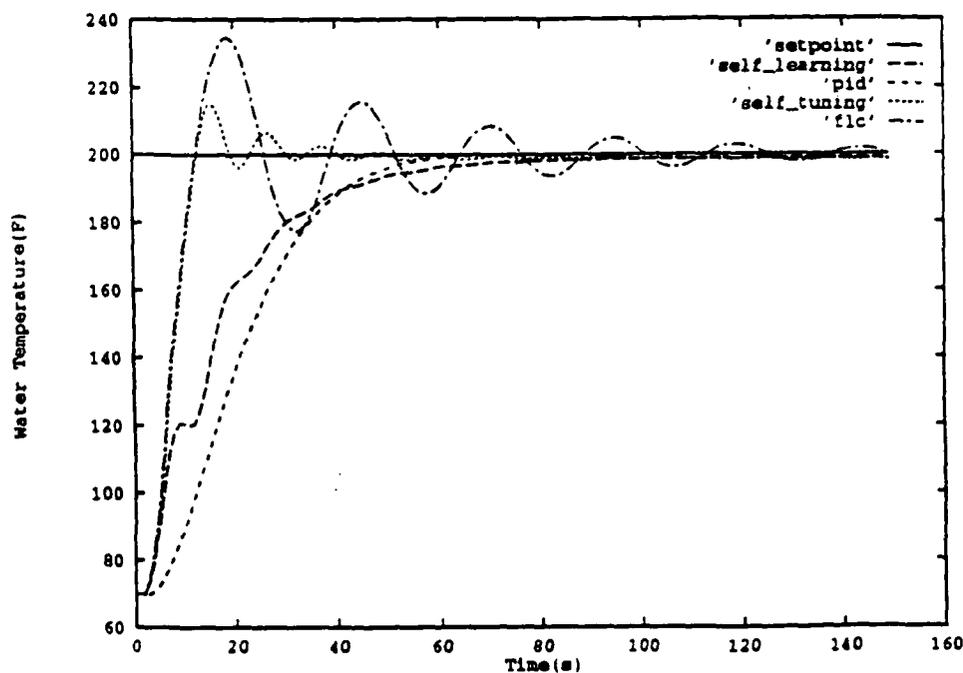


Figure 3: Four different control schemes

The advantages of the self-learning controller over the other schemes are demonstrated in three aspects. First is the short learning cycle. Four learning trials were sufficient for the system to reach the required performance level. Second, the stability of performance. Figure 4 shows the effects of changing the feed rate on the overshoot for PID and self-tuned systems. The SFLC has very little overshoot while varying the feed rate from 2 to 20 gallons per minute.

The third advantage is the wide range of operating conditions. When we varied the feed rate from 2 to 20 gallons per minute, the SLFC can perform as well as under normal conditions with very little fluctuation in performance and more importantly, without any re-training. Figure 5 shows the number of retraining steps needed for various feed rates.

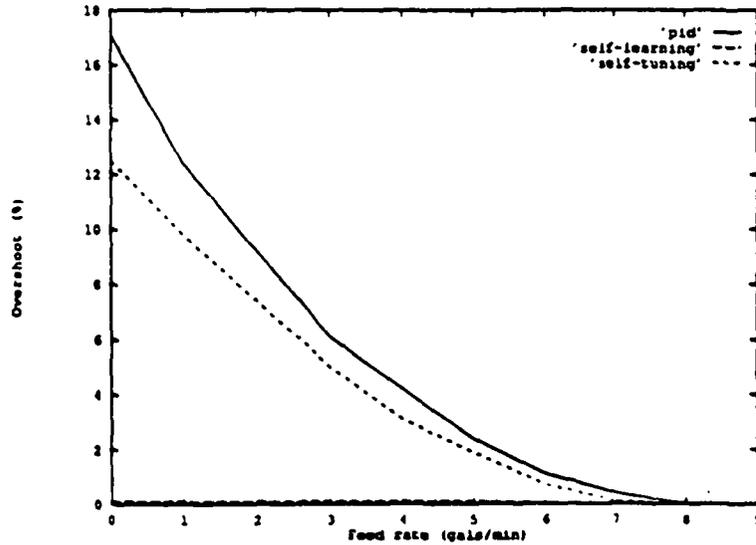


Figure 4: Effect of change of feed rate on overshoot

When the feed rate changes to 25 gallons per minute, the number of training steps needed increased only slightly.

## 5 Summary

In summary, we have discussed some of the design issues for a reinforcement-based self-learning fuzzy controller for application to a petrochemical process based on the approaches proposed by Barto and Lee. The main issues were the choice of training cases and the choice of feedback. Simulation results show that it has some advantages as discussed above over other schemes and that the choice of training cases and feedback has direct impact on the performance of a SFLC. Some issues such as finding a systematic way to determine the parameter values will be considered in future research.

## 6 Acknowledgement

Balaji Rathakrishnan was involved in the early stage of this project. His contribution is greatly appreciated and acknowledged here. Our research also benefited from discussion with C. C. Lee (Sony). Texaco provided the simulator of the water heating system used in this research.

feed rate	self-tuning FLC	self-learning FLC
2.0	2	0
5.0	1	0
8.0	0	0
15.0	1	0
20.0	1	0
25.0	1	2
30.0	1	2

Figure 5: Tuning/learning steps needed for new feed rates

## References

- [1] A. G. Barto and R. S. Sutton. Associative search network: a reinforcement learning associative memory. *Biol. Cybern.*, 40:213-246, 1981.
- [2] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transaction on Systems, Man and Cybernetics*, 13:834-846, 1983.
- [3] Y. Y. Chen, K. Z. Lin, and S. T. Hsu. A self-learning fuzzy controller. In *IEEE International Conference on Fuzzy Systems*, pages 189-196, March 1992.
- [4] Walter Daugherty, Balaji Rathakrishnan, and John Yen. Performance evaluation of a self-tuning fuzzy controller. In *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZY-IEEE)*, San Diego, California, March 1992.
- [5] C. C. Lee. A self-learning rule-based controller employing approximate reasoning and neural net concepts. *International Journal of Intelligent Systems*, 6:71-93, 1991.
- [6] E. H. Mamdani. Advances in the linguistic synthesis of fuzzy controllers. *Int. J. Man Mach. Studies*, 8(6):669-678, March 1976.
- [7] Lotfi A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3(1):28-44, January 1973.

518-63

161 818

N 93 - 29534

**Learning Characteristics of a Space-Time Neural Network  
As a Tether "Skiprope Observer"**

Robert N. Lea and James A. Villarreal  
NASA / Lyndon B. Johnson Space Center  
Houston, Texas 77058

Yashvant Jani  
Togai InfraLogic Inc.  
Houston, Texas 77058

Charles Copeland  
Loral Space Information Systems  
Houston, Texas 77058

**Abstract :**

The Software Technology Laboratory at the Johnson Space Center is testing a Space Time Neural Network (STNN) for observing tether oscillations present during retrieval of a tethered satellite. Proper identification of tether oscillations, known as "skiprope" motion, is vital to safe retrieval of the tethered satellite. Our studies indicate that STNN has certain learning characteristics that must be understood properly to utilize this type of neural network for the tethered satellite problem. We present our findings on the learning characteristics including a learning rate versus momentum performance table.

**1.0 Introduction**

NASA and the Italian Space Agency plan to fly the Tethered Satellite System (TSS) aboard the Space Shuttle in July, 1992. The mission, lasting approximately 40 hours, will deploy a 500 kg satellite upward (away from the earth) [1, 2] to a length of 20 km, perform scientific experiments while on-station, and retrieve the satellite safely. Throughout the deployment, experimentation, and retrieval, the satellite will remain attached to the Orbiter by a thin tether through which current passes, providing power to experiments on-board the satellite. In addition to the scientific experiments on-board the satellite, the dynamics of the TSS itself will be studied. The TSS dynamics are complex and non-linear due to the mass as well as spring-like characteristics of the tether. When the tether is modeled as a massless spring, it typically exhibits longitudinal and librational oscillations [2]. However, when the tether is modeled as beads connected via springs as shown in fig. 1, the dynamics of TSS includes longitudinal, librational and transverse circular oscillations or so-called "skiprope" phenomenon. These circular oscillations are generally induced when current pulsing through the tether interacts with the Earth's magnetic field [3, 4]. The center bead typically displaces the most from the center line. Thus, the "skiprope" can be viewed (fig. 2) by plotting a trajectory of the mid-point of the tether as it is retrieved slowly from the Onstation-2 phase in a high fidelity simulation test case. Detection and control of the various tether modes, including the 'skiprope' effect, is essential for a successful mission. Since there are no sensors that can directly provide a measure of skiprope oscillations, indirect methods like Time Domain Skiprope Observer [4] and Frequency Domain Skiprope Observer [3] are being developed for the mission. We are investigating a Space Time Neural Network (STNN) based skiprope observer.

The STNN is basically an extension to a standard backpropagation network [5,6,7] in which the single interconnection weight between two processing elements is replaced with a number of Finite Impulse Response (FIR) filters [8]. The use of adaptable, adjustable filters as interconnection weights provides a distributed temporal memory that facilitates the recognition of temporal sequences inherent in a complex dynamic system such as the TSS. We have performed experiments in detecting various parameters of skiprope motion using an STNN.

In Bead Model, the Tether mass is distributed in form of beads connected by springs.

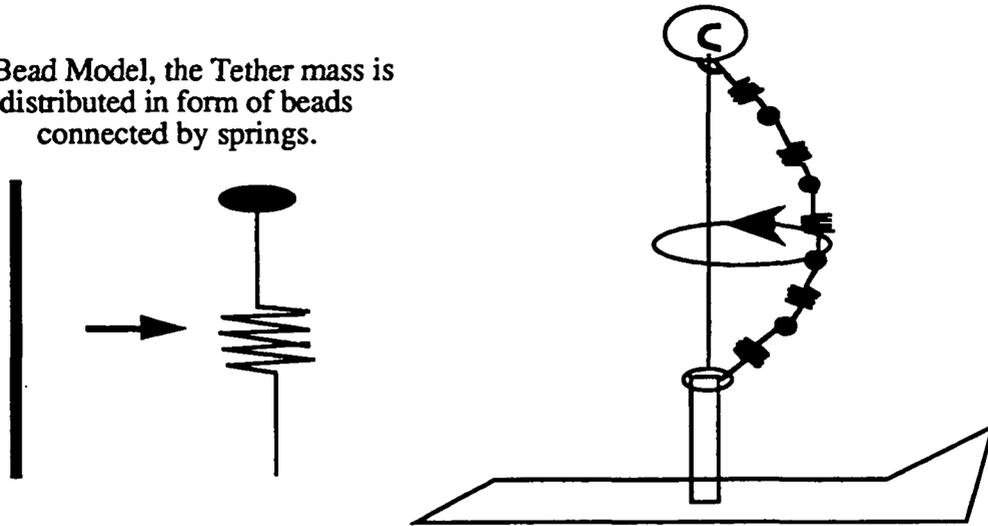


Fig. 1 When tether is modelled as beads, the transverse circular oscillations known as "skiprope" are induced during retrieval.

Extensive studies using high fidelity simulations have shown that the tethered satellite exhibits characteristic rate oscillations in the presence of skiprope motion as shown in figure 3. Since these rate oscillations are measured by the satellite's on-board rate gyros, the measured rates can be used as inputs to a skiprope detection system along with other measured parameters such as tension and length [9]. We have trained an STNN using data logged from a high fidelity Orbital Operations Simulator (OOS) [10] which models the behavior of the TSS. The parameters used in network training include satellite roll, pitch, and yaw rates, sensed tension and length of the tether, and the position of the mid-point of the tether during skiprope motion. In this paper, we first describe the STNN architecture in section 2. The STNN configuration used for skiprope observation is described in section 3 along with training and test data generated by the simulation test cases. Learning characteristics are discussed in section 4, and conclusions are summarized in section 5.

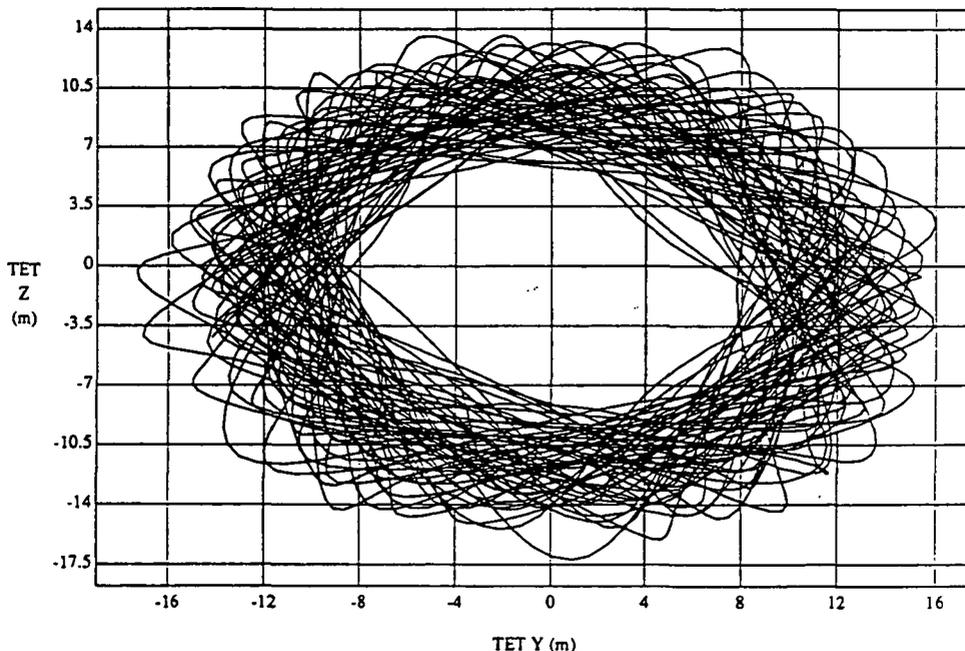


Figure 2 - Trajectory of tether mid-point during "skiprope".

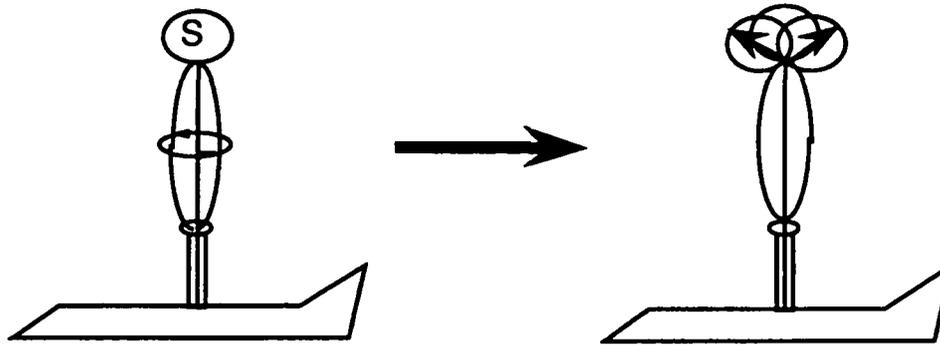


Figure 3 - Tether "skip rope" effect leads to highly characteristic satellite attitude oscillations which can be used to detect the magnitude and phase of the skip rope

## 2.0 STNN Architecture

The STNN architecture [8] allows the dimension of time to be added to the strong spatial modelling capabilities found in neural networks. The time dimension can be added to the standard processing element used in conventional neural networks by replacing the synaptic weights between two processing elements with an adaptable-adjustable filter as shown in figure 4.

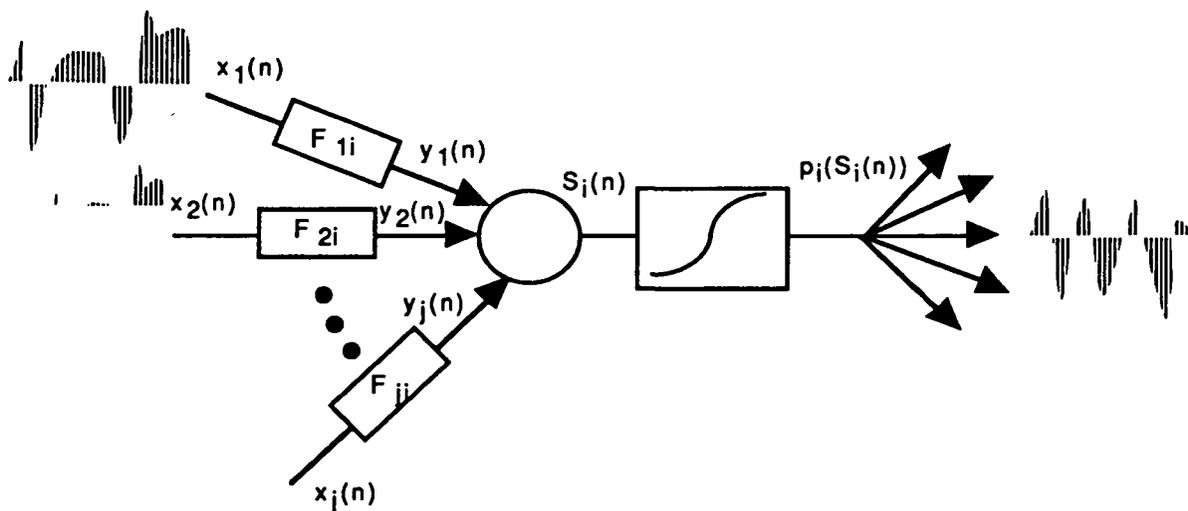


Figure 4 - A pictorial representation of the Space-Time processing element.

Instead of a single synaptic weight with which the standard backpropagation neural network represented the association between two individual processing elements, there are now several weights representing not only spatial association, but also temporal dependencies. In this case, the synaptic weights are the coefficients to the adaptable digital filters:

$$y(n) = \sum_{k=0}^N b_k x(n-k) + \sum_{m=1}^M a_m y(n-m) \quad (1)$$

Here the  $x$  and  $y$  time sampled sequences are the input and output respectively of the filter and  $a_m$ 's and  $b_k$ 's are the coefficients of the filter. Thus, if there are  $j$  parameters going into a neuron, the  $y_j$

are input into the neuron, where each  $y_j$  is a filtered value of the  $x_j$  using  $n$  time series samples as shown in fig. 4. The  $x_j$ 's are the real input from an external source. Thus, the STNN is learning a temporal dependency of the input parameters.

A space-time neural network includes at least two layers of filter elements fully interconnected and buffered by sigmoid transfer nodes at the intermediate and output layers as shown in figure 5. A sigmoid transfer function is not used at the input. Forward propagation involves presenting a separate sequence dependent vector to each input, propagating those signals throughout the intermediate layers until the signal reaches the output processing elements. In adjusting the weighting structure to minimize the error for static networks, such as the standard backpropagation, the solution is straightforward. However, adjusting the weighting structure in a space-time network is more complex because not only must present contributions be accounted for but contributions from past history must also be considered. Therefore, the problem is that of specifying the appropriate error signal at each time and thereby the appropriate weight adjustment of each coefficient governing past histories to influence the present set of responses. A detailed discussion of the algorithm can be found in the reference [8].

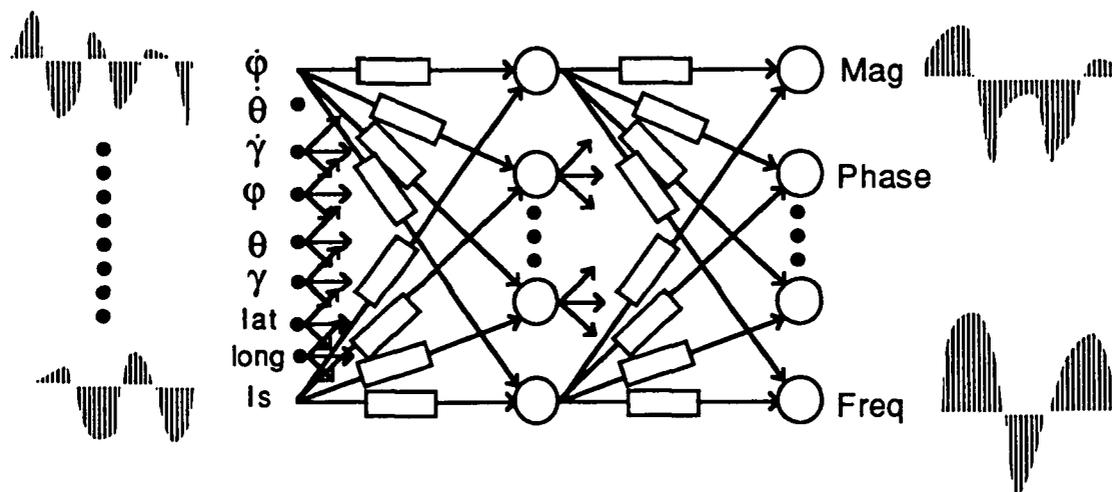


Figure 5 - A depiction of a STNN architecture showing the distribution of complex signals in the input space.

### 3.0 STNN Configuration and Test/Training Data

Several different simulation runs were used to gather data for STNN training. The simulation runs are consistent with the requirement that the skiprope observer must be capable of performing during various combinations of current flow through the tether and satellite spin. For example, one simulation represents a case in which current flows through the tether continuously, and the satellite is in yaw hold. Another simulation represents the case in which current flows through the tether only during the on-station phase, and the satellite is in yaw hold. A third simulation represents continuous current flow, and satellite spin at 4.2 degrees/second. These three scenarios will form the basis for STNN skiprope observer training and testing, and are consistent with simulations that are used for testing the Time-Domain Skiprope Observer (TDSO) [4] which will be flown on TSS-1.

Ultimately, the network should utilize only roll rate, pitch rate, yaw rate, sensed tension and sensed length since these are the only directly measurable parameters. However, we have

conducted experiments using derived parameters such as roll, pitch, and yaw position in addition to rates with no significant improvement. The biggest challenge to network training so far has been to learn the phase mapping. Several different network configurations have yielded good results in predicting skiprope amplitude, but we have not been as lucky with skiprope phase. Since the ultimate goal is to provide the crew with accurate measurements of skiprope amplitude and phase to support the yaw maneuver, the skiprope observer should learn to predict amplitude and phase based on the available inputs. However, decisions concerning the yaw maneuver can be based on the x and y coordinates of the mid-point of the skiprope motion as well. Therefore, the basic network configuration consists of 6 inputs (roll rate, pitch rate, yaw rate, sensed tension,  $x(t)$ , and  $y(t)$ ) and 2 outputs ( $x(t+1)$  and  $y(t+1)$ ). Notice that we are training on the current x and y position and predicting x and y position for the next time step. In previous experiments we focussed on finding the optimum network configuration in terms of numbers of hidden units and numbers of zeros from layer to layer. Through experimentation, we settled on 30 hidden units and 30 zeros from the input layer to the hidden layer, and 30 zeros from the hidden layer to the output layer, although slight deviations in these parameters have little or no effect in network performance. In this paper we concentrate primarily on the effects of learning rate and momentum on the overall generalization of the Space-Time Neural Network.

#### 4.0 Learning Characteristics

A well known characteristic of backpropagation networks, or networks derived from backpropagation, is that in order to achieve reasonable generalization, the network must learn the training data. Experiments have indicated that, like standard backpropagation, the learning characteristics of STNN are such that if the training data is not learned, generalization will not occur. These and other learning characteristics dictate that a particular sequence of steps be followed in the training and testing of STNN. The following general steps were used as guidelines throughout the STNN testing. Please note that the use of the word "momentum" in this report refers to a term in the learning algorithm that represents a fraction of the previous weight change rather than any physical properties of the TSS.

1. Train and test - evaluate learnability of training data.
2. Adjust network as necessary (set learning rate and momentum in updating of interconnection weights).
3. If network is unable to obtain sufficient convergence on training data, test individual parameters one at a time. Eliminate un-mappable parameters and start over.
4. If reasonable convergence is realized on training data, divide the data set into a training set and a separate test set.
5. When reasonable performance is achieved on the separate test data, then go for multi-test case generalization.

Step 2 above generally involves trying different combinations of learning rate and momentum in the interconnection weight update formulas. Table 1 illustrates the test case matrix we have identified in order to test the effects of different combinations of learning rate and momentum.

The results that follow are from training and testing using data from the simulation which includes current pulsing and satellite spin, which is considered the most difficult case. Following our general training and testing steps listed above, we verified that the STNN was able to learn the training data using a learning rate of 0.05, and momentum set to 0.9. We trained and tested on all 3500 Input/Output pairs and achieved a MAX error of 0.08 and RMS error of 0.02 at 140 cycles. Since the network will be trained off-line before being placed in the operational environment, we must determine how well the network will perform when presented with data that it has not previously seen. Therefore, to test the generalization ability of STNN, we train on only the first and last 400 input/output pairs from the full 3500, and test separately on the middle 2700

input/output pairs while trying various combinations of learning rate and momentum with the following results. First, with a momentum of 0.9, we tried learning rates of 0.05, 0.2, and 0.7 (test cases #4-6 in Table 1). Test case #4 resulted in MAX error = 0.43, and RMS error = 0.04 at cycle 100. Figure 6 shows the error plot for test case #4 up to 500 cycles. Figures 7a and 7b show a portion of the x and y predictions from test case #4. Test case #5 resulted in MAX error = 0.43 and RMS error = 0.04 at cycle 480. Figure 8 shows that the network prediction of y in test case 5 is similar to that of test case #4. Increasing the learning rate to 0.7 in test case #6 results in the network never reaching errors as low as in the previous two test cases (at least not within 500 cycles) and overall performance is similarly degraded as is seen in figures 9a and 9b. Next we set momentum to 0.2 and try learning rates of 0.05, 0.2, and 0.7 (test cases #1-3 in Table 1). Test case #1 yielded MAX error = 0.44, and RMS error = 0.05 at 100 cycles, as is shown in figure 10a. Figure 10b shows that the network's prediction of x in this test case is not quite as accurate as test cases #4 and #5. As we increase learning rate from 0.05 to 0.2, performance degrades significantly as is shown in figure 11a. The error graph in figure 11b shows that no learning occurred in test case #2, as RMS error never dropped significantly below 0.5, and MAX error remained near 0.8. Similar results occurred in test case #3 as we increased the learning rate from 0.2 to 0.7. The overall test errors are summarized in Table II.

**Table 1 - Learning Rate Versus Momentum in STNN Weight Update Formulas**

Test Case	Momentum in weight update	Learning Rate
1	0.2	0.05
2	0.2	0.2
3	0.2	0.7
4	0.9	0.05
5	0.9	0.2
6	0.9	0.7
7	0.95	0.05
8	0.98	0.05

**Table II - Number of Training Cycles to Reach Lowest Test Errors.**

Test Case	Max Error	RMS Error	Number of Cycles
1	0.44	0.05	100
2	0.78	0.49	280
3	0.8	0.5	480
4	0.43	0.04	100
5	0.43	0.04	480
6	0.5	0.09	400
7	0.41	0.04	480
8	0.41	0.04	480

## 5.0 Conclusions

Through experimentation, we have gained insight into the learning characteristics of STNN in terms of learning rate and momentum parameters. In particular, we find that the skiprope observer problem requires high momentum and very low learning rate. In test case 4 we have seen that the RMS error drops to 4 % within only 100 cycles of learning. We further verified this by performing two test cases (#7 and #8) with high momentum and low learning rate. It should be noted that the max error is reduced in both cases.

Figure 6 - Test Case 4, Max VS RMS Error

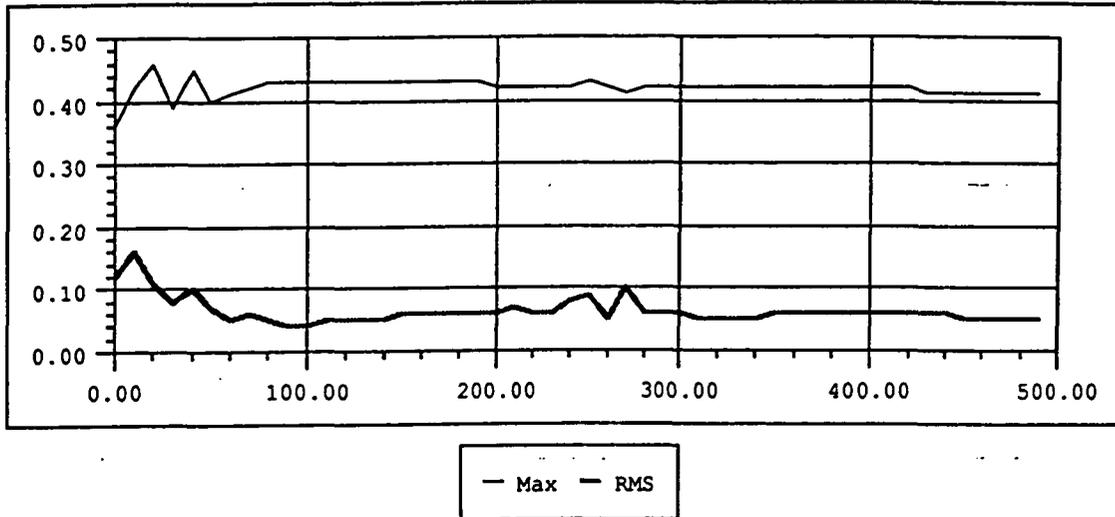


Figure 7a - Test Case 4, Target X VS STNN X, at 100 cycles

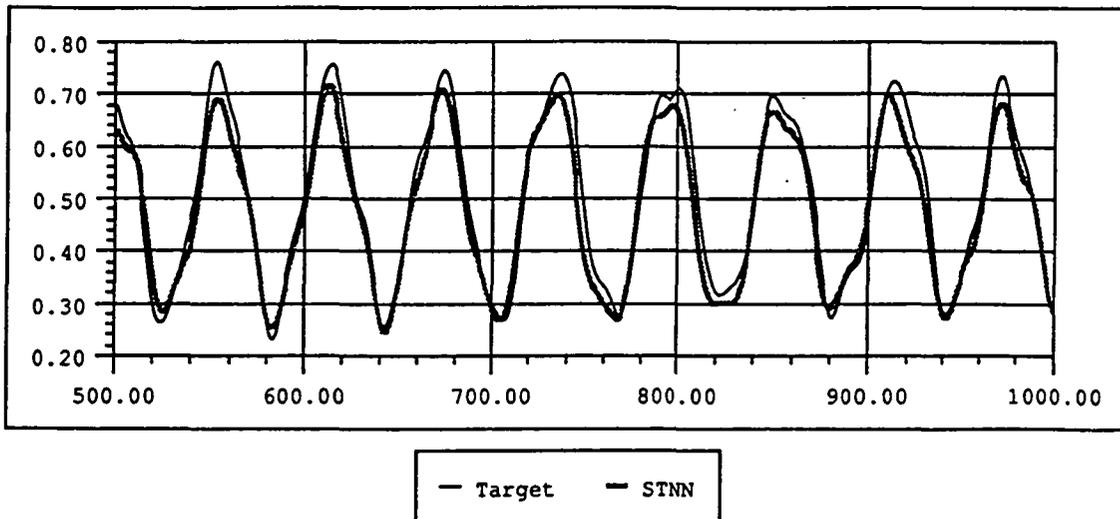


Figure 7b - Test Case 4, Target Y VS STNN Y, at 100 cycles

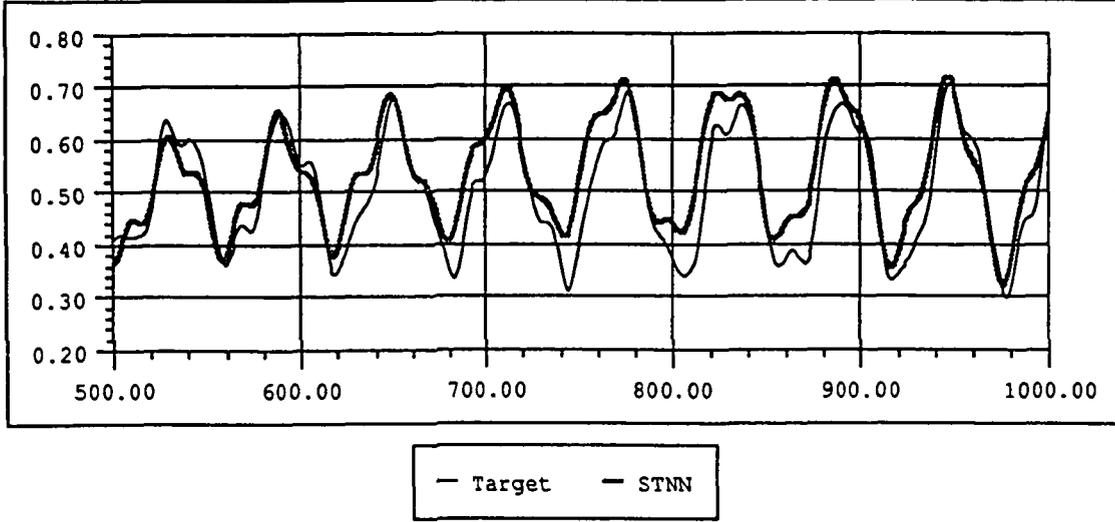


Figure 8 - Test Case 5, Target Y VS STNN Y.

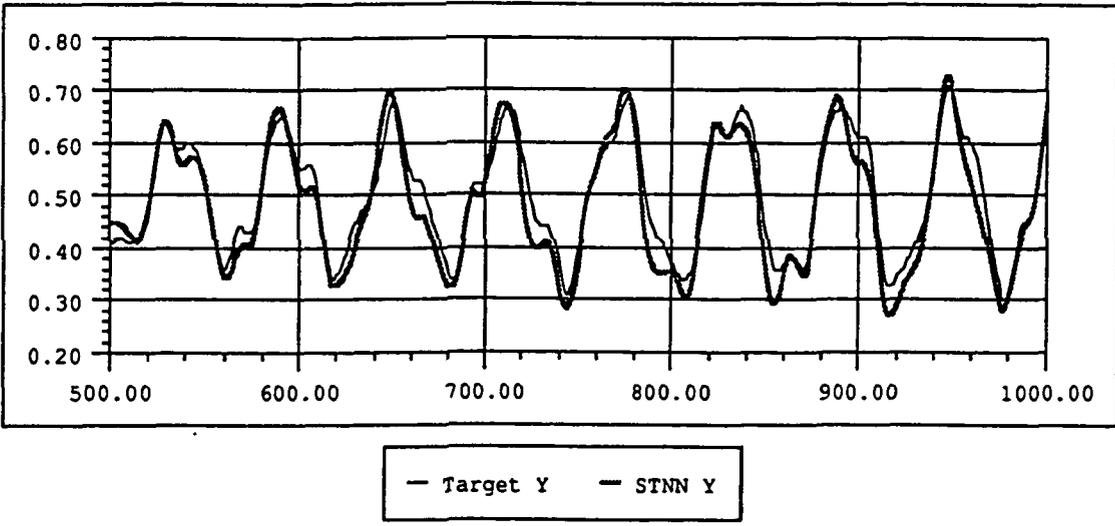


Figure 9a - Test Case 6, Max VS RMS Error

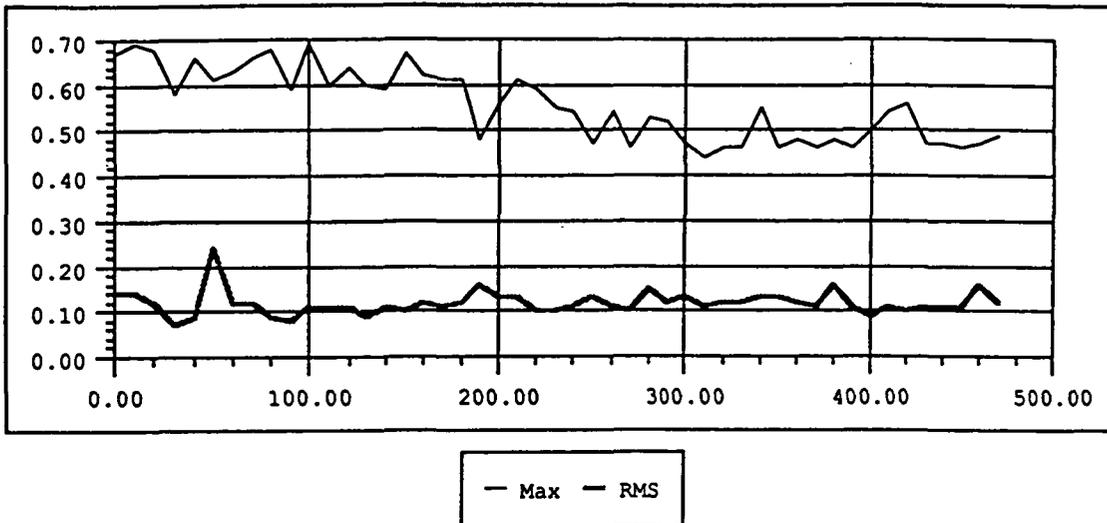
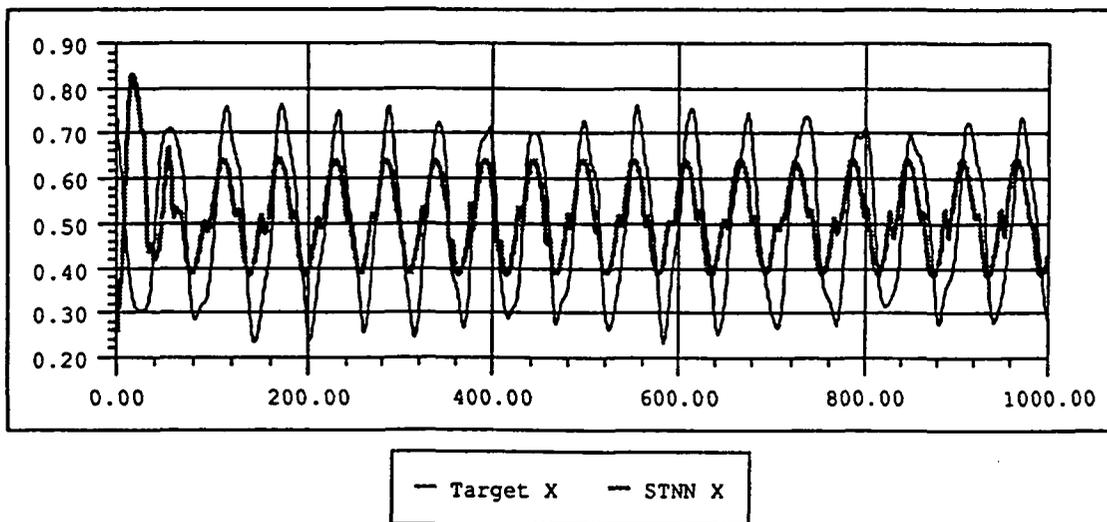


Figure 9b - Test Case 6, Target X VS STNN X, 400 cycles



C-3

Figure 10a - Test Case 1, Max VS RMS Error

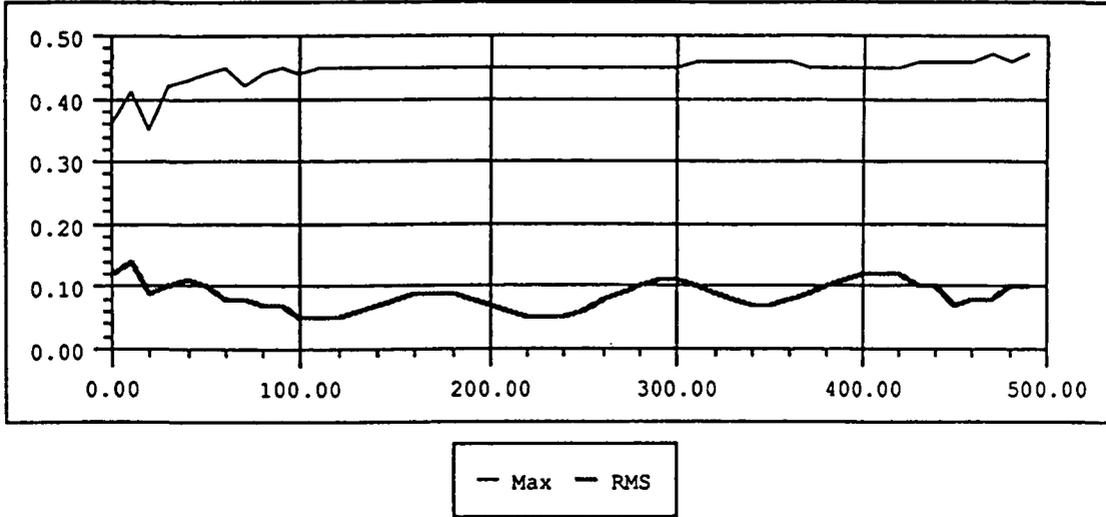


Figure 10b - Test Case 1, Target X VS STNN X, 100 cycles.

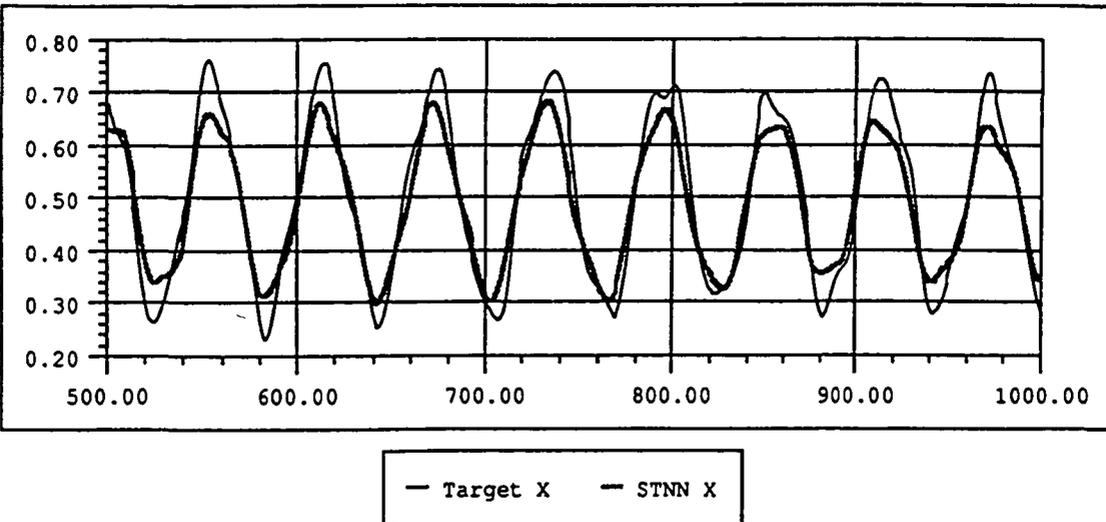
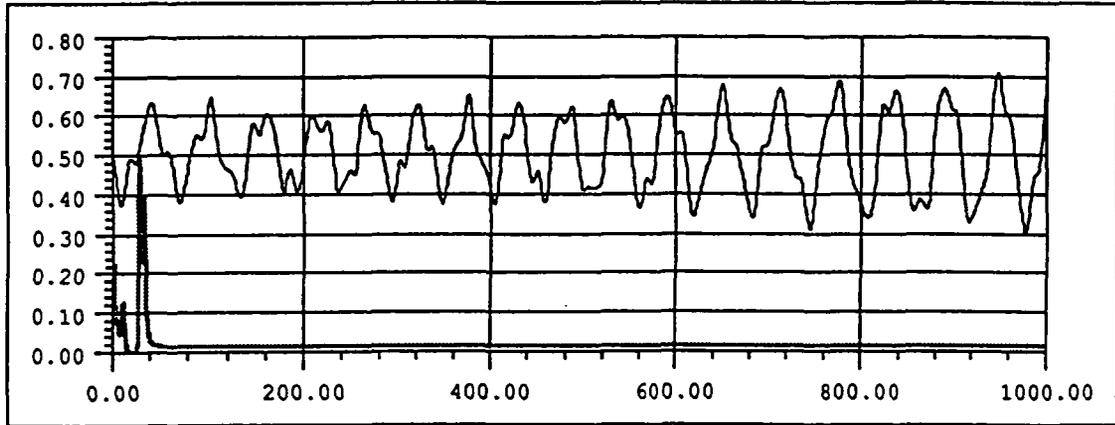
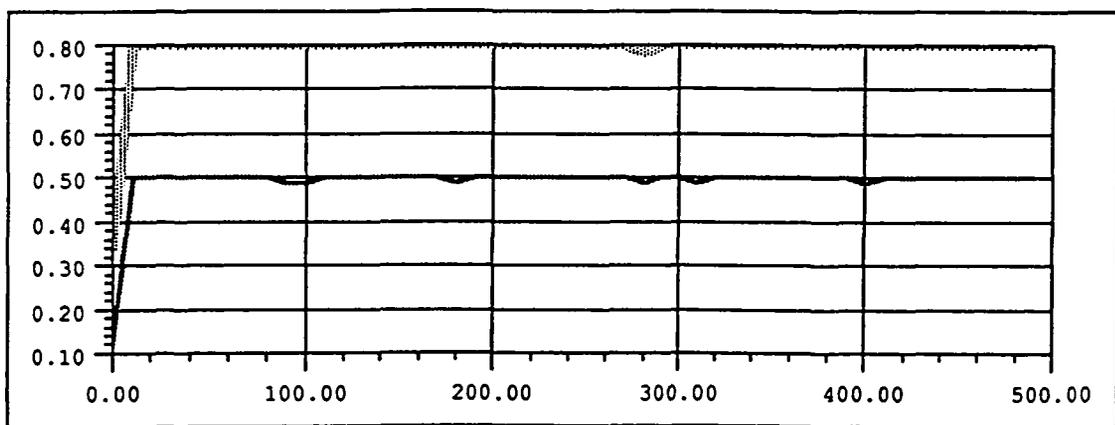


Figure 11a - Test Case 2, Target Y VS STNN Y, 280 cycles.



— Target Y    - - - STNN Y

Figure 11b - Test Case 2, Max VS RMS Error



..... Max    — RMS

Based on our earlier results, we conclude that the STNN is slow in learning sharp discontinuities like those encountered in phase behavior. The value of the phase goes from 180 to -180 abruptly when the circle is complete. When we changed to the x- and y- component form (rather than amplitude and phase), the STNN based skiprope observer performed much better in predicting x and y coordinates of the mid-point of the tether.

We will have an opportunity to perform a side-by-side comparison of the STNN based skiprope observer and the TDSO using simulation data. Next, we will test the STNN based skiprope observer with the post mission data after the TSS-1 flight.

#### References :

1. Coledan, S. : "Tether Satellite Advances", Space News, vol. 2, no. 15, p. 8, 1991.
2. Powers, C.B., Shea, C., and McMahan, T. : "The First Mission of the Tethered Satellite System", A special brochure developed by the Tethered Satellite System Project Office, NASA/Marshall Space Flight Center, Huntsville, Alabama, U.S. GPO 1992-324-999, 1992.
3. Ioup, G.E., Ioup, J.W., Rodrigue, S.M., Amini, A.M., Rayborn, G.H., and Carroll, S. : "Frequency Domain Skiprope Observer", Skiprope Containment Status Meeting held at Denver, Sep. 10-11, 1991. (Research supported by NASA Contract NA8-38841)
4. Glowczwski, R. : " Time Domain Skiprope Observer Overview", Skiprope Containment Status Meeting held at Martin Marietta, Denver, Sept. 10-11, 1991.
5. Kosko, B. : "Neural Networks and Fuzzy Systems", Prentice-Hall, New Jersey, 1992.
6. J.A.Freeman and D.M.Skapura : "Neural Networks Algorithms, Applications and Programming Techniques", Addison-Wesley Publishing Company, Reading, MA. 1991.
7. Wasserman, P.D. : "Neural Computing Theory and Practice" Van Nostrand Reinhold, New York, 1989.
8. Villarreal, J.A., and Shelton, R.O. : "A Space-Time Neural Network", International Journal of Approximate Reasoning , 6(2), 133-149, 1992.
9. Lea, R.N., Villarreal, J.A., Jani, Y., and Copeland, C. : "Application of Space Time Neural Networks to detect Tether Skiprope Phenomenon in Space Operations", Proceedings of the AIAA GN&C Conference held at Hilton Head, South Carolina, August 10-12, 1992.
10. Edwards, H. C., and Bailey, R. : "The Orbital Operations Simulator User's Guide", LinCom corporation, ref. LM85-1001-01, June 87.

519-67  
161819

N93-29535

P-9

Clustering of Tethered Satellite System Simulation  
Data by an Adaptive Neuro-Fuzzy Algorithm

Sunanda Mitra and Surya Pemmaraju  
Department of Electrical Engineering  
Texas Tech University  
Lubbock, TX 79409-3102

**ABSTRACT**

Recent developments in neuro-fuzzy systems indicate that the concepts of adaptive pattern recognition when used to identify appropriate control actions corresponding to clusters of patterns representing system states in dynamic nonlinear control systems, may result in innovative designs. A modular, unsupervised neural network architecture, in which fuzzy learning rules have been embedded is used for on-line identification of similar states. The architecture and control rules involved in Adaptive Fuzzy Leader Clustering (AFLC) allow this system to be incorporated in control systems for identification of system states corresponding to specific control actions. We have used this algorithm to cluster the simulation data of Tethered Satellite System (TSS) to estimate the range of delta voltages necessary to maintain the desired length and length rate of the tether. The AFLC algorithm is capable of on-line estimation of the appropriate control voltages from the corresponding length error and length rate error without a priori knowledge of their membership functions and familiarity with the behavior of the Tethered Satellite System.

**I. INTRODUCTION**

In spite of recent developments in nonlinear dynamical systems modeling, analytical as well as implementation difficulties still remain in many controller design problems [1]. Integration of fuzzy learning rules with neural networks may provide flexibility in designing models for these systems [2]. In supervised learning, a set of correct control actions can be learned and used to estimate other actions required in a dynamic control system whereas unsupervised learning may suggest appropriate control actions corresponding to system states forming a pattern cluster. Applications of tethers in space have demonstrated scope for control using the latter technique. Due to the elasticity and finite mass distribution of the tether, any tethered system has complex, nonlinear dynamics. As a result, control of these systems is not easily achieved. Fuzzy logic based controllers [3] have handled nonlinearities of such a system quite well with no requirement to fully understand the dynamics of the system. Normally, a fuzzy controller defines some linguistic variables and generates fuzzy membership functions and a rulebase for the controlling parameters, using some a priori information regarding the system. Instead, we have used a hybrid neural-fuzzy clustering algorithm namely, Adaptive Fuzzy Leader Clustering (AFLC) [4] to find optimal control actions. This clustering algorithm can be used for optimal clustering in many pattern recognition problems [4] as well as for examining control actions required for complex systems with nonlinear dynamics.

Cluster analysis has been a significant research area in pattern recognition for a number of years [5]-[6]. Despite significant improvements in clustering of specific data sets by incorporating fuzzy membership concepts into hard clustering [7]-[9], partitioning real and noisy data sets still poses difficulties, thus keeping this research area wide open. Integration of fuzzy clustering concepts with neural network architectures may provide further flexibility in identification of appropriate data clusters. One such attempt is presented here by using the AFLC algorithm to cluster the simulation data of the Tethered Satellite System (TSS). AFLC has an unsupervised neural network architecture developed from the concepts of ART-1 [10]-[11], and uses the set of nonlinear equations for centroid and membership values as developed in the fuzzy C means (FCM) algorithm [12] for updating the centroid locations. AFLC learns on-line in a stable and efficient manner and adaptively clusters input discrete or analog signals into classes without a priori knowledge of the input data structure. Each resultant output cluster has a prototype which represents all the data samples in that cluster. In this paper, we apply the AFLC algorithm to effectively cluster the simulation data of the Tethered Satellite system, containing the crucial parameters for controlling the system behavior. This paper is organized as follows. Section II gives a brief description of the AFLC system and algorithm. Section III outlines the behavior of a Tethered Satellite System and suggests a possible architecture for controlling it. Section IV presents the test results of AFLC operation on the TSS data set used. Finally, Section V addresses the potential applications of AFLC in recognition and control of complex data sets and systems respectively.

## II. ADAPTIVE FUZZY LEADER CLUSTERING

### A. The AFLC algorithm overview

AFLC is primarily used as a classifier of feature vectors employing an on-line learning scheme [4]. The algorithm basically consists of three procedures, recognition, comparison and updating. It involves a two-stage classification which takes place in the recognition and the comparison stages. The system is initialized with the input of the first feature vector  $X_1$  and the number of clusters ( $C$ ) is set to zero. Similar to leader clustering, this first input forms the prototype for the first cluster. This cluster is represented by a node in the recognition layer of the AFLC system. Connection of any such node  $i$  to the input vector  $X_j$  in the comparison layer is established through a set of multiplicative weights referred to as the bottom-up weights ( $b_{ij}$ ), whose values correspond to a normalized version of the cluster prototype. Subsequent to this initialization, normal operation commences [4].

The normalized version of the next input vector is applied to the bottom-up weights of all the existing cluster nodes in a simple competitive learning scheme, or dot product. The activation level,  $Y$ , of node  $i$  in the recognition layer is

$$Y_i = \sum_{k=1}^p X_k b_{ik} \quad (1)$$

where  $p$  is the dimension of the input feature vector. The recognition stage winner is the node with the maximum value of  $Y$ . In the specific case of the second input vector, there is only one recognition layer node which was activated by the first input. This node will win the competition

by default, which would lead to a very disappointing performance. Additional processing, however is obtained, as in ART [11], by attempting to match the input to a top-down expectation. This takes place in the comparison stage. The Euclidean distance between the original input vector and the cluster prototype of the winner node is calculated. This value is then compared to the average distance from the centroid of all the samples belonging to that cluster. If this distance ratio  $R$  is less than a user-specified threshold  $\tau$ , then the input is found to belong to the cluster originally activated by the recognition layer. This relation can be represented as

$$R = \frac{\|x_j - v_i\|}{\frac{1}{N_i} \sum_{k=1}^{N_i} \|x_k - v_i\|} < \tau \quad (2)$$

where :  $j = 1 \dots N_i$ , is the number of samples in class  $i$  and  $v_i$  is the centroid of class  $i$ .  $\tau$  is called the vigilance parameter and determines the compactness within a cluster and the inter cluster separation. The choice of the value of  $\tau$  is critical in some applications where unlabelled data consisting of overlapping clusters is to be classified precisely. If the comparison of the input and the cluster prototype does not satisfy the threshold requirement, a search is implemented. This is accomplished by deactivating the currently activated recognition layer neuron with the help of the reset signal and repeating the classification process. If no cluster exists which meets the distance ratio criterion, then a new node is established.

When an input is classified to belong to an existing cluster, it is necessary to update the expectation (prototype) and the bottom-up weights associated with that cluster. This is done in the last stage using the fuzzy C means formulae. The cluster prototype or centroid is recalculated as a weighted average of all the elements within the cluster. The membership values  $\mu_{ij}$  of all the samples in the updated cluster with respect to the new centroid  $v_i$  are calculated. Since the membership values are dependent on the centroid positions, the relocation of the centroid in the winner cluster affects the membership values of the other data samples in the remaining clusters and hence they are recalculated. Equations 3 and 4 given below are the fuzzy C means [12] equations that have been employed for updating the cluster centroid and the membership values of the data samples. It is to be noted that equation 3 updates  $v_i$  only in the columns currently associated with class  $i$  whereas equation 4 involves a full membership updating process. The summation in equation 3 would extend from 1 to  $N$  in full FCM updating of the class prototypes. Here,  $N$  is the total number of data samples and  $m$  is the parameter which defines the fuzziness of the results and is normally set to be between 1.5 and 30. For the following application,  $m$  was experimentally set to a value of 2.

$$v_i = \frac{1}{\sum_{j=1}^{N_i} (\mu_{ij})^m} \sum_{j=1}^{N_i} (\mu_{ij})^m X_j \quad \text{for } 1 \leq i \leq C \quad (3)$$

$$\mu_{ij} = \frac{\left( \frac{1}{\|X_j - v_i\|^2} \right)^{1/(m-1)}}{\sum_{k=1}^C \left( \frac{1}{\|X_j - v_k\|^2} \right)^{1/(m-1)}} \quad \text{for } 1 \leq i \leq C \quad \text{and} \quad 1 \leq j \leq N \quad (4)$$

The updating process is followed by a verification procedure whose function is to check if the previous classification is still valid. The location of the samples which come and join the cluster at a later stage can often cause the prototype (centroid) of the cluster to shift in a particular direction. This depends on the order in which the data is fed to the algorithm. As a result, the distance of some of the data samples from the new centroid of the cluster to which they originally belonged, will increase drastically and hence the vigilance condition might not be satisfied any more. This could result in a misclassification if these samples are found to be closer to another neighboring cluster and satisfy the vigilance condition with respect to it.

The modified AFLC avoids this problem by means of a verification procedure which tests if all the samples in the updated cluster conform to the original classification. A sample that does not satisfy the original classification condition is reclassified by minimizing a simple error function given in equation 5. This error function helps in selecting the cluster which is closest to the input sample, by minimizing the weighted sum of the squares of the distances [12]. Therefore this verification process ensures that the algorithm is immune to the order of data presentation.

$$J(\mu, v) = \sum_{i=1}^C \sum_{j=1}^{N_i} \|X_j - v_i\|^2 (\mu_{ij})^m \quad (5)$$

### III. CLUSTERING OF TETHERED SATELLITE SYSTEM PARAMETERS

#### A. Behavioral Characteristics of TSS

The TSS consists of a reel powered by an electric motor, satellite thrusters, and the orbiter attitude control system [15]. Evaluation of the overall control of the TSS is done by means of tether length, tether tension, longitudinal and librational oscillations as shown in Figure 1. The elasticity of tether and the gravity gradient forces acting on the satellite result in longitudinal oscillations. The motion of the tethered satellite along the velocity vector, i.e., in a line from the nose to the tail of the orbiter causes in-plane libration and that towards the starboard side of the shuttle causes out-of-plane libration. Since it is only the tether length and tether tension that can be directly measured and controlled, the in-plane and out-of-plane

libration amplitude have to be indirectly controlled through tether length and length rate maintenance.

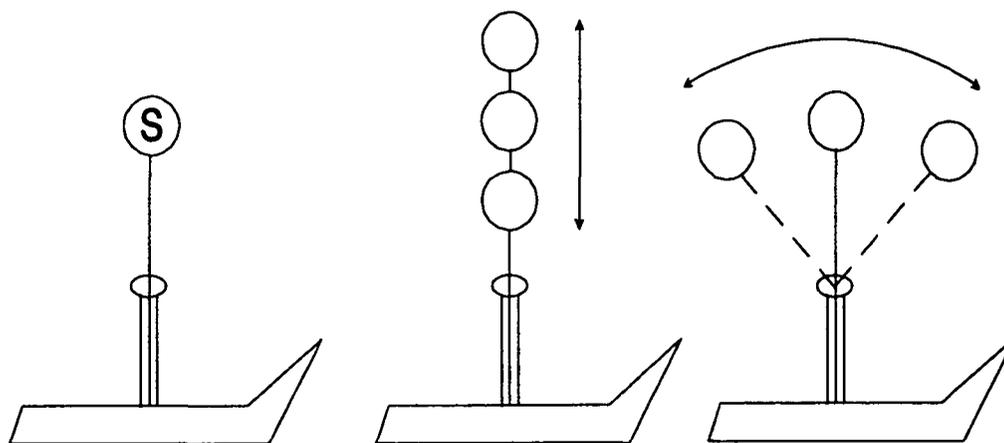


Figure 1. Longitudinal and Librational Oscillations in a Tethered payload System [15]

#### B. TSS Control Parameters

The parameters that are used to control the TSS are the Length Error, the Length Rate Error and the Delta Voltage. Using these three parameters, one can design a stable system by utilizing fuzzy membership functions for such parameters. However some familiarity with the TSS and its behavior is essential to estimate appropriate values for those membership functions [15]. When the TSS simulation data is fed to the AFLC system, it classifies the data into clusters depending on the value of the vigilance parameter. The output of the AFLC system is a classification of the input data into distinct clusters. Each cluster specifies the range of the length error and length rate error for a given value of delta voltage.

This performance is analogous to that of a rulebase describing the relation between the inputs and the output in terms of some linguistic variables. Membership functions for these linguistic variables are defined in terms of its range and its belief values using some intuitive knowledge of the physical system [15]. However, in our case, no a priori knowledge is required. The system being an unsupervised network, learns on-line from the data and classifies each data vector into the appropriate class depending on its past learning. The three TSS parameters can be considered to be the state variables of the system. Input data consists of the length error, the length rate error and the corresponding delta voltage values sampled at 50 seconds intervals.

Figure 2 shows the suggested schematic for an adaptive fuzzy control system using AFLC. Here the AFLC system combined with a functional link acts as a fuzzy controller. A look-up table and an estimator can form the basis for this functional link. The output of this controller is given as input feedback to the tethered satellite simulation system and the actual output of the physical system forms the next stage input to the AFLC algorithm.

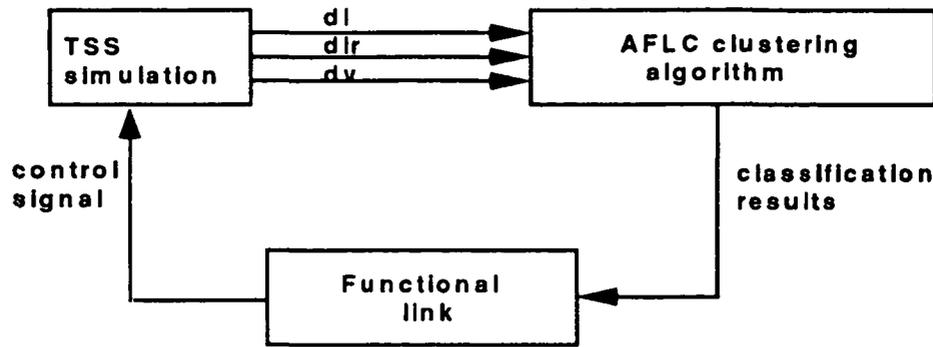


Figure 2. Schematic of a suggested fuzzy controller

#### IV. TESTS AND RESULTS

The data set consists of 1765 samples obtained from the massless tether model in the Tethered Satellite System simulation. This data has been collected at intervals of 50 seconds. Each input vector consists of length error ( $dl$ ), length rate error ( $dlr$ ) and the corresponding delta voltage ( $dv$ ). The entire data set has been classified using the AFLC algorithm. The value of  $\tau$  has been chosen to be 2.5. Figure 3 gives a table showing the classification results. It can be inferred from the table that the data set has been broadly classified into four categories. A few points that have not been classified as belonging to any of the four clusters can be treated as noise/outliers. Each cluster represents a specific range of input and output parameters.

CLASS	SAMPLES	RANGE OF $dl$	RANGE OF $dlr$	RANGE OF $dv$
1.	657	0.19 to 1.49	-0.135 and 0.0 to 0.0096	0.315 to 0.670
2.	528	1.96 to 2.63	-0.0304 to -0.000033 and 0.0 to 0.00786	1.77 to 2.85
3.	383	2.93 to 6.07	-0.0088 to -0.000009 and 0.00001 to 0.0087	3.00 to 4.29
4.	174	-5.2 to -1687.1	-0.765 to -0.01 and 0.01 to 0.354	-3.59 to -7.47

Figure 3. Adaptive Fuzzy Leader Clustering of TSS Data

The results from this table are comparable to those obtained from a rulebase, which specifies the output category for a given combination of input categories [15]. However, the

results obtained by our algorithm do not require a priori information of the system parameters. AFLC actually classifies the data structure using an on-line learning scheme. Hence this is a more realistic approach of solving the problem without requiring any intuitive knowledge of the system. Figure 4 shows a plot displaying the clusters in a two-dimensional feature space. Incorporation of these clustered delta voltages into the orbiter operations simulator (OOS) should provide smoother operational characteristics of the TS system.

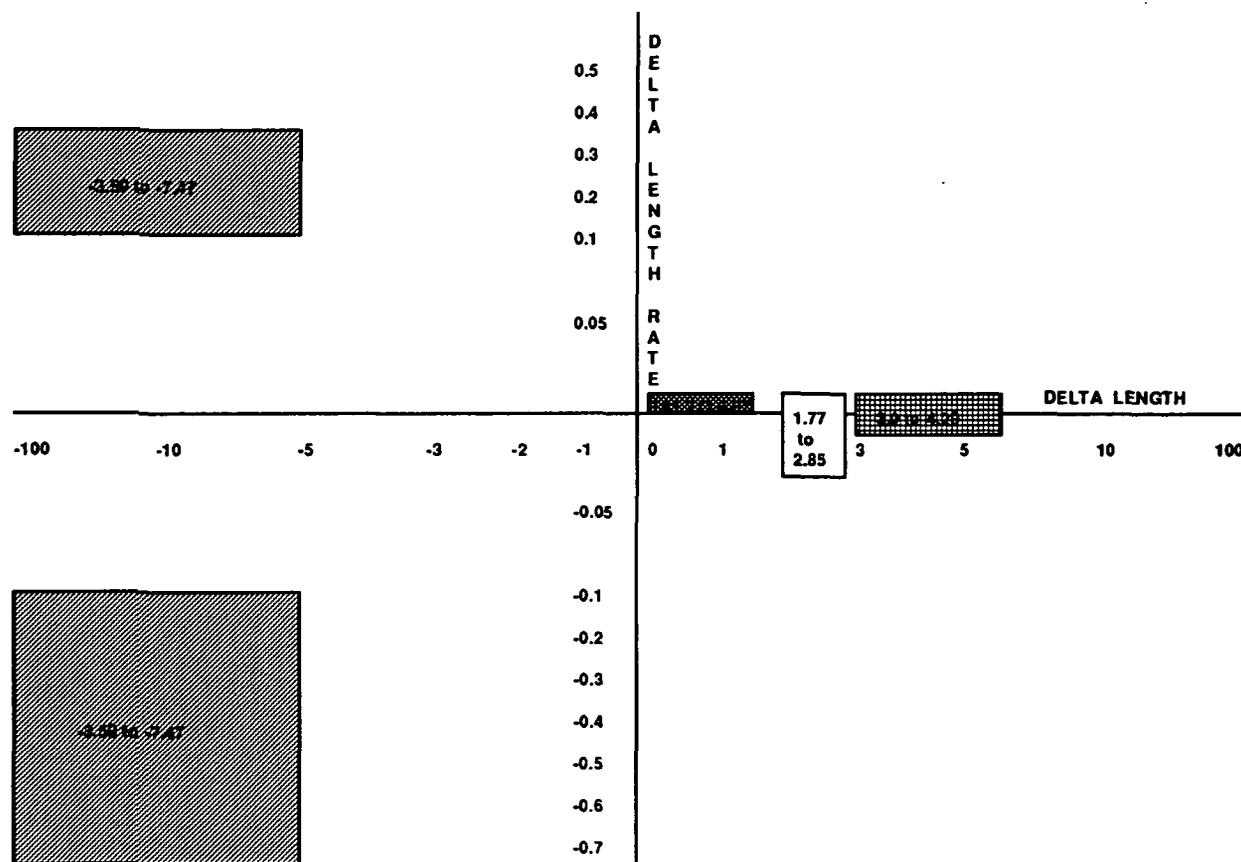


Figure 4. Control Delta Voltages corresponding to length error and length rate error

## V. CONCLUSION

This neuro-fuzzy algorithm, namely AFLC ensures stable, consistent learning of the membership of the new on-line inputs without a priori knowledge of the data structure. The flexibility in the algorithm makes it possible to apply many of the concepts of AFLC operation to typical control problems.

The use of AFLC to generate dynamic control actions corresponding to system state clusters of a nonlinear dynamic system, stems from similar concepts suggested by recent works using neural network for control [1],[2],[16]. Such fusion of adaptive pattern recognition and

control actions may result in innovative designs of dynamic nonlinear control systems and deserves further investigation. Better integration of fuzzy membership function with self-organizing neural network learning rules have been achieved recently [17],[18] demonstrating the applicability of neuro-fuzzy algorithms in complex decision making processes.

## VI. ACKNOWLEDGMENT

This work was supported in part by a grant from NASA-JSC under contract # NAG9-509, a grant from Northrop Corporation under contact # HHH7800809K and a grant from Texas Advanced Technology program under contract # 003644-047. The authors wish to acknowledge collaboration with R. Lea, Y. Jani and J. Villarreal of NASA-JSC and thank them for providing us with the TSS simulation data.

## VII. REFERENCES

- [1] W.T. Miller III, R.S. Sutton and P.J. Werbos, Neural Networks for Control, MIT Press, Cambridge, MA , 1990.
- [2] Y. H. Pao, Adaptive Pattern Recognition and Neural Networks, Addison-Wesley Reading, 1990.
- [3] R.N. Lea and Y. Jani, "Applications of Fuzzy Logic to Control and Decision making," Technology 2000 Proc., NASA Conference Publication 3109, Vol 2, pp. 67, 1990.
- [4] S.C. Newton, S. Pemmaraju and S. Mitra, "Adaptive Fuzzy Leader Clustering of Complex Data Sets in Pattern Recognition," IEEE Trans. on Neural Networks, Vol. 3, No. 5, Sep 1992.
- [5] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, New York, 1973.
- [6] A.K. Jain and R.C. Dubes, Algorithms for Clustering Data, Englewood Cliffs, NJ, Prentice-Hall, 1988.
- [7] L.A. Zadeh, "Fuzzy Sets," Information and Control, 8, pp 338-353, 1965.
- [8] J.C. Bezdek, "A Physical Interpretation of Fuzzy ISODATA," IEEE Trans. on Systems, Man and Cybernatics, Vol. 6, pp 387-389, 1976.
- [9] I. Gath and A.B. Geva, "Unsupervised Optimal Fuzzy Clustering," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 11, no. 7, pp 773-781, 1089.
- [10] G. A. Carpenter and S. Grossberg, "A massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine," Computer Vision, Graphics and Image Processing, 37, pp 54-115, 1987.
- [11] G.A. Carpenter and S. Grossberg, "ART-2 : Self Organization of Stable Category Recognition Codes for Analog Patterns," Applied Optics 2 , pp 4919-4938, 1987.
- [12] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Functions Algorithms, Plenum, New York, 1981.
- [13] S.C. Newton and S. Mitra, "Self-organizing Leader Clustering in a Neural Network Using a Fuzzy Learning Rule," SPIE Proc. on Adaptive Signal Processing, Vol. 1565, July 1991.
- [14] S.C. Newton and S.Mitra, "Application of Neural Networks in Fuzzy Pattern Recognition," presented at the Research Conference on Neural Networks for Vision and Image Processing, Boston University, May 10-12, 1991.
- [15] R. N. Lea, J. Villarreal, Y. Jani and C. Copeland, "Fuzzy Logic Based Tether Control," NAFIPS 1991 Workshop Proc. pp 398-402, 1991.

- [16] K.S. Narendra and K. Parthasarathi, "Identification and Control of Dynamical Systems using Neural Networks," IEEE Transactions on Neural Networks, Vol. 1, No. 1, pp 4-26, March 1990.
- [17] J.C. Bezdek, E.C. Tsao and N.R. Pal, "Fuzzy Kohonen Clustering Networks," Proc. of First IEEE Conference on Fuzzy Systems, San Diego, March 1992.
- [18] Y.S. Kim and S. Mitra, "A Comparative study of the performance of Fuzzy ART type clustering algorithms in pattern recognition," SPIE Proc. on Intelligent Systems, Robotics and Computer Vision, Boston, November 1992.

# Character Recognition Using a Neural Network Model with Fuzzy Representation

NASSRIN TAVAKOLI AND DAVID SENIW  
Department of Computer Science  
University of North Carolina at Charlotte  
Charlotte, NC 28223

520-63

161820

P-10

## Abstract

*The degree to which digital images are recognized correctly by computerized algorithms is highly dependant upon the representation and the classification processes. Fuzzy techniques play an important role in both processes. In this paper the role of fuzzy representation and classification on the recognition of digital characters is investigated. An experimental Neural Network model with application to character recognition has been developed. Through a set of experiments, the effect of fuzzy representation on the recognition accuracy of this model is presented.*

Keywords: Statistical, Syntactical, Neural Network, Fuzzy Techniques

## 1. Introduction

Three primary processes are utilized in most pattern recognition systems. 1) The representation process in which the raw digitized data is mapped into a higher level form, such as a feature vector (statistical techniques) or pattern elements constituting pattern grammars (syntactical techniques). 2) The generation of a known base containing the high-level representation of all known patterns in a problem domain. 3) The identification/classification process which classifies the unknown pattern, given its high-level representation and the known base. A block diagram of a general pattern recognition system is given in Figure 1 [16].

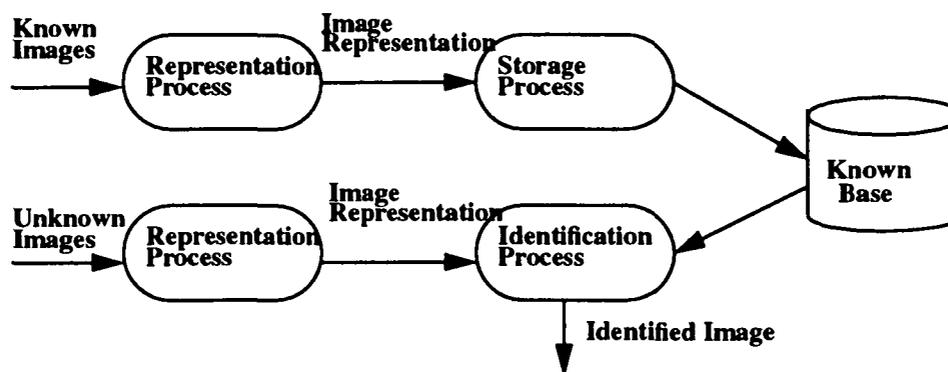


FIGURE 1. A general pattern recognition system.

In both the storage and the identification processes, representation of the image plays a very important role. In fact, the techniques and algorithms used to store the image representation, and the selection of identification techniques are strongly tied to the methods used for representation.

This paper starts by a short introduction to pattern recognition techniques and the role of fuzzy theory on

these techniques. We have selected and implemented a neural network model as the identification process. This model is regarded as a fuzzy classifier since it provides the degree of membership rather than an exact match. For the representation process we start with raw images and through pyramid reductions, provide different levels of resolutions (exactness/fuzziness). Representations of images at each level are used as inputs to the classifier. The purpose is to find an appropriate representation level for this model and gain some understanding on the level of fuzziness required in general (regardless of the classifier) that could result in the best recognition. For comparison, the same character representations were used in conjunction with a template matching identification process.

The results of experiments on a set of 702 unknown digitized characters are given.

## **2. Pattern Recognition Techniques**

Although no unified approach exists for pattern recognition, the majority of techniques that have been developed are in general categorized into two major approaches, namely, the statistical [1,2,10,17] and the syntactical [3,4,6,7,12,14,15] approaches. A distinguishing factor between the syntactical and statistical approaches is the representation and identification processes. Fuzzy techniques play an important role in both the syntactical and Statistical approaches. A thorough discussion and review of fuzzy techniques in pattern recognition is given by Kandel [8].

*Statistical approaches* use a feature component vector, where the vector contains representations of independent pattern elements that are extracted from the image. The identification/classification process is based on a similarity measure that in turn is expressed in terms of a distance measure or a discriminant function. Fu [5] provides a discussion of several important discriminant functions.

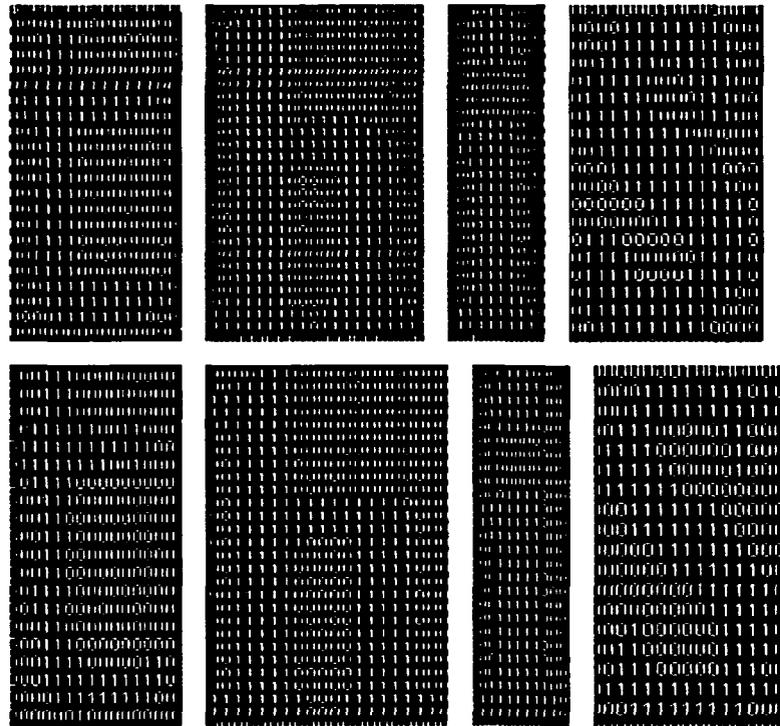
*Syntactical approaches* represent the image as a tree or graph of pattern elements and their relationships. A set of syntax rules, called pattern grammar, is used to represent this relationship. This type of representation would require the identification process to use syntax parsing techniques.

The *fuzzy set theory* introduced by Zadeh [19] has played an important role in both statistical and syntactical approaches. The main purpose of using fuzzy sets has been to represent the inexactness of patterns belonging to certain categories. In statistical methods, the classification algorithm yields the degree of membership of an object in a particular class. In syntactical methods, fuzzy formal languages [11] and parsing methods have been introduced.

*Neural Network* models used in pattern recognition can be considered as a statistical approach in which the classifier (i.e. the neural net) provides the degree of the membership of the unknown object in each of the known classes. Hence the neural network model can be considered as a fuzzy classifier.

## **3. The Representation Process**

The input to the representation process consists of a known base of 26 digitized characters with 5 instances of each character and an unknown base of 702 characters used for recognition. A sample of these characters are shown in Figure 2. These characters were extracted from a digitized text scanned at 240 pixels/inch.



**FIGURE 2. Sample of Digitized Characters**

As noticed from this sample, the characters are noisy, with ragged edges and the digitized representations of the same character are not identical.

In the experimentations, different representations of these characters are used. These representations are:

1. The raw image. As shown in figure 2, the raw image is a digitized character converted into a binary form of zeros (spaces) and ones (body of the character).
2. A pyramid reduction of 2
3. A pyramid reduction of 3
4. A pyramid reduction of 4
5. A pyramid reduction of 5
6. A pyramid reduction 6
7. A pyramid reduction of 7
8. A pyramid reduction of 8

A pyramid is a successive reduction of an image to a lower resolution by representing a block of an image with one pixel. The value of this pixel is determined by the ratio of dark to light pixels (i.e. the threshold factor). We have used a threshold factor of .45 in all pyramid reductions. The selection of this threshold factor was due to a series of experimentations for finding the most optimal value. Figure 3 shows the result of the pyramid reduction.

0000000111111111000000000000  
0000011111111111111100000000  
0000011111111111111100000000  
0001111111111111111100000000  
0001111111111111111100000000  
01111111110111111111000000  
0111111110000000111111000000  
0111111110000000111111000000  
111111100000000111111000000  
1111111000000000111111000000  
011111100000001111111000000  
011111100000001111111000000  
000110000001111111111000000  
00000000111111111111000000  
00000000111111111111000000  
00000111111111111111000000  
00000111111111111111000000  
000111111100000111111000000  
01111111000000111111000000  
011111111000000111111000000  
111111100000000111111000000  
1111111000000000111111000000  
1111111000000000111111000000  
1111111000000000111111000000  
11111110000000111111111100  
11111111111111111111111111  
11111111111111111111111111  
01111111111111111111111100  
01111111111111111111111100  
00011111110000000111000000  
00011111110000000111000000

No reduction

00111111110000  
01111111111000  
11111111111100  
11110001111100  
11110001111100  
11110001111100  
01101111111100  
00111111111100  
01111111111100  
11110001111100  
11110001111100  
11110001111100  
11111111111111  
11111111111111  
11111111111111

Pyramid reduction of 2

0011111000  
0111111100  
1110001100  
1110011100  
0001111100  
0111111100  
1110001100  
1110001100  
1111111111  
1111111111

Pyramid reduction of 3

0111110  
1111110  
1100110  
0011110  
1110110  
1100110  
1111111

Pyramid reduction of 4

011110  
110110  
011110  
011110  
110110  
111111

Pyramid reduction of 5

01110  
11010  
01110  
11010  
11111

Pyramid reduction of 6

1110  
1011  
1111  
1111

Pyramid reduction of 7

111  
111  
101

Pyramid reduction of 8

FIGURE 3. A sample of reduced characters

#### 4. The Neural Network Model

The Neural Network Model implemented for these experiments is based on the Bidirectional Associative Memory Model[9,18] with two layers, without the feed back mechanism. In terms of recognition accuracy this model may not be the most optimal model for this application. However, our purpose was to examine the effect of different levels of representation on the recognition accuracy. Figure 4 shows a high level representation of this model.

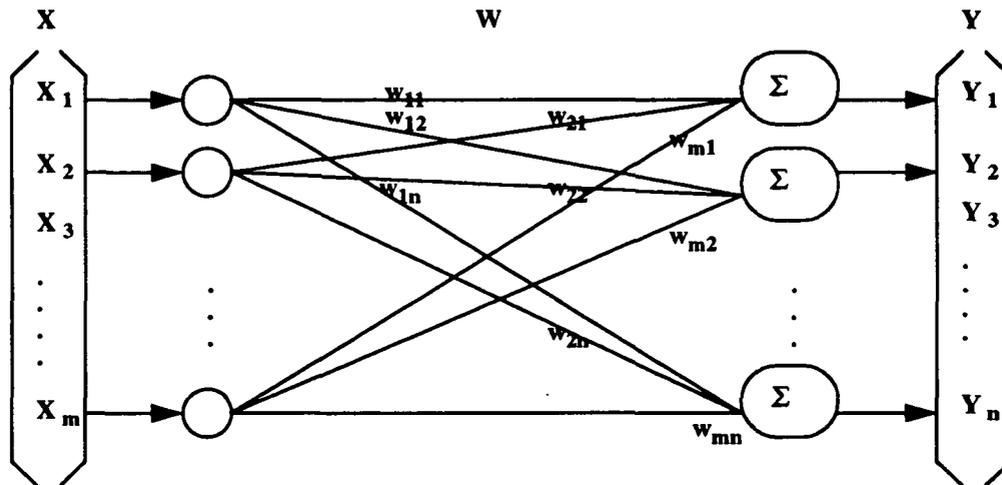


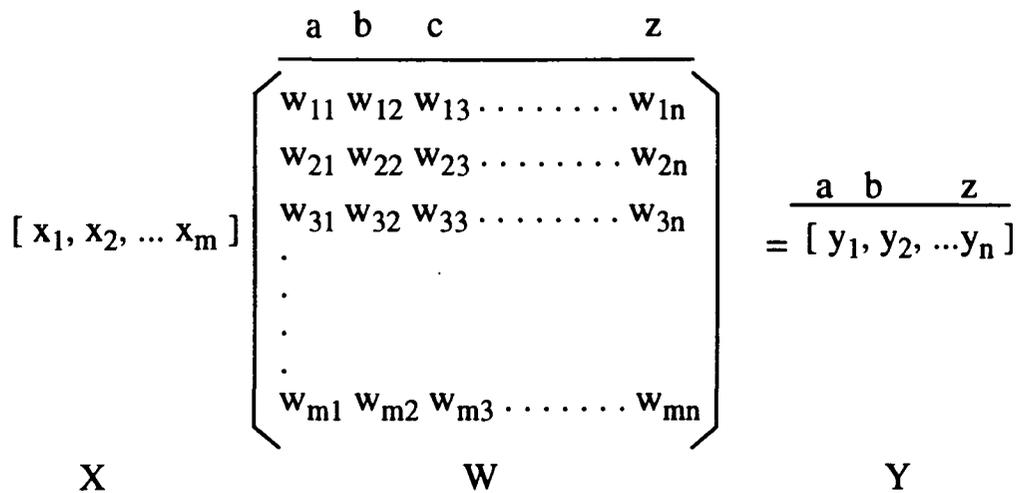
FIGURE 4. The Neural Network Model

This model uses two layers, an input and an output layer. No weighting is performed at the input layer. The links between each of the input nodes and each of the output nodes is weighted with an integer value (positive or negative). The output nodes sum each of the input node values multiplied by it's associated weight. The result of this summation is represented by the vector  $Y$ .

The implementation of the model using the character representations are shown in the following section.

##### 4.1 Implementation of the Neural Network Model

The input layer consists of a node for each feature of the character being recognized. Initially each feature is equivalent to one pixel element. The value of this feature is either zero or one depending on whether the pixel is light or dark. However, after the image is subjected to a pyramid reduction, each feature now represents several pixels. Initially, each character is represented by  $31 \times 31$  or 961 pixels. The output layer is made up of 26 different nodes (one for each possible character). Figure 5 shows an implementation model of this model using the input vector  $X$ , the output vector  $Y$  and the weight matrix  $W$ .



**FIGURE 5. The Neural Network Implementation Model**

In this application, X, Y, and W contain the following values.

X = a vector of pixels belonging to one character.

Y = a vector of positive and negative integers where the index to the vector represents a character such that:

$$Y(a) = y_1$$

$$Y(b) = y_2$$

.

.

$$Y(z) = y_{26}$$

W = a weight matrix where the columns are associated with different characters and rows are associated with the weight per pixel of each character.

$$W(a) = w_{11} \text{ to } w_{m1}$$

$$W(b) = w_{12} \text{ to } w_{m2}$$

.

.

$$W(z) = w_{1n} \text{ to } w_{mn}$$

m = number of pixels per character

n = number of classes of character = 26

Each character is 'taught' to the network by modifying the weight matrix. During the identification phase, the output node with the largest (or the most positive) value indicates the class in which the unknown character belongs to. Note that output nodes with the second, third, etc. largest values indicate characters that are similar to that particular character. In an actual Neural Network hardware solution, the input layer would be 961 processors, each collecting information about their pixel. The output layer would be made up of 26 processors each containing the 961 weights for modifying the signals coming from the input layer nodes. These weights would be created during the learning phase. In our simulation the weights are all

stored in a two dimensional matrix.

## 4.2 Teaching the Neural Network

Teaching the Neural Network is the process of recursively modifying the values of the weight matrix W. This process consists of the following steps:

1. Convert all characters into a one dimensional vector of 0's and 1's.
2. Convert the vector of 0's and 1's into a bipolar vector with negative ones (-1) representing the zeroes. This constitutes the vector X of length m as shown in figure 5. The value of m varies from 961 (no pyramid reduction) to 9 (pyramid reduction of 8).
3. Initialize matrix W to 0.
4. Calculate the weight matrix using the following algorithm:

```

For l = a to z
  Y(l) = -1
End
For Instances = 1 to 5
  For k = a to z
    Y = -1 /* initialize Y vector
    Y(k) = 1
    For i = 1 to m
      For j = 1 to n
        wij = wij + xi * Yj
      End
    End
  End
End
End

```

Figure 6 shows two instances of the weight matrix after characters "a" and "b" with a pyramid reduction of 5 have been taught to the network.

Character	Bipolar Vector	The Weight Matrix after learning 'a'
'a'		
011110	-1	-1 1 1 ... 1
110110	1	1-1-1 ...-1
011110	1	1-1-1 ...-1
011110	1	1-1-1 ...-1
110110	1	1-1-1 ...-1
111111	-1	-1 1 1 ... 1
	1	1-1-1 ...-1
	1	1-1-1 ...-1
	.	.
	.	.
	1	1-1-1 ...-1
	1	1-1-1 ...-1
	1	1-1-1 ...-1
	1	1-1-1 ...-1
		1-1-1 ...-1 <--- the Y vector

Character	Bipolar Vector	The Weight Matrix after learning 'a' and 'b'
'b'	'b'	
110000	1	-2 2 0 ... 0
010000	1	0 0-2 ...-2
011111	-1	2-2 0 ... 0
010001	-1	2-2 0 ... 0
010001	-1	2-2 0 ... 0
011110	-1	0 0 2 ... 2
	-1	2-2 0 ... 0
	1	0 0-2 ...-2
	-1	0 0 2 ... 2
	.	.
	.	.
	1	0 0-2 ...-2
	1	0 0-2 ...-2
	1	0 0-2 ...-2
	-1	2-2 0 ... 0
		-1 1-1 ...-1 <--- the Y vector

**FIGURE 6. Snapshot of the weight matrix after teaching characters "a" and "b"**

### 4.3 Recognizing Characters with Neural Network

The recognition of a character requires the linearization and bipolarization of each of the image features (just as in the teaching section). Then simply perform a matrix multiplication of the vector X on the weight matrix W, placing the result in the vector Y. The closest match is obtained by finding the largest positive value in Y (or the smallest negative value, as Y is usually composed of all negative values). The character associated with index of the node of the largest value is selected. For example, if the third node in Y (i.e. Y(c)) had the largest positive value or the smallest negative value, then the recognized character is "C". Note that other close matches may be discovered by finding the second, third, fourth, etc. largest values in Y. Figure 7 shows a snapshot of the state of network when character "C" is recognized.

Unknown Character	Bipolar Vector	The Weight Matrix after all characters have been taught
011111	-1	-18-6-26...-16
110011	1	-78-78-82...-88
100000	1	-14-34-14...-24
110000	1	-14-34-14...-24
110011	1	-56-76-56...-66
011110	1	4 4 16 ... 14
	1	24 4 20 ...14
	1	-54-54-54...-64
	-1	36 36 36 ...46
	.	.
	.	.
	1	-76-76-76...-86
	1	-54-58-54...-64
	1	-48-48-48...-58
	-1	18 -2 -2 ... 8
		-658 -742 -454...-730 <---The results of the matrix multiplication of Y(a)Y(b)Y(c) ...Y(y) unknown character and the weight matrix.

**FIGURE 7. Snapshot of the network during recognition**

In this example, the least negative value was the integer associated with  $Y(c)$ , indicating that the neural nets choice for the image was a 'c'.

## 5. Experimental Results

A series of experiments were conducted for recognition of the 702 digitized characters. These experiments were varied over the following parameters:

1. The identification process (a. the neural network model, b. the template matching).
2. The number of instances (1,2,3,4,5) of each known character for teaching the identification process.
3. The image representation (raw, pyramid reductions of 2, 3, 4, 5, 6, 7, and 8).

Figure 8 shows the effect of each varying parameter on the recognition accuracy of the 702 unknown characters.

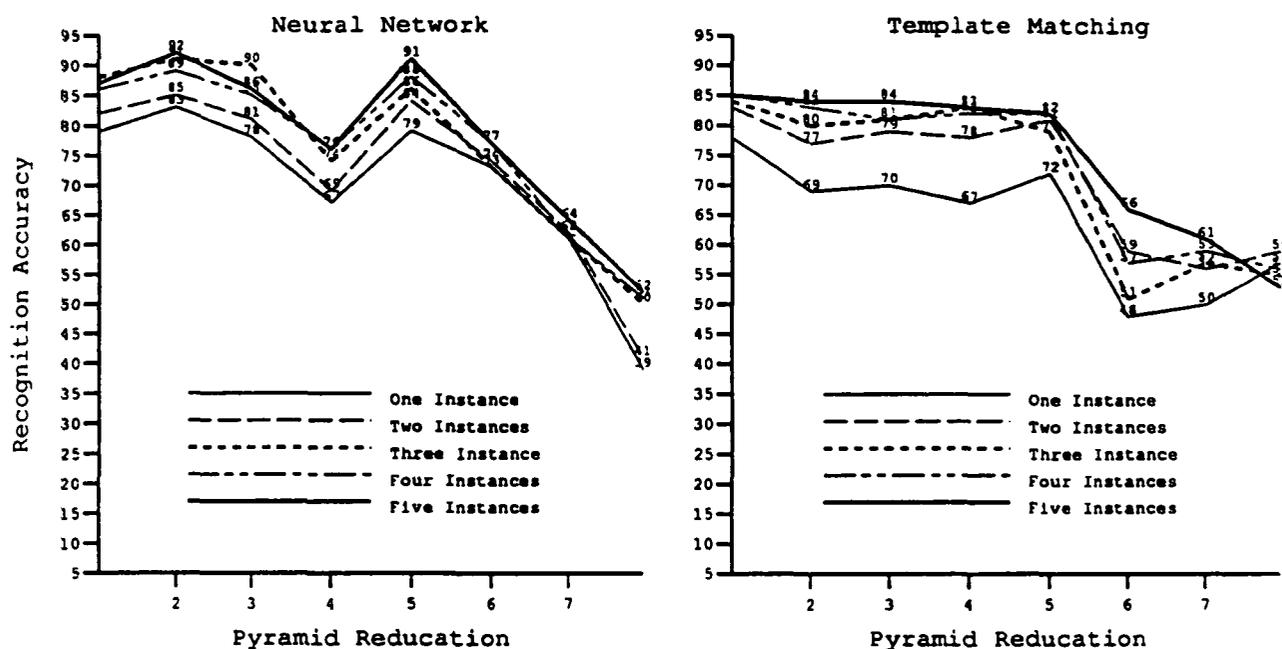


FIGURE 8. Comparison of Neural Network and Template Matching approaches.

The following conclusions can be drawn from the above figure.

- a) The Neural Network approach can provide a higher recognition accuracy than the template matching approach. This is due to the fact that the Neural Network approach is a fuzzy classifier whereas the template matching approach is an exact classifier.
- b) In general as the number of known instances for each character increases, better recognition is achieved in both approaches. The Neural Network approach however, slightly deviates from this fact. In one case (i.e. pyramid reduction of 3) three instances provide better recognition than five instances.
- c) In template matching, the raw image (i.e. no pyramid reduction) is better than any pyramid reduction. As noticed, the curve is almost flat, implying that this approach is less sensitive to the representation process. In Neural Network approach, peaks are shown at pyramid reductions of 2 and 5.

- d) In general, a pyramid reduction of 5 seems to provide a good recognition in both approaches. This is an interesting phenomenon since at this representation level, it is more difficult for the human vision to recognize accurately (see figure 3).

## 6. Conclusion

The purpose of this research was to gain some understanding on the level of fuzziness required in the representation and the identification processes for better recognition of digitized characters. We also were interested in the degree of the interdependency between the representation process and identification process.

A Neural Network model and a Template Matching model for recognition of digitized characters were implemented. For the representation process, we used pyramid reductions of 1 to 8. Through a series of experiments we concluded that the optimal amount of fuzziness to be introduced by the representation process is totally dependant upon the identification process. As expected some level of fuzziness in both the representation and the identification processes contributed to better recognition. The Neural Network approach in general proved to be a better identifier due to its fuzzy classification property. While no general representation can be found to be the optimal, it seems that a pyramid reduction of 5 provided good recognition (i.e. above 80%) in both models.

With further experiments, we found that different approaches for the representation and identification, resulted in the recognition of a different sets of character. By combining two different techniques therefore, a recognition of 100% on the same set of characters were achieved.

## References

- [1] Devijver, P. A., and Kittler, J., *Pattern Recognition, A Statistical Approach*, New York:Academic Press,1982.
- [2] Fukunaga, K., *Introduction to Statistical Pattern Recognition*, New York:Academic Press, 1972.
- [3] Fu, K. S., *Syntactic Methods in Pattern Recognition*, New York:Academic Press, 1974.
- [4] Fu, K. S., "Recent Advances in Syntactic Pattern Recognition." *Proc. 4th Int. Joint Conf. on Pattern Recognition*, 1978, pp. 13-18.
- [5] Fu, K. S. (Ed.), *Digital Pattern Recognition*, New York:Springer-Verlag, 1980.
- [6] Fu, K. S., *Syntactic Pattern Recognition and Applications*, Englewood Cliffs:Prentice-Hall, 1982.
- [7] Gonzalez, R. C. and Thompson, M. G., *Syntactic Pattern Recognition, An Introduction*, Reading:Addison Wesley, 1978.
- [8] Kandel, A., *Fuzzy Techniques in Pattern Recognition*, John Wiley & Sons, 1982.
- [9] Kosko, B., "Constructing an Associative Memory", *Byte*, Sep. 1987, pp. 137-144.
- [10] Kovalevsk, V. A., "Recent Advances in Statistical Pattern Recognition." *Proc. 4th Int. Joint Conf. on Pattern Recognition*, 1978, pp. 2-12.
- [11] Lee, E.T., "Fuzzy Languages and Their Relation to Automata", *Ph.D. Thesis*, Dept. of Elect. Eng. and Comput. Sci., Univ. of California, Berkeley, 1972.
- [12] Pavlidis, T., *Structural Pattern Recognition*, Springer-Verlag, 1977.
- [13] Segen, J., "Learning Structural Descriptions of Shape.," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 1985, pp. 96-99.
- [14] Tanimoto, S. and Klinger, A. (Eds.), *Structural Computer Vision*, New York:Academic press, 1980.
- [15] Thomason, M. and Gonzalez, R. C., *Syntactic pattern recognition: an introduction*, Addison-Wesley, 1981.
- [16] Tavakoli, N., "A New Approach to Pattern Recognition.", *Doctoral dissertation*, Kansas State University, 1986.
- [17] Young, T. Y. and Fu, K. S., (Eds.), *Handbook of Pattern Recognition and Image Processing*, New York:Academic Press,1986.
- [18] Wasserman, P. D., *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, New York, 1989.
- [19] Zadeh, L. A., "Fuzzy sets", *Inf. and Control*, vol. 8, 1965, pp. 338-353

Designing a Fuzzy Scheduler for Hard Real-Time Systems R p

John Yen, Jonathan Lee, Nathan Pfluger, and Swami Natarajan  
Department of Computer Science  
Texas A&M University  
College Station, TX 77843

### Abstract

In hard real-time systems, tasks have to be performed not only correctly, but also in a timely fashion. If timing constraints are not met, there might be severe consequences. Task scheduling is the most important problem in designing a hard real-time system, because the scheduling algorithm ensures that tasks meet their deadlines. However, the inherent nature of uncertainty in dynamic hard real-time systems increases the problems inherent in scheduling. In an effort to alleviate these problems, we have developed a *fuzzy scheduler* to facilitate searching for a feasible schedule. A set of fuzzy rules are proposed to guide the search. The situation we are trying to address is the performance of the system when no feasible solution can be found and therefore certain tasks will not be executed. We wish to limit the number of important tasks that are not scheduled.

## 1 Introduction

Real-time scheduling is a problem which is the key part of designing the operating system for a hard real-time system, and is thus tightly dependent on the architecture of the target system.

Basically, there are two types of real-time systems[2], *soft* real-time systems, and *hard* real-time systems, see figure 1. In soft real-time systems, tasks are performed by the system as fast as possible, but they are not constrained to finish by specific times. The only constraint on the system is to minimize response time. On the other hand, in hard real-time systems, tasks must be performed before their deadlines or there might be severe consequences.

To further break down this taxonomy, hard real-time scheduling can be classified into two categories, static [3], and dynamic [4, 6, 5, 7, 8]. A static real-time scheduler computes schedules for tasks off-line and requires complete prior knowledge of a set of tasks' characteristics such as arrival time, computation time, deadline and so on. A dynamic approach, on the other hand, calculates the schedules on-line and allows tasks to be dynamically invoked. Although static approaches have low run-time cost, they are inflexible and cannot adapt to a changing environment or to an environment whose behavior is not completely predictable. When new tasks are added to a static system, the schedule for the entire system must be recalculated,

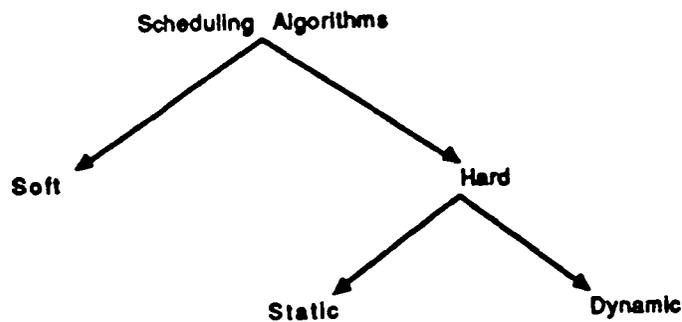


Figure 1: A Taxonomy of Real-Time Scheduling Algorithms

which is expensive in terms of the time and cost. In contrast, dynamic approaches involve higher run-time costs, but, because of the way they are designed, are flexible and can easily adapt to changes in the environment.

Our motivation to develop a fuzzy logic based approach to the dynamic scheduling problem are two fold. First, in a dynamic hard real-time system, not all the characteristics of tasks (e.g., precedence constraints, resource requirements, etc.) are known *a priori*. For example, the arrival time for the next task is unknown for aperiodic tasks. To be more precise, there is an inherit uncertainty in hard real-time environment which will worsen scheduling problems (e.g. arbitrary arrival time, and uncertain computation time). Characteristics of a task that may be uncertain include expected next arrival time, criticality, or importance of the task, system load and/or predicted load of individual processors, and run time, or more specifically average vs. worst-case run time.

Second, there is the possibility of system overload. In the case of overload we want to degrade gracefully by ensuring that the most important tasks are run first, thus allowing an amount of flexibility in the scheduler under adverse conditions to determine which tasks are run and which are not.

Therefore, our goal is to develop an approach to hard real-time scheduling that can be applied to a dynamic environment involving a certain degree of uncertainty and a possibility of overload situations. In this paper, we concentrate on a hard real-time system on a nonpre-emptable uniprocessor system with a set of independent tasks. These tasks will have arbitrary arrival times and will be characterized by worst-case computation time and task criticality.

Therefore, we have developed a *fuzzy scheduler* that includes the following features. First, the scheduling process is treated as a search problem, as suggested by [7, 8], in which the search space consists of a tree where the root is an empty schedule, an intermediate node is a partial

schedule, and a leaf is a complete, though not necessarily feasible, schedule. Second, a set of fuzzy rules are used to guide the search of a feasible schedule. A feasible schedule is one that schedules all the tasks so that they may meet their deadlines.

In case no feasible schedule can be found, we then want the scheduler to ensure that it schedules the tasks according to some intelligent heuristic. Some possible heuristics include scheduling the most tasks, scheduling the most important tasks, etc. We also wish to include the possibility of more intelligent heuristics, such as schedule the most important tasks, but only if it allows most of the tasks to be executed.

A background about hard real-time schedule is introduced in the next section. An overview of our fuzzy scheduler and an example set of rules for one type of overload heuristic are presented in section 3. An outline of the benefits and applications for our scheduler is given in section 4. An example to demonstrate the feasibility of our approach is in section 5. Finally, we summarize the advantages and disadvantages of our approach.

## 2 Background on Hard Real Time Scheduling

The function of a scheduling algorithm is to determine, for a given set of tasks, whether a schedule for executing the tasks exists such that the timing, precedence, and resource constraints for the tasks are satisfied, and to calculate such a schedule if one exists. A schedule is said to be feasible if it contains all the tasks, and all tasks will meet their deadlines. A scheduling algorithm is said to be optimal if it finds a feasible schedule whenever one exists for a given set of tasks.

Most of the work in hard real-time scheduling in the early 70's is accredited to Liu and Layland[3]. In that paper, two algorithms were discussed, tested, and declared to be optimal. These algorithms are the RMS, rate monotonic scheduler, and EDF, earliest deadline first. The largest problem with these algorithms is the set of restrictions placed on the problem set that they solve. Later, another dynamic algorithm, least laxity, was also proposed and found to be optimal. For the case of least laxity and EDF, optimal is defined to be that if there is a feasible schedule for a set of tasks, then these algorithms will find one.

Stankovic and Ramamritham wanted to broaden the areas covered by real-time systems to include intelligent schedulers working on distributed systems [4, 6, 5, 7, 8]. Their method for designing a hard real-time system on a distributed system was to associate with each node in the distributed system a local scheduler. The function of the local scheduler was to receive tasks from the system and attempt to guarantee them to be run on this node. Those tasks that could not be guaranteed were then sent to another node. The method of sending tasks was through a bidding system, where each of the nodes bid for a task depending on its current state and predicated amount of free time. The basic rationale behind their approach is the notion of a "guarantee algorithm". An algorithm is said to guarantee a newly arriving task if the algorithm can find a schedule for all the previously guaranteed tasks and the new task, such that each

task finishes by its deadline.

The main problem with this approach was determining a fast uniprocessor scheduling algorithm, that was both adaptive and intelligent. The method that they developed was to transform the scheduling problem into a tree search problem, where the root of the tree was an empty schedule, an intermediate vertex was a partial schedule, and the leaves were complete schedules. It can be proven that if a partial schedule is found to be infeasible, i.e., all tasks currently scheduled do not meet all the deadlines, then any complete schedule derived from this schedule will still be infeasible.

Although determining infeasibility of partial schedules cut down on the number of nodes to be searched, the problem was still NP-complete. The next step to making this method economical and intelligent was to derive a heuristic function which helped guide the search. The earliest heuristic functions simulated EDF and Minimum Processing Time first. Later more complex functions were tested and found to be better than simple heuristic functions.

Another related work regarding the integration of the importance (i.e., criticality) and the deadline of a task in hard real-time scheduling is addressed in [1]. In this paper, they adopt a similar approach to us by adding criticality as one of the major factors considered into their heuristic function to guide the search of feasible schedule.

Our work is an attempt to extend the notion of using a heuristic function for guiding the search in an intelligent manner. The goal of our work is to consider a complicated situation involving several major factors (e.g. deadline, criticality, and earliest starting time). Due to the use of fuzzy logic in representing our guidance mechanisms, they will be easy to express, comprehend, and modify.

### **3 A Fuzzy Scheduling Approach**

#### **3.1 A Scheduling Approach Based on Fuzzy Logic**

In dynamic hard real-time scheduling, the nature of the task involves a certain degree of uncertainty, which increases the difficulty of developing a feasible and reliable scheduling algorithm. In order to alleviate the problems associated with hard real-time scheduling, we first treat the scheduling problem as a searching problem. We then developed a set of fuzzy rules to guide the search for a feasible schedule.

We have designed our system to handle a set of aperiodic tasks with arbitrary arrival times. In addition, any periodic task is considered to be a series of aperiodic tasks, each of which is an instance of that periodic task, and are denoted by  $T(x)$ , where  $x = 0, 1, 2, \dots, n$ . This method allows us to handle both periodic and aperiodic tasks equally, while still gaining benefits about knowing some of the tasks arrival times.

The major factors considered in our approach to determine the scheduling are task deadline,

criticality, and earliest start time. A deadline is a specific time by which a task must be finished. A task's criticality is the importance of the task. The earliest starting time for a task is the earliest time that a task can submit itself to the scheduler. The earliest start time for an aperiodic task is the current time, while the earliest start time for a periodic task will be a known future time computable from the task's characteristics.

The inputs of these parameters are fuzzified and represented as linguistic variables. The computation of these variables must be done every time the scheduler is executed, due to the dynamic nature of the system. Fuzzy rules are then applied to those linguistic variables to compute the *level value* for deciding which task will be selected to be scheduled next.

The linguistic variables for the three parameters chosen are:

- Task deadline: early, medium, late.
- Task criticality: important, average, unimportant.
- Task earliest starting time: early, medium, late.

To compute the results, we perform a reasoning process using fuzzy rules. The format of our fuzzy rules begins with IF as the left-hand side and ends with THEN as right-hand side. In the left-hand side, we use both basic and modified linguistic variables for the above-mentioned factors, while, in the right-hand side we assign a fuzzy number as a level value of that particular task. For example,

- IF the incoming task has an early deadline, an important criticality, and a medium earliest starting time, THEN assign level  $\sim 7$ .

As a result of the inference, several fuzzy rules will be initiated. We will then combine the fuzzy conclusions of all the rules that are initiated to produce a fuzzy variable which represents the level of the task. This variable will then be defuzzified to produce a crisp level to be compared to the other tasks for the purpose of choosing which task to schedule next.

For example, a scheduling snapshot starting at time 7 has three tasks T1, T2, T3 with the following characteristics:

Task	Earliest-Start-Time	Deadline	Computation-time	Criticality	Level-value
T1	10	15	3	5	$\sim 5$
T2(1)	20	32	6	10	$\sim 3$
T3	7	14	3	10	$\sim 10$

Comparing the deadline of three tasks, we interpret T3 as early, T1 as medium, T2(1) as late. The criticality of T3 is important, T1 is medium and T2(1) is important. According to the example set of rules (in section 3.2), the level value of each task is: T3 ( $\sim 10$ ), T1 ( $\sim 5$ ), and T2(1) ( $\sim 3$ ). Therefore, the task T3 will be scheduled first, and then T1, and then finally T2(1).

### 3.2 Example Set of Rules

Fuzzy rules can be expressed as English sentences or as a set of if-then clauses using linguistic variables. One possible set of rules is:

- if deadline is early
  - if criticality is important  $\Rightarrow$  level  $\sim 10$
  - criticality is average
    - \* if earliest starting time is early  $\Rightarrow$  level  $\sim 9$
    - \* if earliest starting time is medium  $\Rightarrow$  level  $\sim 8$
    - \* if earliest starting time is late  $\Rightarrow$  level  $\sim 7$
  - if criticality is unimportant
    - \* if earliest starting time is early  $\Rightarrow$  level  $\sim 8$
    - \* if earliest starting time is medium  $\Rightarrow$  level  $\sim 7$
    - \* if earliest starting time is late  $\Rightarrow$  level  $\sim 6$
- if deadline is medium
  - if criticality is important
    - \* if earliest starting time is early  $\Rightarrow$  level  $\sim 6$
    - \* if earliest starting time is medium  $\Rightarrow$  level  $\sim 5$
    - \* if earliest starting time is late  $\Rightarrow$  level  $\sim 4$
  - if criticality is average  $\Rightarrow$  level  $\sim 5$
  - if criticality is unimportant  $\Rightarrow$  level  $\sim 4$
- if deadline is late
  - if criticality is important  $\Rightarrow$  level  $\sim 3$
  - if criticality is average  $\Rightarrow$  level  $\sim 2$
  - if criticality is unimportant  $\Rightarrow$  level  $\sim 1$

We chose to treat deadline as the most important principle behind choosing a task for scheduling because the major purpose of hard real-time scheduling is to meet the deadline. After this came criticality, and then earliest starting time. We felt that when a task must be scheduled, it is important to consider the tasks that must be done immediately. After that, more critical tasks are considered so that they are scheduled even when some of the tasks fail to be scheduled.

## 4 Benefits and Applications

There are several benefits of our approach: (1) robustness because of the boundary conditions of scheduling parameters are represented as a part of fuzzy subset; (2) flexibility due to easy integration of other requirements into the fuzzy rule set; (3) simplicity in term of understandability and using less rules; and (4) the intelligent scheduling system that may be derived using fuzzy logic.

By using fuzzy logic, the rules for determine task order are concise, intelligible, and easily modified. The rules that we have derived are based on basic rule of thumb theories about task scheduling and the properties we wish our system to show.

Recently, the development of fuzzy logic chip has a major progress. The Microelectronics Center of North Carolina successfully completes the fabrication of the world's fastest fuzzy logic chip. In the architecture of our fuzzy scheduler, a fuzzy logic chip can be used to implement part of fuzzy scheduler.

The use of a uniprocessor scheduling algorithm may be limited due to the distributed nature of today's systems. However, our approach, when used with a method of offloading unscheduled tasks to other nodes, may be useful for distributed systems. In future work, we hope to apply fuzzy logic to the problem of inter-processor communication and load balancing.

## 5 An Example

To demonstrate our system we have constructed an example scenario (Figure 2). The time frame for this example is limited to 35 for simplicity. The scheduling snapshots for different starting times are given to illustrate our approach:

In figure 3, tasks T1 and T2 are periodic tasks. T1 has four instances, T1(0), T1(1), T1(2) and T1(3). T2 has two instances, T2(0) and T2(1). T1(0) will be scheduled first, then T2(0), T1(1), T1(2), T2(1), and T1(3), in that order. In figure 4, task T2(0) continues to occupy the processor until it finishes at time 9 because we assume non-preemption for all tasks. T3 will take over and execute for 3 units of time. T1(1) and T1(2) will be scheduled before T2(1) because of the level values assigned. Finally, T1(3) will be executed. In figure 5, task T3 continues to execute until it finishes at time 12. T1(1) will be executed next, followed by T4 for 3 units of time. And then T2(1), T1(3) will take over.

In figure 6, task T1(1) continues to execute until 15. T4 will take over and finish at 18, which is followed by T1(2). T5 will be executed at time 21 and completed by 25. T2(1) and T1(3) are scheduled to be executed then. In figure 7, the last snapshot, T4 continues to execute up to time 18. T6 with the highest level value will be scheduled next and finished at time 20. Task T1(2) will be scheduled next. T5 should be scheduled next and finished up by 27. But by doing so, T5 will *miss its deadline*, therefore, T5 will have to be offloaded. Finally, T2(1) and T(3) will be scheduled.

**Periodic Tasks**

Task	Period	Deadline	Computation-time	Criticality
T1	10	5	3	5
T2	20	12	6	10

**Aperiodic Tasks**

Task	Arrival-time	Deadline	Computation-time	Criticality
T3	7	14	3	10
T4	12	18	3	1
T5	13	25	4	1
T6	16	21	2	20

Figure 2: An example Scenario

Task	Earliest-Start-Time	Deadline	Computation-time	Criticality	Level-value
T1(0)	0,10,20,30	5,15,25,35	3	5	~9
T2(0)	0,20	12,32	6	10	~3

Figure 3: Scheduling snapshot at time 0

Task	Earliest-Start-Time	Deadline	Computation-time	Criticality	Level-value
T1(1)	10,20,30	15,25,35	3	5	~5
T2(1)	20	32	6	10	~3
T3	7	14	3	10	~10

Figure 4: Scheduling snapshot at time 7

Task	Earliest-Start-Time	Deadline	Computation-time	Criticality	Level-value
T1(1)	12,20,30	15,25,35	3	5	~9
T2(1)	20	32	6	10	~3
T4	12	18	3	1	~4

Figure 5: Scheduling snapshot at time 12

Task	Earliest-Start-Time	Deadline	Computation-time	Criticality	Level-value
T1(2)	20,30	25,35	3	5	~5
T2(1)	20	32	6	10	~3
T4	13	18	3	1	~8
T5	13	25	4	1	~4

Figure 6: Scheduling snapshot at time 13

Task	Earliest-Start-Time	Deadline	Computation-time	Criticality	Level-value
T1(2)	20,30	25,35	3	5	~5
T2(1)	20	32	6	10	~2
T5	16	25	4	1	~4
T6	16	21	2	20	~10

Figure 7: Scheduling snapshot at time 16

Notice that although a feasible schedule was not found for time 16, there was an intelligent choice as to which task was not scheduled. Task T5 not only had a late deadline, but was also the least critical task present.

## 6 Conclusion

Due to the fact that there is an inherit amount of uncertainty in dynamic hard real-time systems which increases the problems inherent in scheduling, there is a need to develop a flexible scheduler. We have presented a fuzzy scheduler for hard real-time systems in which we treat the scheduling problem as a search problem, utilize a set of fuzzy rules to guide the search for a feasible schedule, and the scheduler is triggered by a newly arrival task.

The main advantage of our system is that an intelligent choice is made during overload situations to determine which task or tasks cannot be scheduled. This allows the system to gracefully degrade when overloaded.

The current scope of our work is confined to uniprocessor systems. In the future, we plan to (1) address the utilization of the existing schedule when a new task arrives, (2) address the issues of considering and predicting the load of individual processors, (3) investigate the possibility of using fuzzy logic chip as the scheduling co-processor, and (4) extend to distributed systems using either a focused addressing or bidding algorithm to offload tasks that can not be scheduled locally.

## References

- [1] S. R. Biyabani, J. A. Stankovic, and K. Ramamritham. "The Integration of Deadlines and Criticalness in Hard Real-Time Scheduling". *RTSS'88*, pages 152-160, Dec 1988.
- [2] S. Cheng, J.A. Stankovic, and K. Ramamritham. "Scheduling Algorithms for Hard Real-Time Systems - A Brief Survey".
- [3] C.L. Liu and J.W. Layland. "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment". *J. ACM*, pages 46-61, Jan 1973.
- [4] K. Ramamritham and J. A. Stankovic. "Dynamic Task Scheduling in Hard Real-Time Distributed Systems". *IEEE Software*, pages 65-75, July 1984.
- [5] J.A. Stankovic and K. Ramamritham. "The Design of the Spring Kernel". *RTSS'87*, pages 146-157, Dec 1987.
- [6] J.A. Stankovic, K. Ramamritham, and S. Cheng. "Evaluation of a Flexible Task Scheduling Algorithm for Distributed Hard Real-Time Systems". *IEEE Trans. Comp.*, pages 1130-1143, Dec 1985.
- [7] W. Zhao, K. Ramamritham, and J.A. Stankovic. "Preemptive Scheduling Under Time and Resource Constraints". *IEEE Trans. Comp.*, pages 949-960, Aug 1987.
- [8] W. Zhao, K. Ramamritham, and J.A. Stankovic. "Scheduling Tasks with Resource Requirements in Hard Real-Time Systems". *IEEE Trans. on Software Eng.*, May 1987.

**WARP:  
WEIGHT ASSOCIATIVE RULE PROCESSOR  
A DEDICATED VLSI FUZZY LOGIC MEGACELL**

P<sub>1</sub> 10

A. Pagni, R. Poluzzi, G.G. Rizzotto

Corporate Advanced System Architectures  
SGS-THOMSON Microelectronics  
Via C. Olivetti 2  
20041 Agrate Brianza (MI) ITALY

*Abstract.*

*During the last five years Fuzzy Logic has gained enormous popularity, both in the academic and industrial worlds, breaking up the traditional resistance against changes thanks to its innovative approach to tackling problems.*

*The success of this new methodology has led microelectronics industries to create a brand new class of machines, called Fuzzy Machines, to overcome the limitations of traditional computing systems when utilized as Fuzzy Systems.*

*This paper gives firstly an overview of the methods by which Fuzzy Logic data structures are represented in the machines (each with its own advantages and inefficiencies), then introduces WARP (Weight Associative Rule Processor) which is a dedicated VLSI megacell allowing the realization of a fuzzy controller suitable for a wide range of applications.*

*WARP represents an innovative approach to VLSI Fuzzy controllers utilizing different types of data structures for characterizing the membership functions during the various stages of the Fuzzy processing.*

*WARP dedicated architecture has been designed in order to achieve high performance exploiting the computational advantages offered by the different data representation adopted.*

## Section 1. Fuzzy Machines

Computer evolution is tending towards specialized machines which are optimized to meet the needs of particular languages or classes of problems. One result of this trend is that many systems now contain one or more general purpose processors supported by a variety of specialized devices (e.g. mathematical or graphical coprocessors) optimized for specific operations.

While the numerical computation field is comprehensively served by machines and specialized integrated components able to calculate numerical algorithms at very high speed and with great accuracy, there is little cost-effective hardware to support newer approaches to logic, particularly those involving non exact information processing.

In particular, the type of processing required to solve problems using Fuzzy Logic with its peculiar data structures cannot be effectively carried out on machines designed for completely different kinds of algorithms and data representations. To deal with the calculus involving the data structures of Fuzzy Logic such as Fuzzy Sets (with their related membership functions) and Term sets [1], Fuzzy Machines have been introduced.

With respect to the functionality these devices can be gathered into two main groups:

- FUZZY COPROCESSORS
- FUZZY CONTROLLERS

**Fuzzy Coprocessors** represent the equivalent of a general purpose machine with respect to Fuzzy calculus: they are the key for turning standard systems into Fuzzy Systems. These machines should not to be considered as the main processors of a system, but rather as an indispensable support in speeding up fuzzy applications.

**Fuzzy controllers** represent the next step in the evolution of intelligent controls and their use can lead to a technological breakthrough in this area. A Fuzzy controller is a particular Fuzzy device equipped with an interface suitable for driving physical actuators: it accepts deterministic values and produces a deterministic value.

With respect to the technology utilized a Fuzzy Machine can be implemented in the following ways:

- SOFTWARE IMPLEMENTATION
- DEDICATED HARDWARE IMPLEMENTATION
  - HYBRID MACHINES
  - FULLY DIGITAL MACHINES

The **Software implementation** of Fuzzy machines is presently the most widely used one; while this approach well suites off-the-line processing it becomes inadequate whenever processes requiring high or medium high dynamics appear.

Among the **Dedicated hardware implementations**, the **Fully Digital** approach to Fuzzy Logic Dedicated Machines is up to now the most widely employed method of implementation of dedicated machines [2], [3]. The advantages of this technology are the generally known ones:

- Complex data management architectures
- Easy interfacing in existing systems
- Low sensitivity to technology changes

The **Hybrid** (mixed Analog/Digital computing) realization of fuzzy machines may possibly represent the next edge in the computer world [4], in fact Hybrid machines provide a number of significant advantages over digital ones:

- Very high speed system throughput
- High parallelism allowed
- No need for expensive A/D and D/A converters

With this type of technology the problems mainly lie in the representation for the Fuzzy data structures, on the analog memories required by the machine and in the sizing of some components, but great improvements in those areas are expected in the next few years.

A rough picture of performances in terms of FIPS (Fuzzy Inference Per Second) obtainable with various types of platforms (and the type of applications were they are mostly applied) are illustrated in fig. 1.

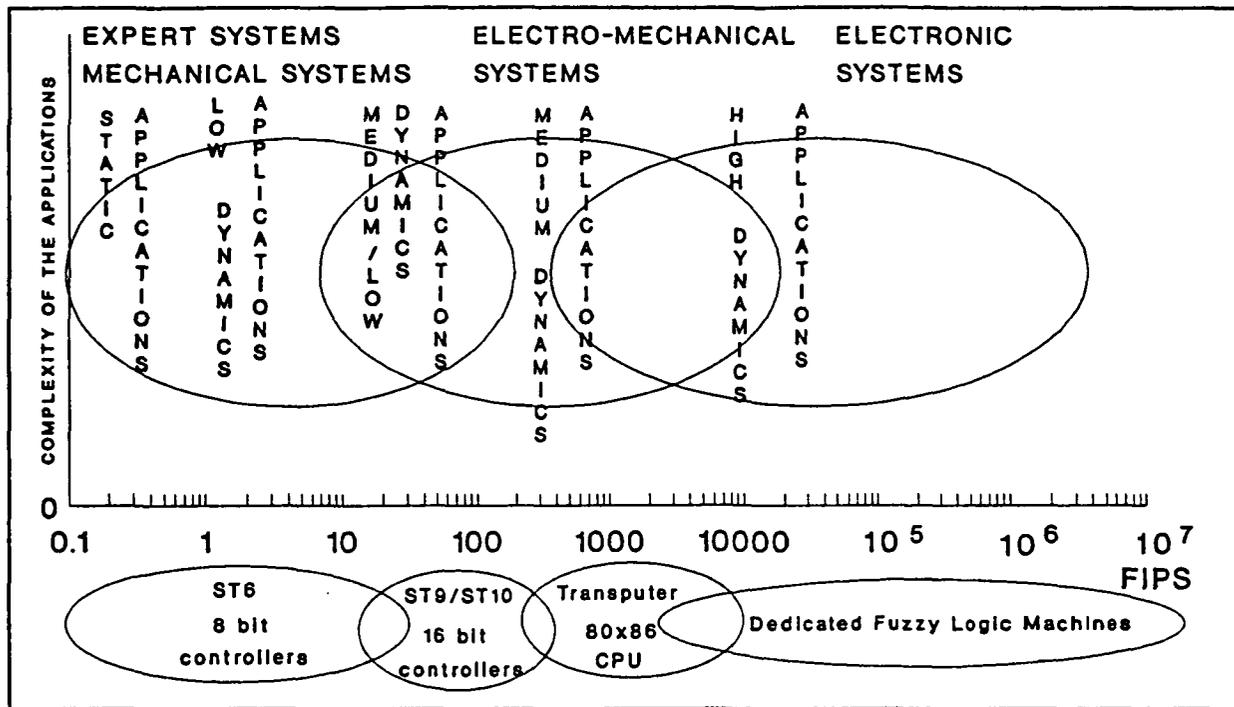


Figure 1

## Section 2. Design of Fuzzy Machines

Among the design approaches to fuzzy machines, and in particular the Software and Fully digital ones, a great advantage lies in the possibility of deciding during the architectural design phase the "kind of machine" that one wants to realize, ranging from the two opposite poles:

- Memory oriented machines
- Computing oriented machines

Memory oriented machines are characterized by having most of the computing performed off-line and then stored in suitable formats inside the memory. This lead to the utilization of large amounts of memory because the membership functions must be described by means of non-optimized data structures (in most cases vectors). Generally with this solution higher performances are possible although with a certain loss in precision.

Computing oriented machines come at the other end of the spectrum. Here the membership functions are stored in compact formats and it is the machine that must operate on those complex data structures performing all of the necessary computing (that is generally finding intersection points and calculating area/weight values). This solution it is generally slower than the previous one but allows a greater precision.

The performances obtainable by the above approaches are greatly influenced by the level of internal parallelism that is actually implemented. It is worth noting that this parameter affects Computing oriented machines more than Memory oriented machines.

Another very important factor in the designing of the fuzzy machines, is the way of representing the membership functions; different methods can be utilized according to where in the rule (IF-part or THEN-part) the connected variable acts.

For the membership functions bounded to the IF-part of the rules there are two main types of representation that are commonly utilized:

- Vectorial representation
- Analytical representation

With the vectorial representation of the membership functions the universe of discourse is divided into a number of elements  $N$ , and the interval  $[0, 1]$  in  $L$  truth levels, creating a vector  $\mu_1(x), \dots, \mu_N(x)$ , where  $\mu_i(x)$  represents the truth level that best approaches the value of the membership function  $\mu(x)$  in the point  $x$ . With this type of characterization, the more critical decision is choosing the most appropriate values of  $N$  and  $L$ . Generally the values for  $N$  range from 25 to 256 and for  $L$  from 10 to 256, according to the type of technology utilized.

With the analytical representation a function that maps the universe of discourse onto the closed space  $[0, 1]$  is provided. This is generally a piecewise linear function described by the breakpoints where the function changes gradient. With this kind of characterization it is left to the machine to calculate the intersection point between the membership function and the function representing the input.

Clearly, the first method, characteristic of memory oriented machines, allows greater performance to be obtained as it is based on look-up tables rather than calculations. On the other hand the value of the intersection must be restricted to those realistically representable with the adopted technology, while in the analytical formalization values as precise as the machine data representation can be obtained.

The choice among the two above methods is generally a trade off between speed and precision.

The value computed from the IF-part of a rule is used to perform the inferential process on the membership functions of the THEN-part. To perform this operation a suitable inferencing method must be used. The two most widely employed ones are:

- Max-Min inferencing method
- Max-Dot inferencing method

The main difference between the two methods lies in the different truncation that is applied to the Membership Functions of the THEN-part of the rules. The choice between one of the above methods of inferencing is influenced by the type of representation of the Fuzzy Sets adopted. The Max-Min Inference method truncates the membership function up to the threshold value  $\theta$  while the Max-Dot Inference utilize the value as a scaling factor. This is clearly explained by fig 2.

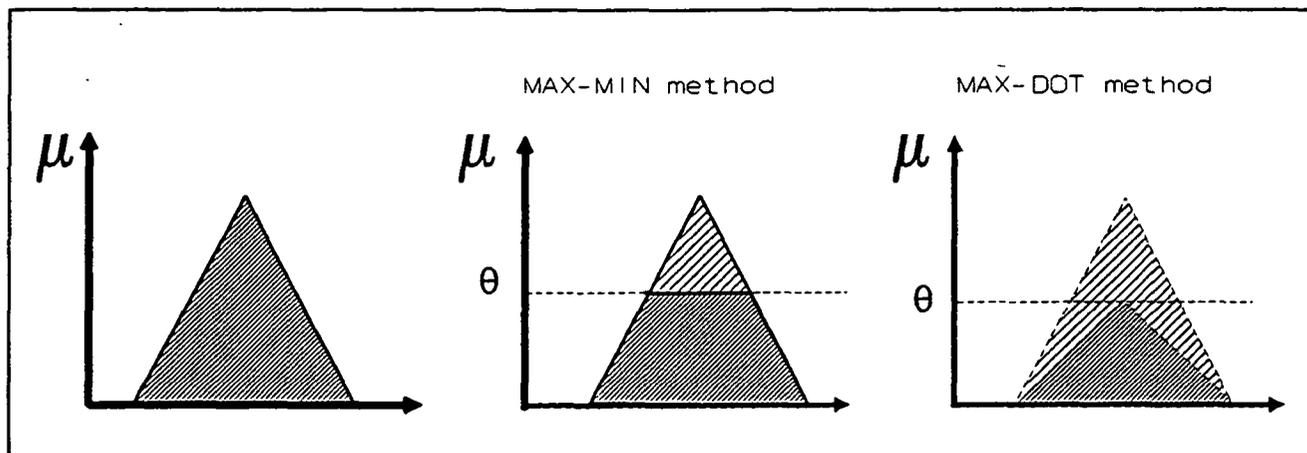


Figure 2

The Max-Min Inference is mainly adopted when the membership function is defined through a vectorial representation, in fact in this case it is relatively easy to compare each value of the M.F. with the threshold value and choose the smaller. The Max-Dot method it is not so easily performed because it is necessary to multiply by the scaling factor each non-zero component of the vector. Conversely, the Max-Dot inference method is preferred with the analytical representation, as it is easy to calculate the resulting M.F. by multiplying each breakpoint value by the scaling factor, whereas the Max-Min method requires a new series of breakpoints to be calculated.

There is a third method of representing the membership functions of the THEN-part in the particular case of Fuzzy Controllers, where the output of a Fuzzy inference is not used as input for another. In this case, the M.F. representation can be reduced to the only two parameters that are effectively needed in the assembling and defuzzification phase: a weight representing the area underlined by the M.F. and its point of application (barycentre). In fact the defuzzified output comes from a linear combination of those values, as clearly illustrated in the defuzzification algorithm generally adopted:

$$C^* = \frac{\sum_{i=1}^{MF_n} A_i' \cdot Xg_i'}{\sum_{i=1}^{MF_n} A_i'} \quad (1)$$

$Xg_i'$  = Barycentre of the  $i$  M.F. truncated at the  $\Theta$  truth level.

$A_i'$  = Area of the  $i$  M.F. truncated at the  $\Theta$  truth level.

$MF_n$  = Number of M.F.s of the output

The inferencing method chosen strongly influences the way in which the M.F. are assembled prior to the defuzzification phase. Essentially the two methods commonly adopted differ in the treatment of the zones of the universe of discourse where two or more M.F.s have non-zero values.

Fig. 3 shows the two approaches: in 3(A) the resultant M.F. is obtained by taking the greater of the two component values at any point whereas, in 3(B) the combined M.F. is obtained by simple addition of the component values. In effect, 3(A) represents a logical combination and 3(B) an arithmetical combination of the two M.F.s.

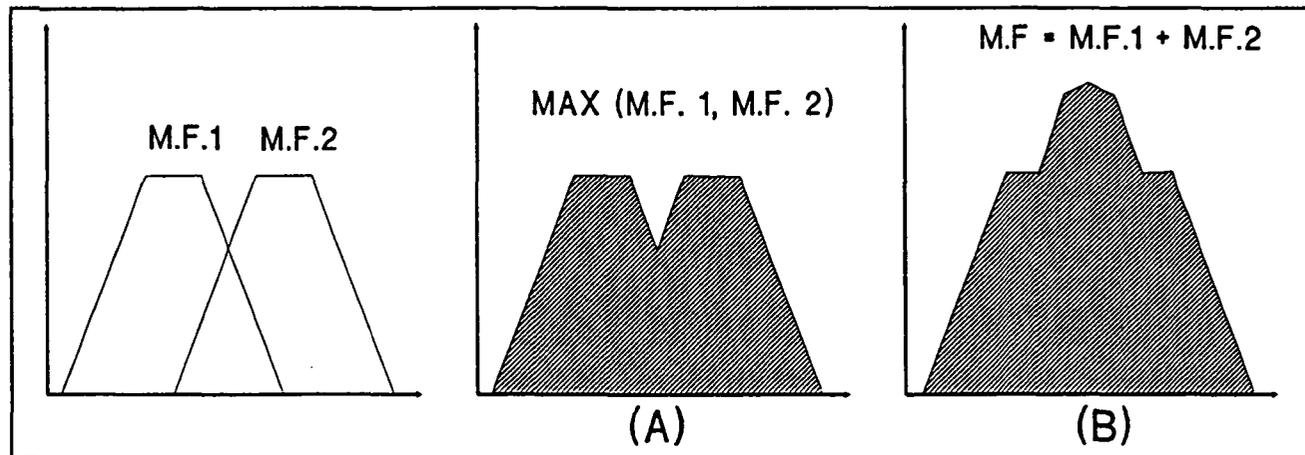


Figure 3

Depending on the method of representing the M.F.s it is possible to choose between the two above methods of assembling: only the arithmetical combination is allowed with the weight/barycentre representation while either of the two assembling methods can be chosen with the other two representations. However, the way in which M.F.s are to be represented and combined greatly affects the machine architecture, so these decisions must be made at an early stage in the design of a particular Fuzzy Machine.

It appears clear from what has been presented above, that an efficient general purpose fuzzy machine cannot exist but rather one must rely on machines tailored to meet the needs of a particular class of problems.

### Section 3. WARP: Weight Associative Rule Processor

WARP is a dedicated VLSI machine whose architecture has been designed in order to efficiently exploit all the advantages of Fuzzy calculus. The major innovation with respect of traditional approaches to Fuzzy Machines has been the adoption of different data structures for the various phases of the computational cycle. In fact one of the greatest limiting factor in the traditional fuzzy architectures is the use of the same data representation for both the computation connected to the IF-part and to the THEN-part of a rule.

In order to represent the membership functions connected to the Fuzzy variables of the antecedent of the rules we adopted a vectorial representation of the Membership Functions based on 64 ( $2^6$ ) or 128 ( $2^7$ ) elements, each possessing 16 ( $2^4$ ) truth levels.

The utilization of vectors for this phase of the Fuzzy calculus has the great advantage that in the case of a controller, for each rule the data involved in the computing are one or more M.F.'s (representing the knowledge of the system) and one or more crisp values (representing the input from the "external" world). With this data representation, in order to find the matching level (hereafter called  $\alpha$ ) between the input and the stored M.F.'s it is sufficient to get the various  $\alpha$  corresponding to the truth level of the element located by the projection of the input in the universe of discourse. Classically the vectors characterizing the membership functions of a term set are stored sequentially in the memory as illustrated in fig. 4.

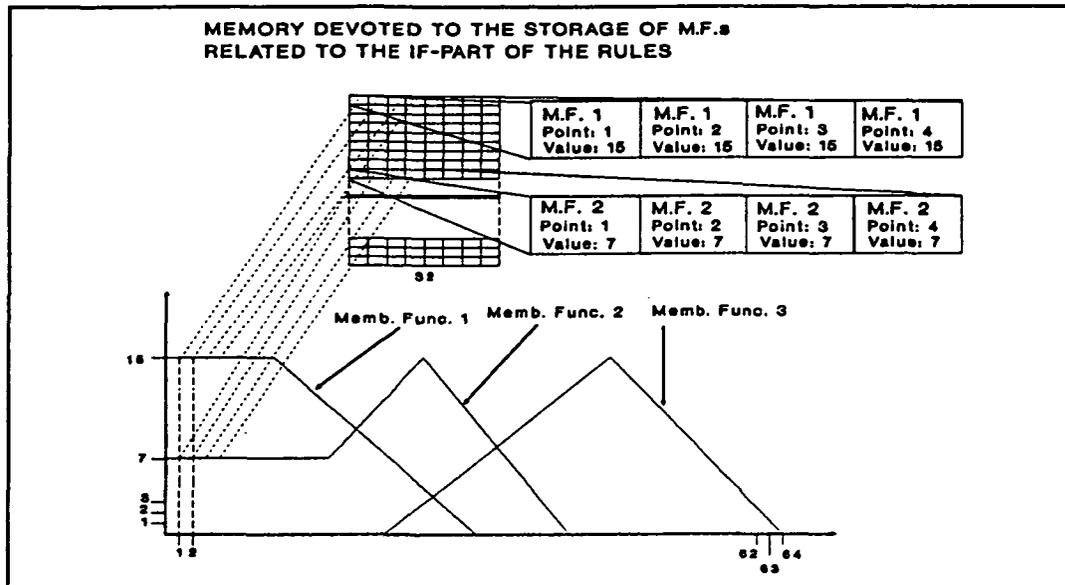


Figure 4

In this situation it is necessary to independently address each memory word containing a needed  $\alpha$  value. The number of memory accesses is thus a function of the membership functions comprising the term set. The memory access time being one of the most critical parameter of the computation, it appears clear that in order to obtain high performance the number of memory accesses must be reduced as much as possible.

In order to efficiently perform the computation of the IF-part of the rule, WARP architecture has been built up around a different idea for storing the membership functions. The WARP approach consists in storing in successive memory location of the same memory word all the  $\alpha$  values comprising a term set. This term set is formed by the membership functions connected to the IF-part of the rule, as showed by fig. 5. In this way it is possible to retrieve all the  $\alpha$  value of a term set using the crisp input value to calculate the memory word address in the fuzzy memory device utilized.

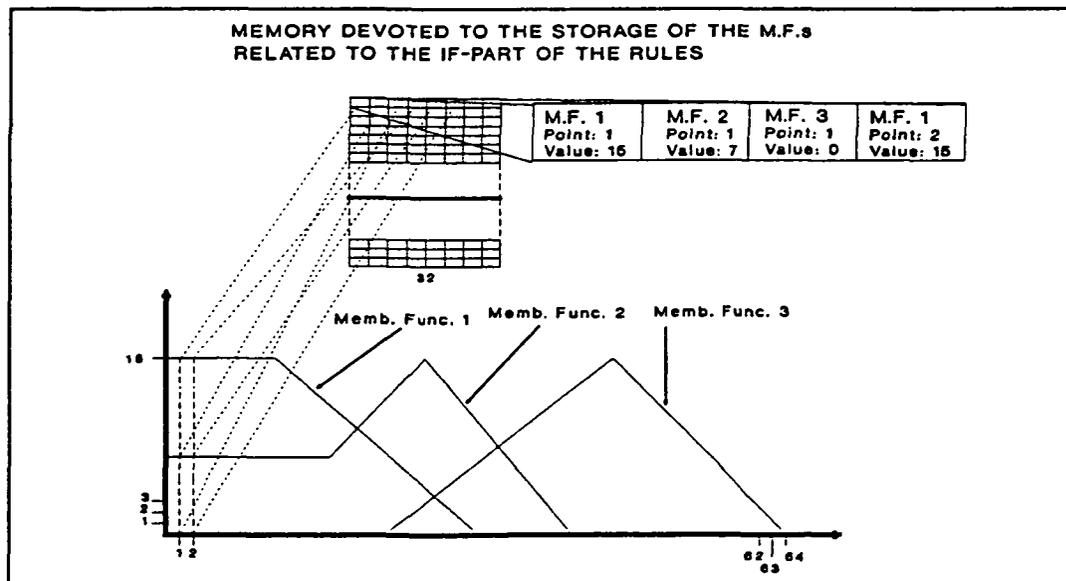


Figure 5

The number of memory accesses is a function of the M.F.s comprising a term set and inversely proportional to the size of the memory word, obtaining a significant reduction of the number of access in comparison with the traditional information storage methods. Assuming memory words with the same width (32) and elements of the vectors with the same characterization (4) the number of accesses is reduced by a factor of 8 (32/4).

Although the illustrated method for storing and retrieving the various  $\alpha$  values connected to a fuzzy variable is highly efficient, once the related  $\Theta$  value (the truth level for modifying a variable of the THEN-part) has been calculated, the vectorial computation becomes slow due to the huge time consuming process of modifying the M.F.s of the right side of the rule with the threshold value provided and assembling all the M.F.s that will form the M.F. furnished as output. Taking in account some limiting factors like:

- The number of parallel computational elements that realistically can comprise such a device
- The linear increase in memory size when trying to augment the number of elements which characterize a M.F
- The necessity to cycle over all the elements of the M.F. provided as output in order to carry out the defuzzification phase

It is clearly apparent how inefficient is such information management.

WARP avoids the above limitations. Having a limited number of possible truth values (15 excluding 0) coming from the IF-part of a rule, it is possible to represent a membership function connected to the THEN-part utilizing 15 words of memory, each containing both the value (weight) of the area underlined by the M.F. and the point of application (barycentre). In order to achieve a more efficient computation, for each memory word characterizing a truth level WARP directly stores both the area multiplied with the barycentre and the area itself, as illustrated in fig. 6.

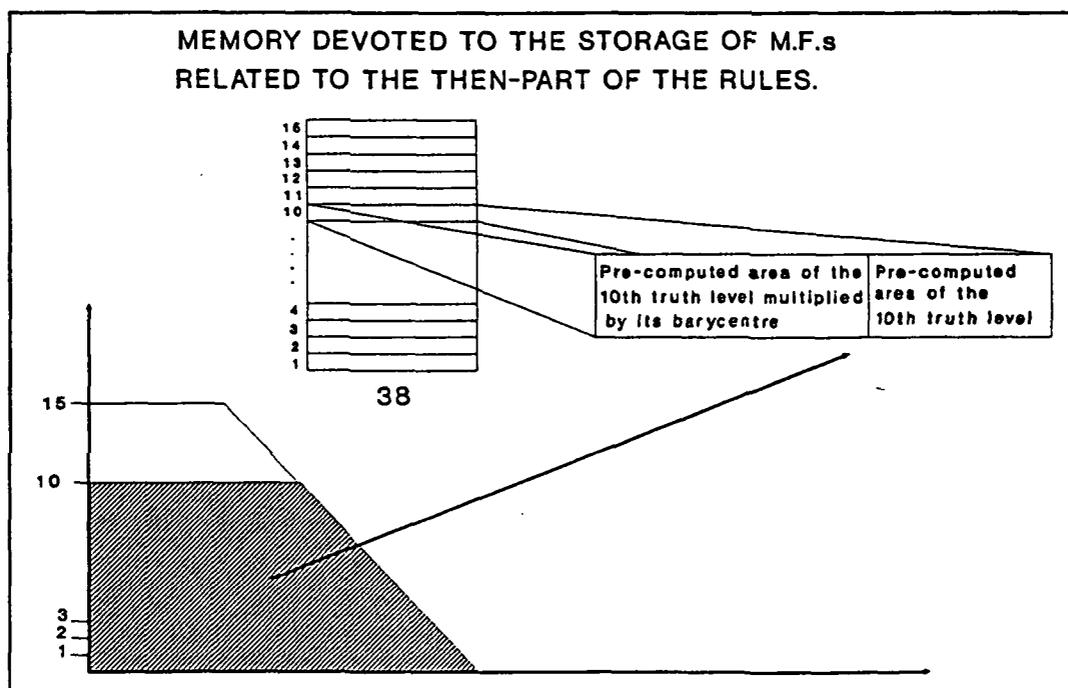


Figure 6

With such a method for storing information, the inferencing method adopted (Max-Min or Max-Dot) is perfectly transparent with respect of the computational architecture, in fact the only difference between those methods lies in the different value of the area of the resulting M.F. as clearly illustrated in fig. 2.

A great computational advantage of the approach is that a great part of the fuzzy computing can in effect be performed off line. The particular data structure adopted in WARP for representing the M.F.s allows an assembling methods of type (B) with reference to fig. 3.

WARP is a VLSI Megacell whose architecture has been designed in order to be employed in different environments. The dedicated memories and the computing blocks have been defined with the purpose of efficiently operating with a representation of the membership functions as previously illustrated. The architectural data flow/block diagram of the Fuzzy megacell core is shown in Fig. 7.

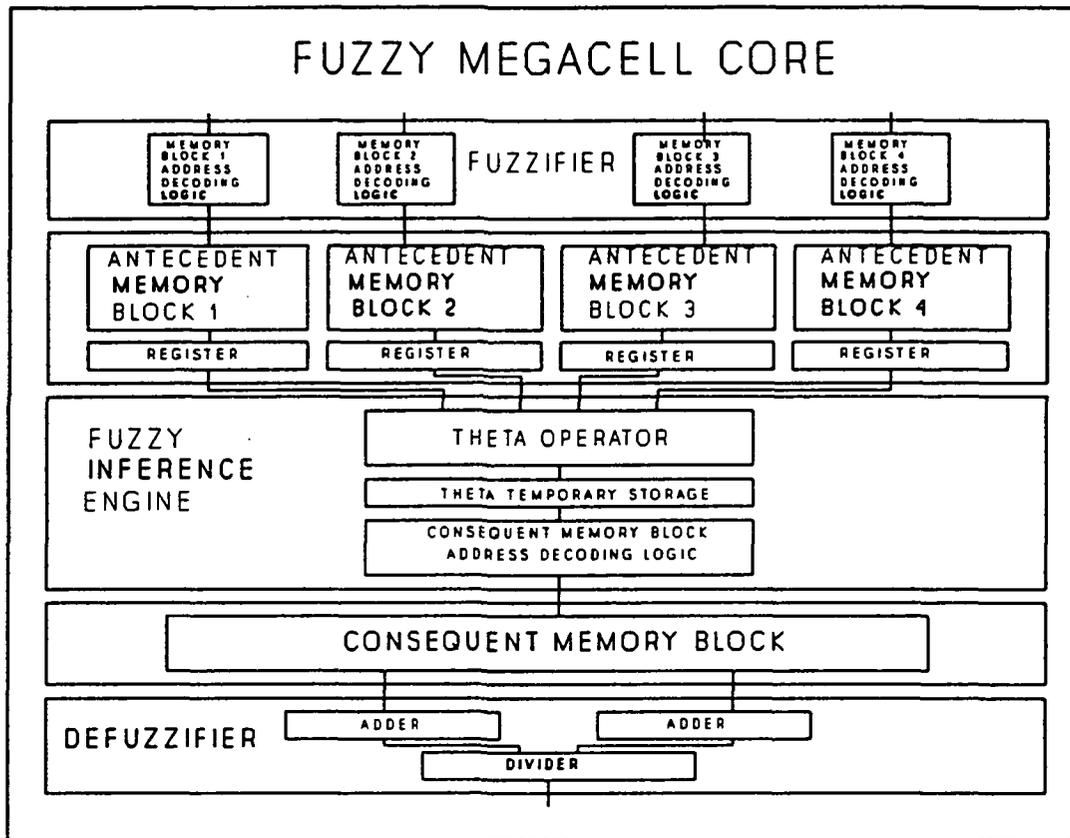


Figure 7

The Fuzzifier section is devoted to the calculus of the memory address corresponding to an input and the retrieving of the stored information. The assumption of always expecting as input a crisp value combined with the particular storage method has allowed the fuzzifier to be reduced to its simplest structure.

To obtain high performances the memory devoted to the storing of the membership functions of the IF-part of the rules has been divided in 4 independent blocks. Each of these blocks contains all the  $\alpha$  values of one or more fuzzy variables, allowing the parallel retrieval of the  $\alpha$  values. Inside the memory block, the data representing the membership functions are stored according to the scheme of section 3.

This splitting of the memory has also induced the necessity of also having 4 fuzzifier sections (one for each memory block). The  $\alpha$  values found are memorized in a set of devoted register and then opportunely processed to calculate the  $\theta$  value of each rule.

The adoption of the vectorial data representation for the M.F.s of the IF-part of the rules allows this operation to be performed in a highly efficient and flexible way inside the Fuzzy Inference Engine via the Theta-operator, whose block diagram is illustrated in fig. 8. This operator has been designed in order to carry out operations with an unlimited number of terms connected by OR and/or AND connectives. This block is utilized mainly to augment the performances of the device, in fact practically all the Fuzzy computing is performed here (the defuzzification although computationally heavy cannot be properly classified as fuzzy computing).

The  $\theta$  values are used to calculate the address of the memory word in the memory block where the membership functions bounded to fuzzy variables of the THEN-part of the rules are stored. Inside this memory block, the values of the M.F.s are stored with the technique illustrated in fig. 6.

The memory block devoted to the fuzzy variables of the THEN-part of the rules has not been divided because the computational requirements and the architectural simulations have clearly shown that the addition of dedicated hardware is not balanced by a significant increase in performance.

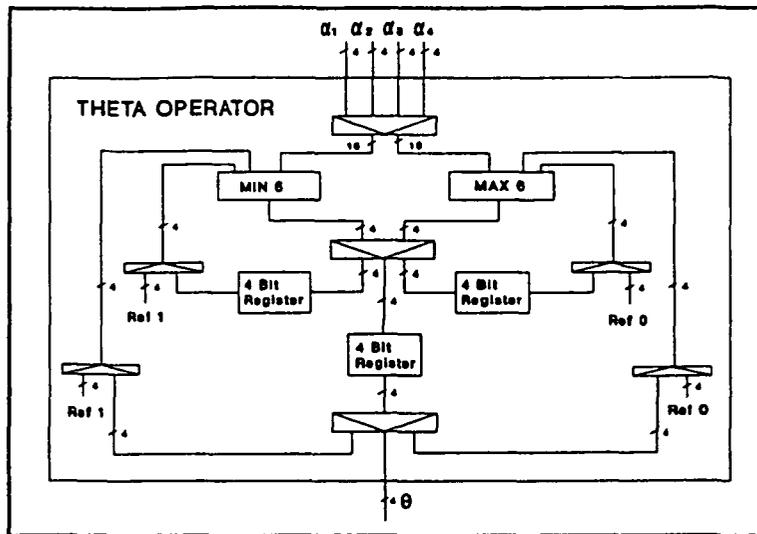


Figure 8

The assembling of all the membership functions comprising an output and the defuzzification process are carried out in the Defuzzifier block. Thanks to the particular representation of the membership functions, this phase can be performed with a limited number of operations. The studied architecture utilizes 15 memory words, each 38 bit wide, to store the relevant information of each M.F. Having adopted the defuzzification algorithm previously illustrated in section 2, a saving of 2 or 3 multiplying operations is obtained (actually those necessary to calculate  $A_i'$  and  $A_i' \cdot X_{g_i}'$ ) with related hardware and, most of all, a great freedom in defining the M.F.s themselves is allowed. In fact in this way a membership function doesn't need to be symmetrical as would be the case if it was described giving only the whole weight and its barycentre. With the adopted method each truth level is characterized by the actual weight and its point of application thus effectively overcoming any constraint related to symmetry.

The Fuzzy megacell can be employed in different environments. The ST9 microcontroller thanks to its flexible architecture is well suited to being augmented as in the configuration illustrated in fig. 9.

In this way the microcontroller can perform normal control task while WARP will be responsible for all the fuzzy related computing in independent mode.

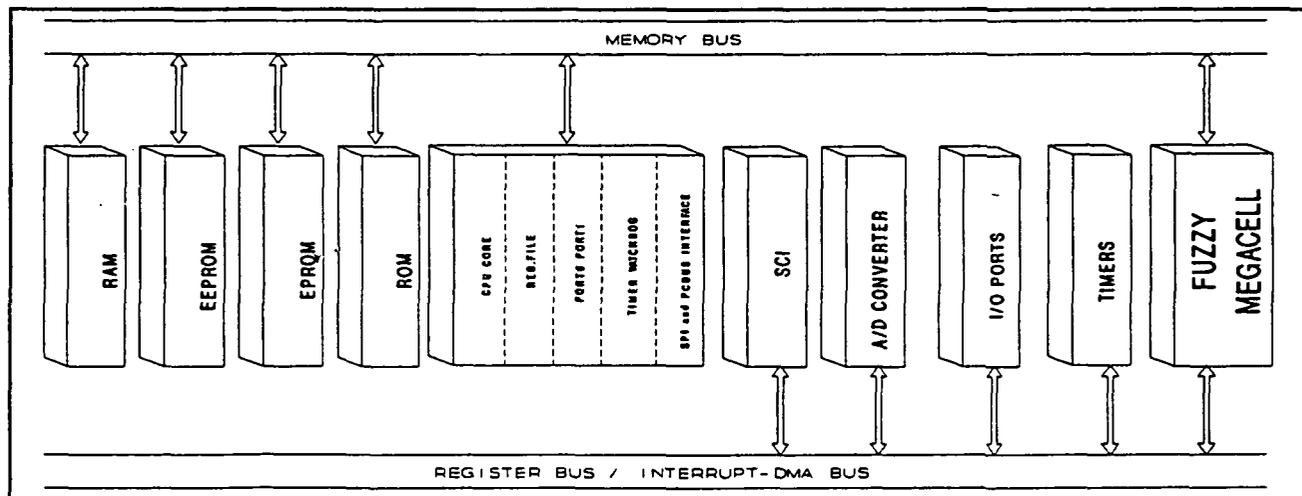


Figure 9

The Fuzzy megacell can also be configured as an embedded controller in a configuration as the one illustrated in fig. 10.

WARP is currently in the advanced design phase. In order to guarantee high compatibility with customers needs and assure maximum flexibility, a TOP-DOWN design methodology has been adopted for it and the VHDL language to implement it. VHDL (VHSIC, or Very High Speed IC, Hardware Description Language) is the IEEE standard language for the description and simulation of electronics circuits. WARP hardware structures have been synthesized utilizing SGS-THOMSON's own 0.8  $\mu\text{m}$  technology. The subsequent structural simulations have displayed performances in the order of 10 MFIPS.

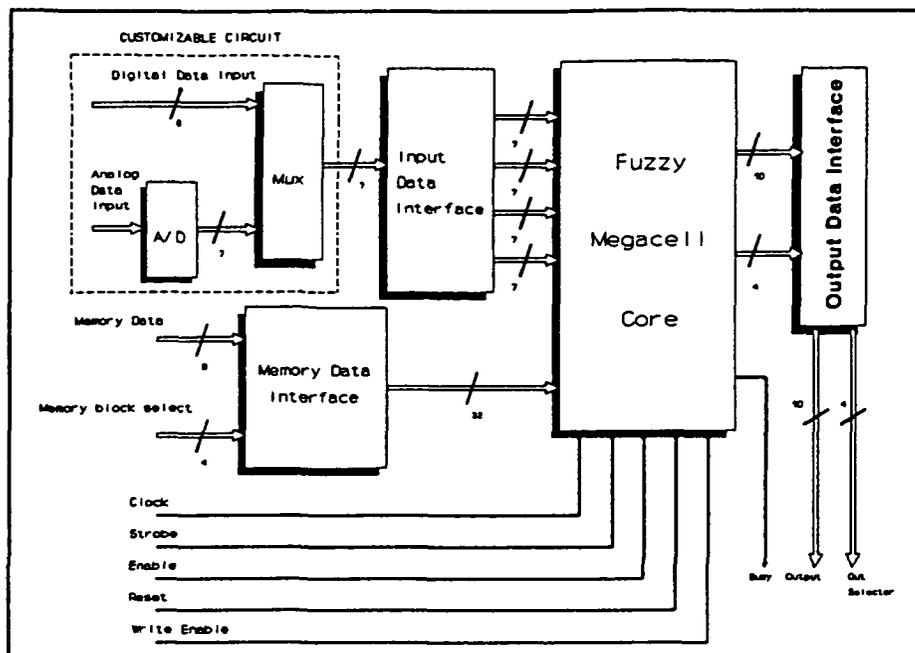


Figure 10

## Section 5 Conclusions

In order to provide an answer to a wide number of application requests, WARP design relies on concepts of flexibility and modularity. The innovative approach of WARP is represented by the adoption of different data structures to represent the membership functions characterizing the fuzzy variables of the left and right sides of the rules. Great emphasis has been put on granting the user maximum flexibility in defining the membership functions. This has been carried out allowing the definition of Term Sets with no fixed numbers of fuzzy sets; moreover the possibility of defining the single membership functions without any constraint like symmetry/shape proved very useful in characterizing complex control applications. The careful analysis of the computational requirements during the various stage of the fuzzy processing and the subsequent mapping in adequate hardware structures has lead to the achievement of high level of computational efficiency permitting performance in the order of 10 MFIPS to be obtained while reducing the number of parallel computational elements. Moreover the architecture is totally transparent with respect of the types of memory utilized (EEPROM, Flash ...) and technology (Sub- $\mu$  CMOS, ...) so allowing the device to be used for a wide range of applications.

## Section 6. References

- [1] L. A. Zadeh, "Fuzzy Logic", Computer, April 1988
- [2] H. Watanabe, W. D. Dettloff and K. E. Yount, "A VLSI Fuzzy Logic controller with Reconfigurable, Cascadable Architecture", IEEE Journal of solid-state circuits. vol 25, No. 2, April 1990.
- [3] H. Ikeda, N. Kisu, Y. Hiramoto and S. Nakamura, "A Fuzzy Inference Coprocessor Using a Flexible Active-Rule-Driven Architecture", pp. 537-544, IEEE International Conference on Fuzzy Systems 1992
- [4] T. Yamakawa, "Intrinsic Fuzzy electronic circuits for sixth generation computer" in Fuzzy Computing, M.M Gupta and T. Yamakawa, pp. 157-171, Elsevier Publisher B.V. (North Holland), 1988
- [5] A. Pagni, R. Poluzzi and G.G. Rizzotto, "WARP: Weight Associative Rule Processor. An innovative Fuzzy Logic Controller", pp. 543-546, Proceedings of IIZUKA'92 2<sup>nd</sup> International Conference on Fuzzy Logic and Neural Networks, 1992

## EVALUATION OF FUZZY INFERENCE SYSTEMS USING FUZZY LEAST SQUARES

p. 10

Joseph M. Barone  
Loki Software, Inc.  
P.O. Box 71  
Liberty Corner, NJ 07938 USA

## SUMMARY

Efforts to develop evaluation methods for fuzzy inference systems which are not based on crisp, quantitative data or processes (i.e., where the phenomenon the system is built to describe or control is inherently fuzzy) are just beginning. This paper suggests that the method of fuzzy least squares can be used to perform such evaluations. Regressing the desired outputs onto the inferred outputs can provide both global and local measures of success. The global measures have some value in an absolute sense but are particularly useful when competing solutions (e.g., different numbers of rules, different fuzzy input partitions) are being compared. The local measure described here can be used to identify specific areas of poor fit where special measures (e.g. the use of emphatic or suppressive rules) can be applied. Several examples are discussed which illustrate the applicability of the method as an evaluation tool.

## INTRODUCTION

Smith and Comer [1] point out that evaluation of the behavior of a fuzzy system can be quite difficult. They also mention (p. 20) that the qualitative knowledge of the controller designer is more suited to accurate specification of the antecedent portions of the control rules than to accurate specification of the consequent portions. This is because (presumably and at least in part) the role of the input variables in system dynamics is more easily understood in general, and also because the input variables are often more directly and more easily expressible in fuzzy (linguistic) terms (e.g., temperature as high, medium, and low). This is perhaps even more true in "softer" areas like psychology and sociology, where "harder" inputs like age and socioeconomic status are used to control (predict) softer outputs like behavior or risk (for interesting comments along these lines in the context of fuzzy classification see [2]). In fact, the very foundations of some methods of analysis and prediction used in these soft areas, especially classical least squares, are predicated upon input variables whose values are assumed to be error-free measurable (see e.g. [3], Section 1.1).

Methods for the evaluation and tuning of fuzzy systems do not really challenge this assumption; they typically assume that the designer has the input distributions about right and then adjust formal "parameters" of the inference mechanism to improve controller performance. Again, this works well in hard areas but should prove difficult to apply in emerging softer applications where there is no aspect of the inference process that can be trusted completely. It becomes important, therefore, in soft applications, to have some way of evaluating the accuracy and effectiveness of a fuzzy inference system which assumes as little as possible about the validity of the rules, and even of their essential characteristics, beyond the linguistic properties they express. Furthermore, there may often be no real way of knowing whether interpolated consequent fuzzy values (values not supplied directly by an expert) are accurate to the point where they can serve to confirm the chosen system and parameters. It should prove useful, therefore, to have available methods which can provide overall evaluation measures given certain assumptions about the structure and regularity of the output (consequent) fuzzy distributions.

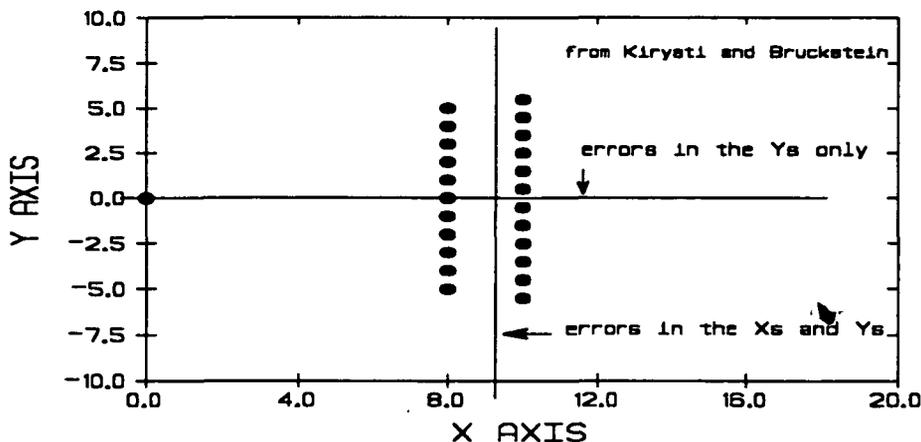
Perhaps the most well-characterized and formalized methods for the evaluation and tuning of fuzzy controllers are those based on the concept of cell mapping [1, 4-5]. Nonetheless, the application of cell mapping to evaluation and tuning depends crucially on the existence of sufficient crisp input-output pairs to generate the cell maps (actually, this is a bit of an oversimplification - see [5], pp. 749-750), and also provides no real way to distinguish between competing fuzzifications of the input state space (unless of course the fuzzification is so bad that tuning is impossible). This paper suggests that an evaluation based on fuzzy least squares can indeed distinguish between competing input state space fuzzifications and can be used (quite easily) in cases where neither the input nor the output is readily defuzzifiable.

## FUZZY LEAST SQUARES

The method of fuzzy least squares was introduced by Diamond [6] as an approach to the fuzzy regression problem, i.e. as a method for parameterizing the relationship between two sets of fuzzy numbers; its advantage over other techniques (besides computational simplicity), as Diamond points out, is the amenability of the parameterization to evaluation by standard measures, e.g., examination of residuals. For the purpose at hand, it is particularly important that the spatial geometry of the fuzzy least squares method be understood; to accomplish this goal, we turn briefly to crisp models.

Basically, the solutions to linear parameter estimation problems as well as their computational simplicity depend heavily on assumptions regarding which measurements may be considered to be error-free and which measurements may not. If either the independent or the dependent variable measurements are taken to be error-free, then ordinary least squares may reasonably be applied to the data. In such cases, the error (residual) vectors are orthogonal to the axis (or axes) along which the error-free values are measured. If, on the other hand, both dependent and independent variable measurements must be assumed to be made with error, the parameter estimation problem becomes considerably more difficult (even analytically intractable in the general case). In any case, if a solution can be generated, the error vectors will be orthogonal to the fitted line itself (the first chapter of [3] contains an excellent summary and relevant examples).

In extreme cases, especially those in which the data points are contaminated by outliers, the differences in the various solutions may be striking, as is illustrated in the figure below (from [7]). If the  $x$  coordinates are assumed to be error-free and a line is fitted by the method suggested in [7] (not ordinary least squares but equivalent for the present purpose), then errors orthogonal to the  $x$  axis are minimized by a fitted line which passes through the outlier (the point at 0,0). This is clearly a most undesirable solution. If both the  $x$  and  $y$  coordinates are assumed to contain errors, on the other hand, (even isotropic ones), the method yields a much more reasonable fitted line (the one parallel to the  $y$  axis).



To return to fuzzy considerations, the point is that the method of fuzzy least squares, despite its "ordinary least squares" character, is more closely related (in spirit, as it were) to fitting (regression) approaches in which both dependent and variables are measured with error. It should be emphasized, however, that this is not true from a purely analytic point of view. Once a distance metric is decided upon, and once the hypergeometric characteristics of the set of triangular fuzzy numbers are established, the fuzzy least squares parameter vector is derived by an orthogonal projection of the dependent variable vector onto the "cone" of potential solutions exactly as in ordinary least squares (Diamond's paper [6], pp. 142-146 contains an elegant exposition of these facts, and section 2.3 of [3] contains highly instructive comments and diagrams in a crisp context). Thus, from an analytical point of view, though both the independent and dependent variable vectors are fuzzy, one is assumed to be measured without error while the other is not.

From another point of view, however, fuzzy least squares is more like a "total least squares" approach [3] in which both the dependent and independent vectors (or matrices) are assumed to be measured with errors. This is because the fuzzy least squares method, with its two separate spatial components (mode and spread), permits the search for a solution vector to move about a more complex (and hence more flexible) space (in effect, of course, since the solution is derived analytically). The result of this is that fuzzy least squares can preserve an extremely good fit in fuzziness even if, for some reason, one or more values in the data are outliers relative to mode. Since the fuzziness of the dependent and independent variables, taken together, are a measure of the overall uncertainty of the system, this characteristic has the effect of preserving the degree of overall uncertainty in a manner similar to total least squares methods.

## FUZZY LEAST SQUARES AND FUZZY INFERENCE

It would surely be instructive to pursue the analogy between fuzzy least squares and total least squares further and more formally, but that would take us far beyond the scope of this paper. It is worth mentioning, though, by way of leaving the previous topic and beginning the current one, that Diamond's fuzzy least squares minimization condition (1) could conceivably be replaced to good effect by (2), where minimization of the square of the distances between the measured ( $Y_i$ ) and calculated ( $E + bX_i$ ) is replaced by minimization of some scalar norm of the "total error" matrix ( $\Delta$ ) and where  $X_0$  is the unobservable "true" vector of fuzzy predictors (see [3], p.186 and p. 23). If the Frobenius norm were

$$\sum d ( E + b X_i , Y_i )^2 \quad ( 1 )$$

$$F ( E , b , X_0 ) = [ d ( X , X_0 ) ; d ( E + b X , Y ) ] \quad ( 2 )$$

used, a solution to (2) would be equivalent to a solution of the "fuzzy total least squares" minimization function (cf. [3], p. 186).

Be all of this as it may, it seems fair to conclude that fuzzy least squares is a relatively "robust" form of regression which is eminently suitable for parameterizing the relationship between two n-dimensional fuzzy vectors with elements of regular shape (at least triangular and trapezoidal [6]). The vectors being compared do not necessarily have to be particularly "linear", though they must at least be "coherent" ([6], pp. 150-151); vectors produced as result of fuzzy inference are as likely to be coherent as not, one would imagine, but the condition is easily tested for [6], so inference systems which do not produce coherent output should simply not be subjected to the evaluation procedures suggested here.

Fuzzy least squares, then, forms the basis for a simple evaluation technique for fuzzy inference systems. Given two possible solutions, regress the known (fuzzy) output (the "correct" values) on the output fuzzy sets generated by the two inference processes. Compare the two solutions via any of many available evaluation methods, and keep the one which evaluates higher. Certain evaluation methods may even suggest ways in which the better solution can be improved. Space does not permit further general discussion, so we conclude by introducing a few evaluation measures and by providing examples of their use. It is worth noting at this point that the calculations needed to perform fuzzy least squares and to compute the evaluation measures are straightforward and can be performed with minimal computational overhead. It is also worth noting that it is may be possible to extend the domain of this method to inference systems which do not produce fuzzy "numerical" output by "fitting" fuzzy numbers over the fuzzy sets by linear interpolation as is done in fuzzy modeling (see, e.g. [8]), but this matter will not be pursued here.

## EVALUATION MEASURES

1. GLOBAL MEASURES. The most obvious global measure of success are the least squares residuals. A related value which varies conveniently between 0 (no correlation) and 1 (perfect correlation) is the correlation coefficient. For generality, we define (see [9], p.280) the fuzzy multiple correlation coefficient (MCC) as

$$MCC = \left( \frac{\sum d(E + bX_i, \bar{Y})^2}{\sum d(Y_i, \bar{Y})^2} \right)^{1/2} \quad (3)$$

where  $d$  again is the distance between two fuzzy numbers [6] and where  $\bar{Y}$  is the mean dependent fuzzy value, even though all examples in this paper are univariate and extensions to the multivariate case are non-trivial ([6], p. 156).

Another useful global measure of success is the relative entropy of the fuzzy least squares solution as defined in [10]. This form of relative entropy is a measure of the success of the regression "line" in tracking the fuzziness of the elements of the dependent variable vector. It is defined as (see [10] for a detailed description and rationale):

$$h_r = - \left\{ \sum_{i=1}^n \left[ \mu_e(y_i) \ln(\mu_e(y_i)) + (1 - \mu_e(y_i)) \ln(1 - \mu_e(y_i)) \right] \right\} \quad (4)$$

$$\text{where } \mu_e(y_i) = \max(0.5, \min(\frac{\text{spread}(y_i)}{\text{spread}(\hat{y}_i)}, \frac{\text{spread}(\hat{y}_i)}{\text{spread}(y_i)}))$$

and where  $\hat{y}_i$  is the estimated  $y_i$  (i.e.,  $E + bX_i$ ).

2. LOCAL MEASURES. The only local evaluation method discussed here will be the weighted squared standardized distance [11-12]. In the univariate case, the WSSD can be written as:

$$WSSD_i = \frac{(n-1)b^2 \cdot d(x_i, \bar{X})^2}{\sum_{j=1}^n d(y_j, \bar{Y})^2} \quad (5)$$

where  $\bar{X}$  and  $\bar{Y}$  are the  $X$  and  $Y$  means

where  $b$  is the regression coefficient, and where  $d$  again is the fuzzy distance. In ordinary least squares regression, the magnitude of  $WSSD_i$  is used to determine whether or not point  $i$  is a "high-leverage point", i.e., a point in a sparse region of the  $X$ -space (see, e.g., [12], pp. 94 ff.). We are interested here in the WSSD because a fuzzy inference tends to produce similar or identical output when the inference mechanism operates near the centers of the involved fuzzy sets and to produce rapidly changing output as the inference mechanism operates near areas of overlap (and thus near areas of heavy interpolation). A good inference mechanism should produce transitional areas in its output which correspond to areas of overlap in the output data partition. Thus, the output vector produced by a fuzzy inference should have clusters of similar or identical values which match the reference values near the centers of the elements of the output reference partition, and rapid changes in value which match the reference values in and near the overlap areas of the output reference partition. This phenomenon will produce clusters of points with similar or identical leverage in the regression followed by points with unique leverage values (at the transitional areas). In a good model, then, the clusters and transitions in WSSD values will line up nicely with the centers and overlap areas of the output reference partition respectively.

#### A NOTE ON "PIECEWISE" APPROXIMATIONS

It is important to note that this paper is not suggesting that fuzzy least squares is to be used to construct an accurate "piecewise" approximation to some unknown "functional relationship" between input and output fuzzy sets. To understand better what is being suggested, consider a fuzzy Lagrangian interpolation polynomial which relates the true output fuzzy sets and the ones generated by the inference (as in [13] with  $n+1$  fuzzy points). As with crisp Lagrangian interpolation, such a polynomial could be used, for instance, to compute error bounds (using contour integrals in the complex plane [14]) if we knew the "true" functional relationship between the actual output fuzzy sets and the generated ones; such a relationship may not exist, of course, in the general case

and in the usual sense of the word functional, but would in any event depend on the accuracy of the inference process as an interpolator and "smoother". The least squares regression line, then, serves in this context as a crude continuous approximation to some presumably nonlinear and possibly unrecoverable "difference" function.

## EXAMPLES

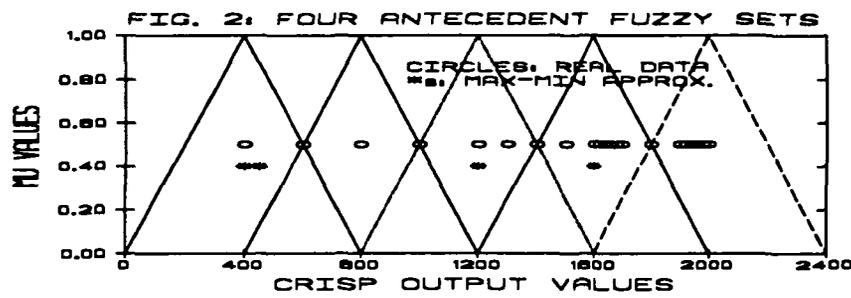
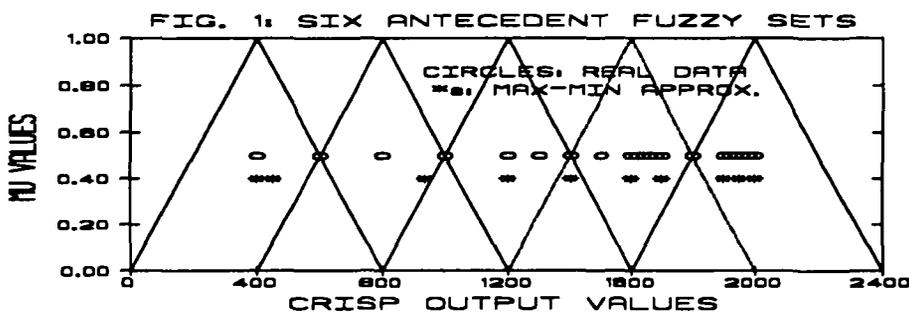
All of the examples discussed here are based on examples from Cao and Kandel [15]. Since their examples are based on crisp input and output (rotational speed of a d.c. series motor as a function of input current), the output was "refuzzified" as described below so that fuzzy regression could be applied. The data sets in our examples, therefore, are not "inherently" fuzzy, but they do have the advantage of being associated with thoroughly analyzed models which are easy to evaluate for accuracy (in a crisp sense). Note also that as was mentioned earlier, the notion of "reshaping" the output of a fuzzy inference process so that it can be analyzed by fuzzy least squares is not an unreasonable one, though of course for useful application it would require more elaborate methods than the one used here.

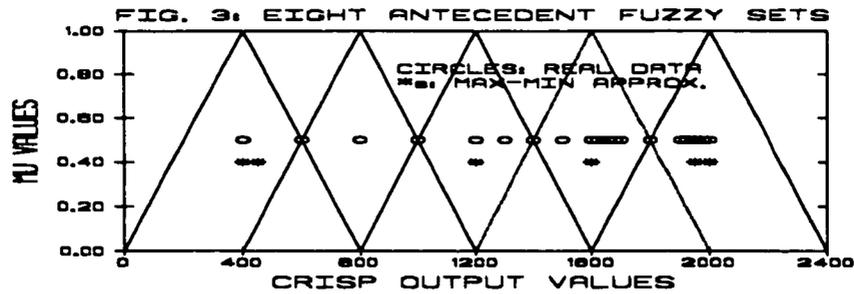
1. The "model" curve of Example 7 in [15] is a connected piecewise linear curve of five segments with overall rising trend. The model curve is "covered" by the five overlapping fuzzy sets shown in Figure 1. Assuming that this consequent set representation is reasonable, the fuzziest areas of coverage (i.e., the areas of maximal overlap) are those around the output values 800, 1400, and 1800 (800 because the rules do not reference the second set (SMALL)). Ideally, the inference system should map the corresponding input values (1.0, 3.0, and 7.0) into these same transition areas. Cao and Kandel cover the input range by six overlapping fuzzy sets; we use the WSSD, MCC, and relative entropy to compare their six input set partition with a four input set partition and an eight input set partition. The rules in the four and eight input set cases are adjusted to conform insofar as is possible with the content of Cao and Kandel's original (six input set) rules. The crisp output data and the crisp inferred output data were fuzzified by using 10%, 15%, or 20% of the mode as the left and right spreads, increasing the percentage as the numbers got larger; in this manner, a reasonably coherent output data set and inferred output data set were constructed. The rules themselves are as follows (in each case the input domain is distributed equally among the component sets):

NULL -> ZERO	NULL -> ZERO	NULL -> ZERO
ZERO-SMALL -> MEDIUM	ZERO -> MEDIUM	ZERO -> MEDIUM
SMALL-MEDIUM -> LARGE	SMALL -> LARGE	ZERO-SMALL -> MEDIUM
LARGE -> VERY LARGE	MEDIUM -> LARGE	SMALL -> LARGE
	LARGE -> VERY LARGE	SMALL-MEDIUM -> LARGE
	VERY LARGE -> VERY LARGE	MEDIUM-LARGE -> VERY LARGE
		LARGE -> VERY LARGE
		VERY-LARGE -> VERY LARGE

As Table 1 shows, good results were obtained when the fuzzified inferred values were regressed on the fuzzified output data (the table shows only the crisp values, i.e., the modes of the fuzzified values). The transition points match nicely, the MCC is high, and the relative entropy is low (of course, the MCC and entropy values are most meaningful when compared with other prospective solutions). When only four antecedent sets are used, however, the results suffer dramatically. The transition points miss the mark by a considerable margin, the MCC is lower, and the relative entropy is higher. With eight antecedent sets results are better but still not as good as with six (it is important to note here that overlap was retained at 50%). If one had started with the four or eight set inference machine, the lack of matchups in the transition areas would have been a clue that the results could be improved upon. It is worth noting that the relative magnitudes of the fuzzy constants are a decent guide to the relative merits of the various models. Figures 1, 2, and 3 show the distributions of the crisp output values relative to the output set and to the covering fuzzy sets (the fuzzy partition) for the consequent portions of the inference rules; note that only the six antecedent set solution produces distinct transition values in the vicinity of the transition regions of the output fuzzy partition, and that this fact is reflected in the WSSD values. For details of the membership functions, input and output data, and the rules themselves refer to [15].

TABLE 1: RESULTS FOR EXAMPLE 7 OF CAO and KANDEL WITH MAX/MIN INFERENCE						
ASTERISKS MARK TRANSITION POINTS						
DATA	6 ANT. SETS	WSSD FOR 6	4 ANT. SETS	WSSD FOR 4	8 ANT. SETS	WSSD FOR 8
2000	2000.0	1.1587	1600	1.0106	1950	0.8653
1980	1950.0	0.9439	1600	1.0106	1950	0.8653
1960	1900.0	0.7513	1600	1.0106	2000	1.0614
1940	1950.0	0.9439	1600	1.0106	1950	0.8653
1920	2000.0	1.1587	1600	1.0106	1950	0.8653
1900	1950.0	0.9439	1600	1.0106	2000	1.0614
1800*	1700.0	0.2024*	1200	0.0129	1950	0.8653*
1700	1600.0	0.0608	1200	0.0129	1600	0.0569
1680	1600.0	0.0459	1200	0.0063	1600	0.0432
1660	1600.0	0.0459	1200	0.0063	1600	0.0432
1640	1600.0	0.0459	1200	0.0063	1600	0.0432
1620	1600.0	0.0459	1200	0.0063	1600	0.0432
1600	1600.0	0.0459	1200	0.0063	1600	0.0432
1500	1600.0	0.0459	1200	0.0063	1200	0.3541
1400*	1400.0	0.0446*	1200	0.0063	1200	0.3541
1300	1200.0	0.4275	1200	0.0241	1200	0.3847
1200	1200.0	0.4275	1200	0.0241	1200	0.3847
1000	1200.0	0.4275	450	2.6917	1200	0.3847
800*	937.50	1.4192*	450	2.6917	1200	0.3847
600	450.00	4.8495	400	3.0746	450	4.3971
400	400.00	5.3181	400	3.0746	400	4.8227
MCC		0.9837		0.9411		0.9537
R. ENT		3.7707		5.8724		4.4829
FOR 6 ANTECEDENT SETS $Y = (20.4, 3.82, 3.82) + .99X$						
FOR 4 ANTECEDENT SETS $Y = (248.0, 40.50, 40.50) + 1.08X$						
FOR 8 ANTECEDENT SETS $Y = (90.6, 15.32, 15.32) + .95X$						

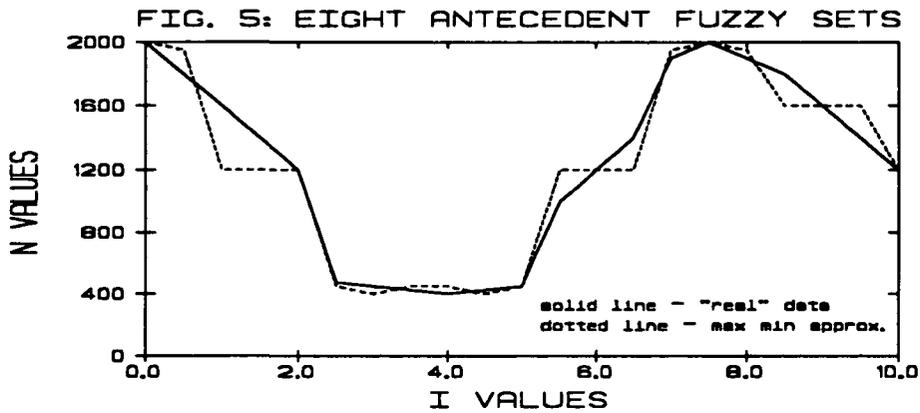
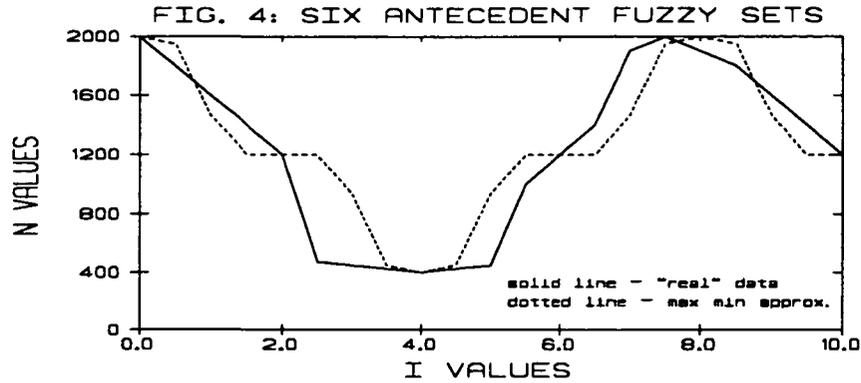




2. The model curve of Example 3 in [15] is a piecewise linear curve with two trend shifts. For this example, we flattened the bottom and shifted the second peak left to conform with the output of an eight antecedent fuzzy set approximation. As can be seen from the results below (and as would be expected), the eight-antecedent model yields better statistics. Nevertheless, the six-antecedent model conforms better to the transition points (not shown, but fairly obvious from an inspection of Figures 4 and 5). This suggests that the flattened area might be better approximated by emphasizing the appropriate rule in the rule set [15] and retaining the six antecedent fuzzy sets (note that to do this it is necessary to switch from max-min to product-sum inference - see [16]). As can be seen from the third column of values in Table 2, this hypothesis proves correct - there is little difference between the eight-antecedent results and the six-antecedent results with emphasis, and the six-antecedent version is truer through the transitions. If the second peak is shifted back to its original spot, in fact, the six-antecedent version with emphasis is better on all statistics. Note again that the magnitude of the fuzzy constant is a good indication of the relative merits of the various models. The rules are as follows:

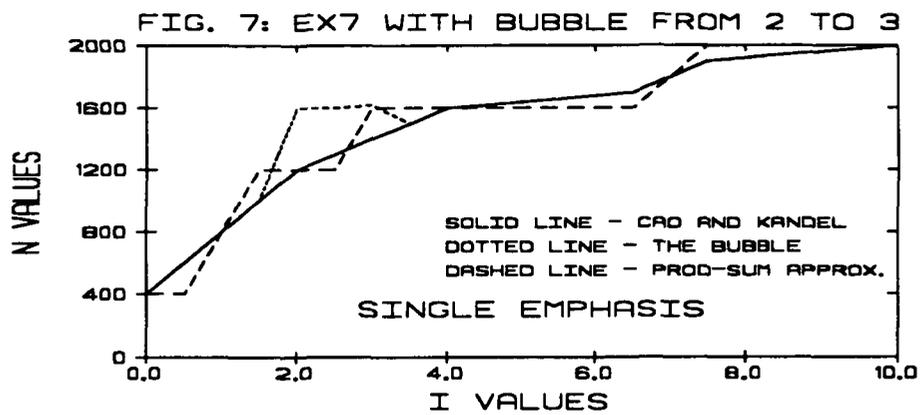
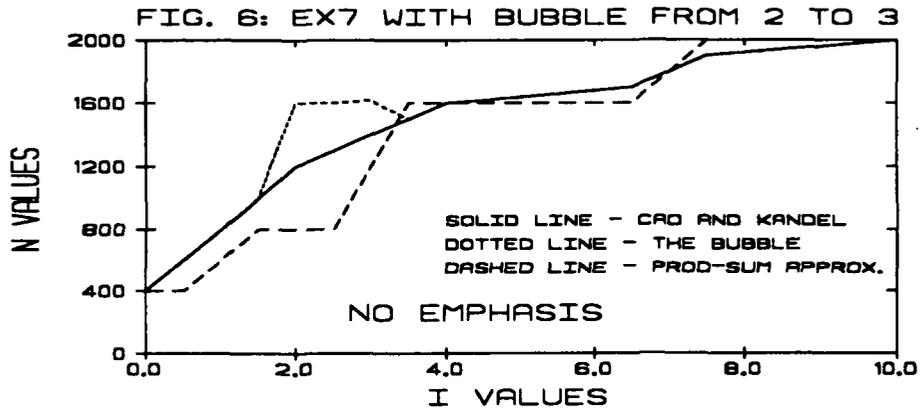
NULL -> VERY LARGE	NULL -> VERY LARGE	NULL -> VERY LARGE
ZERO -> MEDIUM	ZERO -> MEDIUM	ZERO -> MEDIUM
SMALL -> ZERO	ZERO-SMALL -> ZERO	SMALL -> ZERO
MEDIUM -> MEDIUM	SMALL -> ZERO	(repeat above for emphasis)
LARGE -> VERY LARGE	SMALL-MEDIUM -> MEDIUM	(repeat above for emphasis)
VERY LARGE -> MEDIUM	MEDIUM-LARGE -> VERY LARGE	MEDIUM -> MEDIUM
	LARGE -> LARGE	LARGE -> VERY LARGE
	VERY-LARGE -> MEDIUM	VERY-LARGE -> MEDIUM

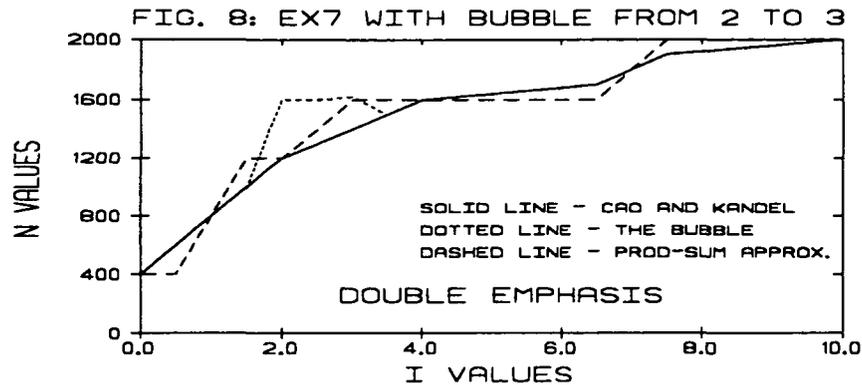
TABLE 2: RESULTS FOR EX. 3 OF CAO AND KANDEL WITH BOTTOM FLATTENED AND ONE PEAK SHIFTED			
	6 ANT. SETS	8 ANT. SETS	6 ANT. SETS
INFERENCE TYPE	MAX-MIN	MAX-MIN	PROD-SUM
MCC	0.896	0.972	0.969
REL. ENTROPY	7.68	4.97	5.76
FOR 6 ANTECEDENT SETS MM $Y = 1.06X - (126.10, 16.29, 16.29)$			
FOR 8 ANTECEDENT SETS MM $Y = (55.78, 7.85, 7.85) + 0.97X$			
FOR 6 ANTECEDENT SETS PS $Y = (79.82, 14.16, 14.16) + 0.95X$			



3. In this example we return to the data of Example 7 in [15], but we add a bubble to the line at input values 2 to 3. As we emphasize the rule which raises the output values in that area (SMALL -> LARGE), first once and then twice, we observe corresponding improvement in the results. This improvement is obvious in the figures below, and is also tracked nicely once again by the statistics. Note that only the "double emphasis" inference creates a transition point in WSSD values in the center of the bubble. Since the effect of emphasis is essentially to shift a transition point toward the emphasized region, this is a sign that the input and output data sets are a good match. As an illustration of the value of the WSSD, we modified the single emphasis inference results so that just the spreads matched better in the bubble. Note that, as one might expect, this improves the overall least squares solution, but note also that this creates a WSSD transition point in the proper place. Since this would not be apparent from an inspection of the modes alone, the value of the WSSD to a detailed evaluation of the inference results is clear.

TABLE 3: RESULTS FOR EXAMPLE 7 OF CAO AND KANDEL WITH BUBBLE ADDED				
STATISTIC	DOUBLE EM	SINGLE EM	NO EMPHAS	SINGL EM +
MCC	0.967	0.9565	0.896	0.9566
ENT	3.90	4.45	6.48	4.27
WSSD 3.5	0.022243	0.026888	0.071660	0.027036
WSSD 3.0	0.022243	0.026888	0.133615	0.027036
WSSD 2.5	0.089423*	0.398910	1.005964	0.378739*
WSSD 2.0	0.438807	0.398910	1.005964	0.400629
WSSD 1.5	0.438807	0.398910	1.005964	0.400629
FOR DOUBLE EMPHASIS $Y = (163.27, 24.75, 24.75) + 0.91X$				
FOR SINGLE EMPHASIS $Y = (199.36, 30.17, 30.17) + 0.89X$				
FOR NO EMPHASIS $Y = (503.50, 77.23, 77.23) + 0.73X$				
FOR SINGLE EMPHAS. + $Y = (198.51, 28.33, 28.33) + 0.89X$				
+ DIFFERS FROM SINGLE EMPHASIS ONLY IN FUZZINESS OF VALUES IN BUBBLE (BETTER MATCH)				





References:

- [1] S. Smith and D. Comer, "Automated Calibration of a Fuzzy Logic Controller Using a Cell State Space Algorithm," *IEEE Control Systems* 11:5 (August, 1991), 18-28.
- [2] K. Ostaszewski and W. Karwowski, "Fuzzy Clustering Methods for Assessment of Population Risks in the Insurance Industry," in P. Ralston, K. Stoll, and T. Ward (eds.), *Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference*, Louisville, Ky., 156-160.
- [3] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Philadelphia: SIAM, 1991.
- [4] C. S. Hsu, *Cell-to-Cell Mapping*, New York: Springer-Verlag, 1987.
- [5] Y. Chen and T. Tsao, "A Description of the Dynamical Behavior of Fuzzy Systems," *IEEE Transactions on Systems, Man, and Cybernetics* 19:4 (1989), 745-755.
- [6] P. Diamond, "Fuzzy Least Squares," *Information Science* 46 (1988), 141-157.
- [7] N. Kiryati and A. Bruckstein, "What's in a Set of Points," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14:4 (1992), 496-500.
- [8] M. Sugeno and T. Yasukawa, "Linguistic Modeling Based on Numerical Data," *IFSA '91 Brussels (Computer, Management, and Systems Science Section)*, 264-267.
- [9] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [10] J. Barone, "Fuzzy Least Squares and Fuzzy Entropy," in P. Ralston, K. Stoll, and T. Ward (eds.), *Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference*, Louisville, Ky., 170-181.
- [11] C. Daniel and F. Woods, *Fitting Equations to Data: Computer Analysis of Multifactor Data*, New York: John Wiley & Sons, 1971.
- [12] S. Chatterjee and A. Hadi, *Sensitivity Analysis in Linear Regression*, New York: John Wiley & Sons, 1988.
- [13] O. Kaleva, "Interpolation of Fuzzy Data," *IFSA '91 Brussels (Mathematics Section)*, 99-101.
- [14] D. F. Mayers, "Convergence of Polynomial Interpolation," in D. C. Handscomb (ed.), *Methods of Numerical Approximation*, Oxford: Pergamon Press, 1966, 15-26.
- [15] Z. Cao and A. Kandel, "Applicability of Some Fuzzy Implication Operators," *Fuzzy Sets and Systems* 31 (1989), 151-186.
- [16] M. Mizumoto, "Fuzzy Controls with Fuzzy Rules of Emphatic and Suppressive Types," in P. Ralston, K. Stoll, and T. Ward (eds.), *Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference*, Louisville, Ky., 134-140.

## A model for amalgamation in group decision making

Vincenzo CUTELLO  
*International Computer Science Institute  
Berkeley, California, USA and  
Fuzzy Logic R & D Project  
CoRiMMe\*, Catania, Italy*

Javier MONTERO†  
*Faculty of Mathematics  
Complutense University  
Madrid 28040, Spain*

### Abstract

In this paper we present a generalization of the model proposed by Montero in [Mon87a, Mon87b, Mon92], by allowing non complete fuzzy binary relations for individuals. A degree of unsatisfaction can be defined in this case, suggesting that any democratic aggregation rule should take into account not only ethical conditions or some degree of rationality in the amalgamating procedure, but also a minimum support for the set of alternatives subject to the group analysis.

**Key words:** Aggregation rules, fuzzy preferences, group decision making.

## 1 Introduction

When dealing with the problem of amalgamating individual (or group) opinions, it is usually stated that the set of alternatives is fixed and has been previously (well) defined. Moreover, individuals are assumed to be not only able to judge which alternative is the best between any pair of alternatives, but also in favor of at least one of them. However, we know that these assumptions are not true in practice. Indeed, in any democratic voting process there always is some level of *abstention*. A portion of this abstention can be analyzed through statistical techniques since it can be associated with sampling difficulties. Another portion of this abstention gives instead important information and may become a decisive factor since a too high level of abstention can even make null the whole democratic process. Many can be the causes of abstention, among them:

- low interest: people think that the issues subject to vote are not relevant, so perhaps more information was needed;
- dislike of alternatives: people do not like any of the alternatives presented to them, therefore different alternatives should be proposed.

\*Consorzio per la Ricerca sulla Microelettronica nel Mezzogiorno, Università di Catania ed SGS-Thomson

†Present address: School of Business Administration, 350 Barrows Hall, University of California at Berkeley, Berkeley, CA 94720.

No interest (or low information) is usually avoided by means of hard and expensive campaigning. The second situation represents however a key issue in democracy. In fact, abstention is sometimes used by political parties (that are mixing in this way an important democratic opinion with no interest or even non democratic attitudes). This is the case for referendums where a particular law  $\mathcal{L}$  already existing is subject to vote with three basic ballots: yes (*I want the law to be abolished*), no (*I do not want the law to be abolished*), blank (*I do not care*). Blank votes allow to reach a fixed level of participation (usually 50 %) which is requested in order for the referendum to be legally valid. Therefore, even though blank votes are intended to be indifferent to both alternatives, in reality they are helping both of them (and in particular the winner) and justifying the process itself. For instance, 20 % yes, 15 % no, 20 % blank and 45 % of abstention will cause the law  $\mathcal{L}$  to be abolished. Therefore, blank votes must be understood as representing a *positive* indifference to the outcome of the voting process. Thus, this kind of indifference must be distinguished from the *negative* indifference, which represents the fact that both alternatives are rejected. A *red* (rejection) vote can then be included in some democratic voting procedures in order to estimate the real support given by the people to the set of alternatives (the technical vote *null* cannot be understood as a *red* vote in any way). Total participation, *yes*, *no*, *blank* and *red* votes, gives information about interest or information level, and no democratic meaning exists if a minimum of votes is not reached. The proportion of red votes over the total number of votes estimates the degree of dissatisfaction with the set of alternatives under analysis and if such a degree is too high the whole set of alternatives is rejected.

Our objective here is to show how fuzzy preferences over the set of alternatives provide us with an easy way to model such a negative indifference. Fuzzy preferences are modeled naturally by means of fuzzy binary relations. The theory of fuzzy relations was originally introduced by Zadeh in his seminal paper [Zad65] and subsequently developed in [Zad71].

In this paper, we generalize the model initially introduced in [Mon87a, Mon87b] where the set  $\mathcal{P}(X)$  of all fuzzy preference relations on  $X$ , i.e.

$$\mu : X \times X \rightarrow [0, 1]$$

verifying

$$\mu(x, y) + \mu(y, x) \geq 1, \quad \forall x, y \in X$$

were considered in order to model individual and social opinions. Adopting Arrow's crisp model [Arr64], completeness assumption of fuzzy preferences was introduced and postulated to model comparability between alternatives. The following values were introduced

$$(I) \quad \mu_I(x, y) = \mu(x, y) + \mu(y, x) - 1$$

$$(B) \quad \mu_B(x, y) = \mu(x, y) - \mu_I(x, y)$$

$$(W) \quad \mu_W(x, y) = \mu(y, x) - \mu_I(y, x)$$

and were understood as the degree of indifference of the alternatives  $x, y$ , the degree of strict preference of alternative  $x$  over alternative  $y$  and the degree of strict preference of alternative  $y$  over alternative  $x$ , respectively. In this paper, we drop the completeness hypothesis and therefore, assuming a meaningful level of comparability between alternatives, we also drop the hypothesis that comparability is modeled by completeness. We will also show how intensities of negative indifference can be associated with non complete fuzzy preference relations.

As usual, we will also assume that the set of individuals and the set of alternatives are both finite, with at least two individuals and three alternatives.

## 2 Incomplete fuzzy preferences

Let us assume from now on that each value  $\mu(x, y)$  represents the degree to which comparison between alternatives  $x$  and  $y$  is in favor of  $x$ , i.e. the degree of support of alternative  $x$  over alternative  $y$ . In case  $\mu(x, y) + \mu(y, x) \geq 1$  it can be understood that the comparison between both alternatives generates no problems. We then assign intensities of preference according to the above formulae, associating the exceeding value  $\mu(x, y) + \mu(y, x) - 1$  with the degree of positive indifference. On the other hand, if  $\mu(x, y) + \mu(y, x) < 1$ , the value  $1 - \mu(x, y) - \mu(y, x)$  can be understood as the degree to which the comparison has not been accepted. Notice that this remaining intensity value cannot be associated to any distinction between both alternatives, so it is some kind of indifference mainly due to a negative opinion of the comparison itself. Therefore, given an arbitrary fuzzy preference relation  $\mu : X \times X \rightarrow [0, 1]$ , for any fixed pair of alternatives we define

$$\begin{aligned}\mu_{PI}(x, y) &= \max(\mu(x, y) + \mu(y, x) - 1, 0) \\ \mu_{NI}(x, y) &= \max(1 - \mu(x, y) - \mu(y, x), 0)\end{aligned}$$

in order to capture the degrees of negative and positive indifference. Notice, in particular that  $\mu_{PI}(x, y)$  is the *Lukasiewicz* T-norm representing in this case  $x \geq y$  and  $y \geq x$ . It is, then, obvious that the meaning of the expressions

$$\begin{aligned}\mu_B(x, y) &= \mu(x, y) - \mu_{PI}(x, y) \\ \mu_W(x, y) &= \mu(y, x) - \mu_{PI}(x, y)\end{aligned}$$

is kept. Obviously, this model is based on the assumption that both positive and negative indifference are basically indifferences, so that standardized (complete) preferences can be defined

$$\mu^*(x, y) = \mu(x, y) + \mu_{NI}(x, y) = 1 - \mu_W(y, x)$$

The  $\mu^*$  will be called the *completion* of  $\mu$ .

Moreover, each value

$$\sigma(x, y) = 1 - \mu_{NI} = \min\{\mu(x, y) + \mu(y, x), 1\} \quad (1)$$

can be associated to the degree to which the comparison between the pair of alternatives  $x, y$  is being supported.

According to the above comments, we should be able to evaluate in some way the degree of support for the process itself, and afterwards (assuming a minimum support) obtain the aggregated fuzzy preference relation. Though we will not comment here on how a final decision can be made from such information, we will analyze how to aggregate support and preference intensities.

## 3 Ethical conditions and rationality

The definition given in [Mon87a, Mon87b] for the measure of acyclicity of a fixed chain

$$G = (x_1, x_2, \dots, x_k, x_{k+1})$$

with  $x_{k+1} = x_1$ , of different alternatives is obviously still valid for standardized preferences. Indeed, we can define  $A^*(\mu) = A(\mu^*)$ , where  $A$  is defined as  $A(\mu^*) = \min_G A_{\mu^*}(G)$ , the minimum is evaluated along all chains in  $X$  and

$$A_{\mu^*}(G) = 1 - (\prod_{i=1}^k \mu^*(x_i, x_{i+1}) + \prod_{i=1}^k \mu^*(x_{i+1}, x_i) - 2 \prod_{i=1}^k \mu_j^*(x_i, x_{i+1})).$$

Notice that the value

$$\mu_I^*(x, y) = |\mu(x, y) + \mu(y, x) - 1| = \max\{\mu_{PI}(x, y), \mu_{NI}(x, y)\}$$

is understood as a degree of *technical* indifference. However, going back to the referendum example, the opinion of those people against the referendum cannot be used to discriminate between *yes* and *no*. In the following, our final aggregation model will make use of those non complete preferences but as pointed out such aggregated values cannot be properly considered without estimating the support of the global decision problem.

The problem can then be stated as follows: is it possible to find fuzzy aggregation rules that are non (absolutely) irrational? Therefore, we shall assume that all individual opinions are non irrational in the above sense (i.e.  $A^*(\mu) = A(\mu^*) > 0$ ) and for simplicity we shall also assume that all fuzzy preferences are *reflexive* meaning  $\mu(x, x) = 1$  for all  $x \in X$ . Hence, a non absolutely irrational (NAI) aggregation rule in this general context will be defined as a mapping  $S : \mathcal{F}^n(X) \rightarrow \mathcal{P}(X)$ , where

$$\begin{aligned}\mathcal{F}(X) &= \{\mu \mid \mu(x, x) = 1 \ \forall x \in X, A(\mu^*) > 0\} \\ \mathcal{P}(X) &= \{\mu \mid \mu \in \mathcal{F}(X) \wedge \mu(x, y) + \mu(y, x) \geq 1 \ \forall x, y \in X\}\end{aligned}$$

i.e.  $\mathcal{F}(X)$  is the collection of all reflexive, fuzzy binary relations over  $X$  and  $\mathcal{P}(X)$  is the collection of all complete fuzzy binary relations over  $X$ . Notice that social aggregated opinion is assumed to be complete, according to the above comments about usual practice in democracy. The information about people supporting aggregation is a question that we will try to answer later on.

Ethical conditions analogous to those given in [Mon87b, Ovc90, Mon92] or deriving from them can be imposed

(UD) *Unrestricted Domain*: the mapping  $S$  is defined over all possible profiles of reflexive fuzzy preferences provided that the support of the set of alternatives is not absolutely null. According to the definition given in (1) this means that for all  $x, y \in X$  there exists  $i$  such that  $p^i(x, y) > 0$  or  $p^i(y, x) > 0$ . This will be subsequently clarified in section 5.

(NNR) *Non Negative Responsiveness*: for any  $(x, y) \in X \times X$  if  $p^i(x, y) \geq q^i(x, y)$  and  $p_w^i(x, y) \leq q_w^i(x, y)$  then

$$S(p^1, \dots, p^n)(x, y) \geq S(q^1, \dots, q^n)(x, y).$$

(IIA) *Independence of Irrelevant Alternatives*:  $p^i(x, y) = q^i(x, y), \forall i$  and  $\forall x, y \in Y \subseteq X$  implies that

$$S(p^1, \dots, p^n)(x, y) = S(q^1, \dots, q^n)(x, y)$$

for any  $Y$  nonempty subset of  $X$ .

(A) *Anonymity*: given any permutation of the set of individuals  $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$  we have

$$S(p^1, \dots, p^n) = S(p^{\pi(1)}, \dots, p^{\pi(n)}).$$

(N) *Neutrality*: given any permutation of the set of alternatives  $\pi : X \rightarrow X$ , if  $p^i(x, y) = q^i(\pi(x), \pi(y))$  for all  $i$  and  $x, y \in X$ , then

$$S(p^1, \dots, p^n)(x, y) = S(q^1, \dots, q^n)(\pi(x), \pi(y))$$

for all  $x, y \in X$ .

(CS) *Citizen Sovereign*: for any given  $p \in \mathcal{P}(X)$  there exists a profile  $(p^1, \dots, p^n)$  such that

$$S(p^1, \dots, p^n) = p.$$

Notice that A implies the following condition

(ND) *Non Dictatorship*: there is no individual  $i$  such that

$$S(p^1, \dots, p^n) = p^i$$

for any  $(p^1, \dots, p^{i-1}, p^{i+1}, \dots, p^n)$ .

Moreover, notice that condition IIA does not imply that  $S(p^1, \dots, p^n)(x, y)$  depends solely on the values  $p^i(x, y)$ . In fact, such a conditions implies that in order to get the value  $S(p^1, \dots, p^n)(x, y)$  we need the values  $p^1(x, y), \dots, p^n(x, y)$  along with the values  $p^1(y, x), \dots, p^n(y, x)$ . For instance,  $p^i(x, y) = 0$  for all  $i$  does not necessarily imply that  $S(p^1, \dots, p^n)(x, y) = 0$ , even if we assume NNR and CS simultaneously. The condition  $p^i(y, x) = 1$  for all  $i$ , needs also to be imposed to reach such a conclusion. Condition NNR has also been modified coherently with IIA. Finally, since we want the social aggregation to be complete the Unanimity condition

(U) *Unanimity*: if  $p^i = p$  for all  $i$  then

$$S(p^1, \dots, p^n) = p$$

cannot be imposed.

In the next section some particular aggregation rules are proposed in order to show that no Impossibility theorem holds in our context.

## 4 Aggregation Rules

First we notice that the mean aggregation rule (analyzed in [Mon88b]) which has been shown to be a NAI aggregation rule in the case that all individual preferences are complete (see [Mon87a, Mon87b])

$$M(p^1, \dots, p^n)(x, y) = \sum_{i=1}^n p^i(x, y)/n$$

does not assure rationality when individual preferences are not required to be complete. Indeed, consider the following example.

**Example 4.1** Let  $p^1$  and  $p^2$  be two individuals expressing their opinions about three different alternatives  $\{x, y, z\}$  in the following way:

$$\begin{aligned} p^1(x, y) = p^1(x, z) = p^1(y, z) = p^1(y, x) = p^1(z, x) = p^1(z, y) = 1 \\ p^2(x, y) = p^2(y, x) = 0, \quad p^2(x, z) = p^2(y, z) = p^2(z, x) = p^2(z, y) = 1 \end{aligned}$$

Intuitively, the individual  $p^1$  is fully satisfied by the set of alternatives. Individual  $p^2$  though not satisfied by alternatives  $x$  and  $y$  is fully content with the final decision as long as alternative  $z$  is taken under consideration.

We then have two NAI individual preferences but the aggregation

$$\begin{aligned} M(p^1, p^2)(x, y) &= M(p^1, p^2)(y, x) = 1/2 \\ M(p^1, p^2)(y, z) &= M(p^1, p^2)(x, z) = M(p^1, p^2)(z, x) = M(p^1, p^2)(z, y) = 1 \end{aligned}$$

is irrational. □

Let us propose now one aggregation rule which is not irrational.

(R1) The rule is based on standardized intensities:

$$T(p^1, \dots, p^n)(x, y) = \sum_{i=1}^n (1 - p_W^i(x, y)) / n \quad \forall x, y \in X.$$

It easy to see that the above rule correspond to the mean rule in the case of standardized preferences, i.e.

$$T(p^1, \dots, p^n)(x, y) = M(p^{*,1}, \dots, p^{*,n})(x, y) = \sum_{i=1}^n p^{i,*}(x, y) / n$$

where  $p^{i,*}$  is the completion of  $p^i$ . Therefore,  $T(p^1, \dots, p^n)$  is not absolutely irrational and verifies all of the ethical conditions.

The following property gives a sufficient condition for NAI aggregation rules that can be easily checked.

**THEOREM 4.1** Let  $S : \mathcal{F}^n(X) \rightarrow \mathcal{P}(X)$  be a mapping verifying condition IIA and such that for any fixed pair of alternatives  $x, y$  the following relations hold

$$(C1) \quad S(p^1, \dots, p^n)(x, y) = 1 \quad \rightarrow \quad \forall i \quad p_W^i(x, y) = 0$$

$$(C2) \quad S(p^1, \dots, p^n)_I(x, y) = 0 \quad \rightarrow \quad \exists i | p^i(x, y) + p^i(y, x) = 1$$

then  $S$  is a NAI aggregation rule.

**Proof.** Let  $G$  be a fixed chain. If, on one hand, there is some individual acyclic path with some strict preference, that is  $p_B^i(x, y) > 0$  for some  $i$  and some  $(x, y)$  in  $G$ , since  $p_W^i(y, x) = p_B^i(x, y)$  we then have  $S(p^1, \dots, p^n)(y, x) < 1$ . Hence, in view of the fact that  $S(p^1, \dots, p^n)$  is complete, it must be the case that  $S(p^1, \dots, p^n)_B(x, y) > 0$ . Therefore, such acyclic path will have positive weight in the aggregated fuzzy preference and the aggregation will not be irrational. On the other hand, if  $p^i(x, y) + p^i(y, x) \neq 1$  for all  $i$  and all pairs  $(x, y)$  in  $G$  then it must be  $S(p^1, \dots, p^n)_I(x, y) > 0$  for all  $(x, y)$  in  $G$ . Therefore the indifferences acyclic path has a positive weight and in this case we also obtain a rational aggregation. ■

The converse of the above theorem does not hold, as can be easily seen by considering the following rule

$$I(p^1, \dots, p^n)(x, y) = 1$$

for all  $x, y \in X$ .

Moreover, consider the following aggregation rule.

*Amortized intensities rule :*

$$T_0(p^1, \dots, p^n)(x, y) = \sum_{i=1}^n p^i(x, y) / C.$$

$$\text{where } C = \sum_{i=1}^n \min(p^i(x, y) + p^i(y, x), 1) = \sum_{i=1}^n p^i(x, y) + p_W^i(x, y).$$

The above rule verifies all ethical conditions and it is obviously complete but, as we will prove below, verifies only condition (C1) of the theorem and in fact it is not rational.

Let us then prove that  $T_0(p^1, \dots, p^n)(x, y) = 1$  implies that  $p_W^i(x, y) = 0 \quad \forall i$ . Let us first define  $H = \{i : p^i(x, y) + p^i(y, x) < 1\}$ .

Suppose on one hand that  $T_0(p^1, \dots, p^n)(x, y) = 1$ . Then

$$\sum_{i=1}^n p^i(x, y) = \sum_{i=1}^n \min(p^i(x, y) + p^i(y, x), 1).$$

Since  $p^i(x, y) \leq \min(p^i(x, y) + p^i(y, x), 1)$  then it must be the case that  $p^i(x, y) = \min(p^i(x, y) + p^i(y, x), 1)$  for all  $i$ . Two cases are possible:

(1) if  $i \in H$  then  $p^i(y, x) = 0$  which implies that  $p_W^i(x, y) = 0$ .

(2)  $i \notin H$  then  $p^i(x, y) = 1$  which implies that  $p_W^i(x, y) = 0$ .

In both cases then  $p_W^i(x, y) = 0$  for all  $i$ .

To prove that  $T_0(p^1, \dots, p^n)$  is not rational consider the following example. We have two individuals  $p^1$  and  $p^2$  and three alternatives  $x, y$  and  $z$ . The two individuals have the same opinion  $p$ :

$$\begin{aligned} p(x, y) &= p(y, x) = 1 \\ p(y, z) &= p(z, y) = 1 \\ p(z, x) &= p(x, z) = \frac{1}{3} \end{aligned}$$

Then, we have

$$\begin{aligned} T_0(p^1, p^2)(x, y) &= T_0(p^1, p^2)(y, x) = 1 \\ T_0(p^1, p^2)(y, z) &= T_0(p^1, p^2)(z, y) = 1 \\ T_0(p^1, p^2)(z, x) &= T_0(p^1, p^2)(x, z) = \frac{1}{2} \end{aligned}$$

and it can be seen that  $A(T_0(p^1, p^2)) = 0$  (cfr. section 3).

We can modify  $T_0$  in the following way

(R2)  $\epsilon$ -amortized intensities:

$$T_\epsilon(p^1, \dots, p^n)(x, y) = \frac{\sum_{i=1}^n p^i(x, y) + \epsilon}{C + \epsilon}.$$

It is easy to see that the above rule (R2) gives a NAI aggregation rule for every  $\epsilon > 0$ : it is complete and verifies conditions (C1) – (C2) of theorem 4.1. About condition (C2) of theorem 4.1 notice that the  $\epsilon$ -amortized aggregated opinion will never verify  $T_\epsilon(p^1, \dots, p^n)(x, y) + T_\epsilon(p^1, \dots, p^n)(y, x) = 1$ .

## 5 Support Analysis

Given a fixed pair of alternatives  $(x, y)$  and the individual preference values  $p^i(x, y)$  and  $p^i(y, x)$ , the support of such a comparison relative to the individual  $i$ , according to (1) will be the value

$$\sigma^i(x, y) = \min(p^i(x, y) + p^i(y, x), 1) = 1 - p_{NI}^i(x, y)$$

i.e. the Lukasiewicz co-T-norm. Our problem is to obtain for each pair of alternatives a social support value to be evaluated from individual preferences. This problem is therefore analogous to the previous aggregation problem, and since both problems are dealing with intensity values, they should be solved in a coherent way.

Let us first describe a representation result related with aggregation rules.

**THEOREM 5.1** Let  $S : \mathcal{F}^n(X) \rightarrow \mathcal{P}(X)$ . Then  $S$  verifies IIA and N conditions if and only if there exists a function  $\phi_S : [0, 1]^{2n} \rightarrow [0, 1]$  such that

$$S(p^1, \dots, p^n)(x, y) = \phi_S(p^1(x, y), p^1(y, x); \dots; p^n(x, y), p^n(y, x))$$

for all  $x, y \in X$ .

**Proof.** In view of the condition IIA, it is clear that for each pair  $(x, y)$  the value  $S(p^1, \dots, p^n)(x, y)$  is perfectly determined once the values  $p^1(x, y), p^1(y, x), \dots, p^n(x, y), p^n(y, x)$  are given. Therefore  $S$  can be defined by a set of mappings  $\phi_S^{(x, y)} : [0, 1]^{2n} \rightarrow [0, 1]$  such that

$$S(p^1, \dots, p^n)(x, y) = \phi_S^{(x, y)}(p^1(x, y), p^1(y, x); \dots; p^n(x, y), p^n(y, x))$$

for such a fixed pair of alternatives. However, due to the N condition, these mappings  $\phi_S^{(x, y)}$  do not in fact depend on the particular pair of alternatives.

The converse is trivial. ■

Notice that since  $p_W^i(x, y) = p^i(x, y) - \max\{p^i(x, y) + p^i(y, x) - 1, 0\}$  social aggregation values will be also determined if the values  $p_W^i(x, y)$  are given instead of the values  $p^i(y, x)$ .

The characterization given by theorem 5.1 allows us to define the social support  $\sigma$  in a coherent way with respect to the social preference:

$$\sigma(x, y) = \phi_S(\sigma^1(x, y), 1 - \sigma^1(y, x); \dots; \sigma^n(x, y), 1 - \sigma^n(y, x))$$

for all  $x, y \in X$ .

With this definition all ethical conditions imposed on the social preferences aggregation rule are automatically imposed on the social support aggregation rule. The final social opinion will contain an ethical and non irrational fuzzy preference relation (complete in order to be useful in the subsequent decision making process) but also the aggregated support function.

## 6 Final Remarks

In this paper a welfare oriented approach (similar to Arrow's model) has been developed, but not a decision oriented one. Real democratic problems are more related to decision making problems, and in this case an analysis of the stability of the final decision should also be included (see for example [Mon90]). In any case, by using fuzzy preference relations, we have been able not only to avoid Arrow's paradox but also other similar restrictive results in the fuzzy context (see [FF75] and also [Mon85, Mon88a]).

Moreover, it has been shown how the problem of negative indifference can be modeled within the fuzzy preferences framework, just dropping out the assumption of completeness. In fact, it is suggested a natural way of dealing with dislike of proposed alternatives and therefore a measure of their support. Critical levels of such a support should be previously defined depending on the characteristic of the alternatives and their social significance.

**Acknowledgements:** The authors wish to thank the referee for her/his helpful suggestions.

This research has been partially supported by Dirección General de Investigación Científica y Técnica (Spanish national grants PB88-0137 and BE91-225).

## References

- [Arr64] K.J. Arrow. *Social Choice and Individual Values*. John Wiley & Sons, Inc., New York, 1964.
- [FF75] L.W. Fung and K.S. Fu. An axiomatic approach to rational decision making in a fuzzy environment. In L.A. Zadeh, K.S. Fu, K. Tanaka, and M. Shimura, editors, *Fuzzy sets and their applications to Cognitive and decision processes*, pages 227-256. Academic Press, 1975.
- [Mon85] J. Montero. A note on fung-fu's theorem. *Fuzzy sets and systems*, 17:259-269, 1985.
- [Mon87a] J. Montero. Arrow's theorem under fuzzy rationality. *Behavioral Science*, pages 267-273, 1987.
- [Mon87b] J. Montero. Social welfare functions in a fuzzy environment. *Kybernetes*, 16:241-245, 1987.
- [Mon88a] J. Montero. Aggregation of fuzzy opinions in a non-homogeneous group. *Fuzzy sets and systems*, 25:15-20, 1988.
- [Mon88b] J. Montero. An axiomatic approach to fuzzy multicriteria analysis. In M.M. Gupta and T. Yamakawa, editors, *Fuzzy logic in Knowledge-Based Systems, Decision and Control*, pages 259-269. North-Holland, 1988.
- [Mon90] J. Montero. Single-peakedness in weighted aggregation of fuzzy opinions in a fuzzy group. In J. Kacprzyk and M. Fedrizzi, editors, *Multiperson decision making using fuzzy sets and possibility theory*, pages 163-171. Kluwer, 1990.
- [Mon92] J. Montero. Fuzzy rationality as a basis for group decision making. In *Fuzz-IEEE 1992*, pages 1237-1243, San Diego, CA, 1992. Proceedings of the IEEE International Conference on Fuzzy Systems.
- [Ovc90] S. Ovchinnikov. Means and social welfare function in fuzzy binary relation spaces. In J. Kacprzyk and M. Fedrizzi, editors, *Multiperson decision making using fuzzy sets and possibility theory*, pages 143-154. Kluwer, 1990.
- [Zad65] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338-353, 1965.
- [Zad71] L.A. Zadeh. Similarity relations and fuzzy orderings. *Information Science*, 3:177-200, 1971.

525-81

161825

l- 5

N93-29541

FUZZY FORECASTING AND DECISION MAKING  
IN SHORT DYNAMIC TIME SERIES

Professor KARPOVSKY Efim Ja., D. Sc. (comp.),  
Head of Computer Science Department,  
Odessa Institute of National Economy.  
OZOYA Ibrahim Ohiokhoya (M. Sc.),  
post-graduate student, Department of Computer Science  
Odessa Institute of National Economy.  
Soviet Armija 8, Odessa 270100, Ukraine

ABSTRACT

This paper focuses on the usage of the fuzzy set theory in forecasting and decision-making in areas with short dynamic time series.

INTRODUCTION

In different areas of human activities, there exists the objective necessity of decision-making and appraising future tendencies of processes under research. When there exists complete and adequate statistical figures or materials on the behavior of the object under research, it is adequate to use well-known orthodox methods or a combination of them to achieve the set objective.

Unfortunately today, there are very many areas of knowledge where, due to many objective reasons, there is the lack of adequate and complete statistical data, i.e. exhaustiveness of basic information.

However, even in cases like this, there is the need for a glance at the future (extrapolation) and a decision made based on that, using the available data. For example: We have a set of factors described by  $Y = (y_1, y_2, \dots, y_k)$ , the activities of which affects a set of other factors described by  $F = (f_1, f_2, \dots, f_r)$ . It is necessary to access the future trend of  $F$  and how it is affected by the factors  $Y$ , and arrive at a decision.

There exists about three possible directions of solving this type of problems:

- i. Classical and traditional (orthodox) methods,
- ii. Expert evaluation methods,
- iii. Fuzzy mathematics.

Classical and well-known traditional methods (like correlation-regression analysis, linear programming, etc.) require the length of the time-series to be about 4-6 times longer than the range of forecast, i.e.  $4n > m$ , where  $n$  = length of the time-series, and  $m$  = range of forecast.

Forecasting methods based on expert evaluation allows for the "informal" usage or part-usage of the statistical information at hand and the subjective evaluation of the experts involved.

In other words, classical methods of forecasting in cases of short time-series are usually not applicable, since they do not satisfy the methodical assumptions and propositions of mathematical statistics. Other methods based on expert evaluation also cannot be used because of the possibility of giving subjective (un-objective) estimates and the "in-complete" usage of the available statistical data.

In view of these problems, it is necessary to advance that type of technique based on a complete statistical assessment of data and excluding all sorts of subjective estimates.

#### CALCULATION OF COMPOSED MEMBERSHIP FUNCTION

Let the linguistic meanings of forecasting the economic indices of the economic problem be expressed with the help of the membership function  $m_k(u) \in [0,1]$ , where  $u \in [0,1]$ ,  $k = \overline{1, N}$

The matrix  $M = \|m_k(u)\|$  consists of a set of the membership function (MF) that in general characterizes the fuzzy model of the economic indices under review.

The composed membership function of the model we suggest should be calculated as the linear functions as below:

$$F = A^{-1} M,$$

where  $A = HG^{1/2}$  - matrix of the weighting coefficients of the principal components;

$G$  - eigenvalue vector of the matrix  $R$ ;

$H$  - matrix of eigenvalue vectors of matrix  $R$

$R = 1/N MM^T$  - correlation matrix of the MF for an index.

To calculate the elements of the vector  $G$  and matrix  $H$ , the following matrix equations are solved:

$$\|R - gE\| = 0,$$

$$g \in G$$

$$(R - gE) H = 0,$$

where  $E = \|e_{ij}\|$  is the matrix, in which  $e_{ii} = 1$ ;  $e_{ij} = e_{ji} = 0$ ,  $i \neq j$ .

More precise description of this method calculating the composed MF can be found in [1].

Let's examine closely an example of decision-making on an economic problem concerning, for instance, the production (or estimated level of output) of a new commodity. At our disposal are three known factors like

- (i) cost of production (cost price),
- (ii) per capital output (capital intensity),
- (iii) taxation policy,

affecting production level.

On these three factors we have only a limited dynamic time-series of 3-4 years. However, it is necessary to make a decision on the production of the new commodity.

Let the following fuzzy linguistic variables express the membership function of the above factors under review:

< cost of production	> _____ >	< satisfactory good	>
< per capital output	> _____ >	< average	>
< taxation policy	> _____ >	< good	>

Employing known formulae of the membership functions' setting we can determine the meanings:

	0	.2	.4	.6	.8	1.0
< cost price >	0	.0	.6	.8	.9	1.0
< capital intensity >	0	.0	.6	1.0	.6	.0
< taxation policy >	0	.0	.4	.6	.8	1.0

Using the method of principal components, the values of the first principal component weighting coefficient for the three factors are as follows:

$$a_1 = 0.3894; a_2 = 0.253; a_3 = 0.363.$$

Then the generalized estimation of our decision-making process has the following composed membership function:

$$MF = 0 | 0 | 0 | .2 + .55 | .4 + .78 | .6 + .75 | 1.0$$

with the average linguistic meaning  $P = 0.72$ , which characterizes the generalized estimation of the decision-making process of the production of the new commodity.

#### FUZZY ANALOGY OF BROWN SMOOTHING METHOD

Calculation of the prognostic models of the estimated level of output of a new commodity put forward in the previous paragraph based on the 3-4 years time-series cannot be done with the help of know formulae. To this end, we advance a technique, the basic conceptions of which are stated below:

I. The smoothing parameter ( @ ) can be calculated as in [2].

$$@ = 2/n + 1, \text{ where } n = \text{length of time-series.}$$

II. Generalized fuzzy model for calculating the fuzzy numbers of a dynamic series ( $i = 1, 2, 3, \dots, n$ ) >

$i = 1:$

$$Y_1 = Y_1(f) \pm @ Y_0$$

$$Y_1(\text{min}) = Y_1 - @ Y_1(f)$$

$$Y_1(\text{MAX}) = Y_1 + @ Y_1(f)$$

| where,

|  $Y_1$  - mean (average) value,

|  $Y_1(f)$  - actual value of factor for first

| member of time-series,

|  $Y_1(\text{min})$  - minimal value,

|  $Y_1(\text{max})$  - maximum value.

Hence,

$$\tilde{Y}_1 = Y_1(f) \pm @ Y_1(f)$$

$i = 2:$

$$Y_2 = @ Y_1 + (1 - @) Y_2(f)$$

$$Y_2(p) = -@ Y_2(1 - @) - (1 - @) Y_1(\text{min}) @$$

$$Y_2(pp) = @ Y_2(1 - @) + (1 - @) Y_1(\text{max}) @$$

| where,

|  $Y_2(\text{min})$  - minimal value

| of second member

| of time-series.

$$Y_2 (\text{min}) = Y_2 - Y_2 (\rho)$$

|  $Y_2$  - mean (average) value

$$Y_2 (\text{max}) = Y_2 + Y_2 (\rho\rho)$$

| of second member

Hence,

| time-series

$$\widetilde{Y}_2 = [Y_2 (\text{min}) ; Y_2 ; Y_2 (\text{max}) ]$$

| etc.

. . . etc

$i = n$ :

| where,

$$Y_n = @ Y_{n-1} + (1-@) Y_n(f)$$

|  $Y_{n-1}$  -fuzzy number of  $n-1$

$$Y_n(\rho) = -@ Y_n (1-@) - (1-@) Y_{n-1} (\text{min}) @$$

| time-series

$$Y_n(\rho\rho) = @ Y_n (1-@) + (1-@) Y_{n-1} (\text{max}) @$$

|

$$Y_n (\text{min}) = Y_n - Y_n (\rho)$$

|

$$Y_n (\text{max}) = Y_n + Y_n (\rho\rho)$$

|

Hence,

$$\widetilde{Y}_n = [Y_n (\text{min}) ; Y_n ; Y_n (\text{max}) ]$$

|

The generalized forecast model of the fuzzy analogy of Brown Smoothing Method can be expressed thus:

$$i = n + 1:$$

$$Y_{n+1} (\text{prog.}) = @ \widetilde{Y}_n + (1-@) Y_n (f)$$

where,

$Y_{n+1} (\text{prog.})$  - forecast level for the year  $n + 1$ ,

$\widetilde{Y}_n$  - fuzzy number of factor for the year  $n$ ,

$Y_n(f)$  - actual value of factor for the year  $n$ .

Based on the calculated fuzzy numbers of the factors for the period (3-4 years) and applying the generalized forecast fuzzy analogy of Brown Smoothing Method, the estimated output level of the new commodity can be calculated.

## FUZZY DECISION MAKING

Based on the basic directive (requirement) measuring the efficiency and profitability of the decision-making process to engage in the production of a new commodity for instance, and the estimates described above, a procedure is developed for decision-making in these conditions.

The composed membership function (MF) of the indices, i.e. the average linguistic meaning  $P$ , serves as a means of making a decision on the production of a new commodity.

The procedure assumes the comparison of two fuzzy numbers based on a set of index values. The result is an interval (span) between the set of fuzzy numbers describing the basic directive of profitability of the economic index on one hand, and the set of fuzzy numbers describing the economic estimates, i.e. the average linguistic meaning  $P$ , on the other hand.

This interval is measured from 0.5 on the relative estimate scale, i.e. decision scale. If any element of the set describing the basic directive of profitability is less (or smaller) than any element of the set of fuzzy numbers describing the economic index estimates, i.e. average linguistic meaning  $P$ , then the interval is positive, if not, then it is negative.

In other words, if the average linguistic meaning  $P$  describing the economic index estimates is to the left of the fuzzy numbers of the basic directive of profitability, then the estimate is negative, i.e. not good enough, and if it is to the right, then it is positive.

The intersection of the basic directive of profitability and the economic index estimates is measured on the decision scale from 0.5 to both sides of the scale. So, if the average linguistic meaning  $P$  falls to the right of this intersection, then the decision-making process based on this is positive. If however, it falls to the left then it is negative.

The composed membership function (MF) on the decision scale describes the fuzzy number corresponding to the profitability of the economic index estimates (estimated level of output of a new product) and the fuzzy decision based on that estimate.

The correctness, i.e. trustworthiness, of the estimate is calculated as a measure of the fuzziness of the resulting fuzzy number.

The validity of the decision is estimated by the square of the fuzzy decision at the interval  $[0.5, 1]$  if the mode of the function is between 0 and 0.5, and in the interval  $[0, 0.5]$  if the mode is between 0.5 and 1.

#### REFERENCES

1. Karpovsky E., Fuzzy Decision Support System Proc. of NAFIPS' 90, Canada, Toronto (1990).
2. Brown R. G., Meyer R. F. The Fundamental Theorem of Exponential Smoothing. Oper. Reser. 1961, Vol. 9, No. 5, pp. 673-687.

# Decision Analysis With Approximate Probabilities

Thomas Whalen  
Professor of Decision Sciences  
Georgia State University  
Atlanta, Georgia 30303-3083

161826

p. 10

## ABSTRACT

This paper concerns decisions under uncertainty in which the probabilities of the states of nature are known only approximately. Decision problems involving three states of nature are studied, since some key issues do not arise in two-state problems, while probability spaces with more than three states of nature are essentially impossible to graph.

The primary focus is on two levels of probabilistic information. In one level, the three probabilities are separately rounded to the nearest tenth. This can lead to sets of rounded probabilities which add up to 0.9, 1.0, or 1.1. In the other level, probabilities are rounded to the nearest tenth in such a way that the rounded probabilities are forced to sum to 1.0. For comparison, six additional levels of probabilistic information, previously analyzed in (Whalen, 1991), were also included in the present analysis.

A simulation experiment compared four criteria for decisionmaking using linearly constrained probabilities (Maximin, Midpoint, Standard Laplace, and Extended Laplace) under the eight different levels of information about probability. The Extended Laplace criterion, which was introduced in [Whalen, 1991] using a second order maximum entropy principle, performed best overall.

## Risk and Uncertainty

The general problem of decision making under uncertainty involves a set of  $n$  states of nature, a set of  $k$  alternative actions, and a utility function that assigns a vector of  $n$  values to each alternative action; each element of this vector specifies the value of the action under the corresponding state of nature. The  $k$  utility vectors typically take the form of row vectors collected into a  $k \times n$  utility matrix associating a specific value to each (state, action) pair.

Standard treatments of decision making under uncertainty fall into two separate branches: decisions under risk and decisions under ignorance [Resnik 1986]. Under risk, the numeric probability of each state of nature is also assumed to be known or estimated. This enables us to reduce the utility vector of each alternative action to a single number, the expected utility found by adding the product of each utility times the probability of the corresponding state of nature. The action whose expected utility is highest is selected.

Under ignorance, there is no knowledge at all about the probabilities of the states of nature. Various criteria exist for making a decision without recourse to probability. Implicitly or explicitly, each of these criteria replaces the weighting role of the missing

probability values with some other weighting scheme to reduce the vector of possible utilities of an action under the various states of nature to a single value to facilitate comparisons between alternative actions. The Laplace criterion emphasizes all states of nature equally. The Hurwicz criterion (of which maximax and maximin are special cases) emphasizes the most favorable and/or the most unfavorable states of nature. The minimax regret criterion emphasizes the states of nature for which the decision makes the most difference.

### Intermediate Cases

In practice, most real decisions use probability information that falls between the well studied extremes of pure risk and pure ignorance. This is especially true in team decision making [Ho & Chu 1972] when one team member assesses a probability distribution but because of time or other constraints can only communicate a standard, concise description of the distribution to the actual decision maker. Each message that can be sent corresponds to a region within a probability space with  $(n-1)$  dimensions, where  $n$  is the number of states of nature. Note that the authors and publishers of handbooks, almanacs, or other sources of potentially useful information can be viewed as generalized "teammates" of everyone who consults their publications.

For example, sometimes we have enough information to arrange the possible states of nature in order from most probable to least probable, or at least identify some as more probable than others, without being able to numerically specify the probabilities of individual states of nature. This ordinal information may come as a summary message from a teammate, or more directly -- e.g. by observing a random walk process after an unknown number of steps. Alternatively, we may have information about which states of nature, if any, have a probability above a specified threshold.

A very important special case of incomplete probability information arises when probabilities are in rounded form; for example, we may be told that  $P(A) = .2$ ,  $P(B) = .3$ , and  $P(C) = .4$  to the nearest tenth. (A, B, and C are a mutually exclusive exhaustive event set whose unrounded probabilities must sum to 1.) When the probabilities are each rounded to the nearest tenth, it is possible that the sum of the rounded probabilities will not equal 1.0. In practice, rounded distributions of this sort are sometimes communicated as-is, but sometimes the probability distribution as a whole is rounded to the nearest set of three probabilities adding to 1.0. Table 1 shows three sets of exact probabilities, which yield different rounded probabilities when rounded separately but all yield the same rounded distribution when forced to sum to 1.0.

Table 1: Two Methods for Rounding Probabilities

Unrounded Probabilities	Rounded Separately	Rounded to add to 1.0
(.333, .336, .331)	(.3, .3, .3)	(.3, .4, .3)
(.310, .360, .330)	(.3, .4, .3)	(.3, .4, .3)
(.366, .367, .266)	(.4, .4, .3)	(.3, .4, .3)

## Linear Probability Constraints & Dempster-Shafer Evidence

The Dempster-Shafer theory of evidence [Shafer, 1976] concerns one particular type of incomplete probability knowledge, represented by basic probability assignments. However, this model does not account for some kinds of probability knowledge that are of great practical importance.

Probability threshold information cannot reliably be expressed by basic probability assignments. For example, with three states of nature we can represent all messages about probability thresholds of  $1/4$  or  $1/3$  by basic probability assignments, but not all messages about a probability threshold of  $1/2$  can be so represented.

When there are only two possible states of nature, the ordinal information that state 1 is more probable than state 2 corresponds to the probability threshold information that  $P(s_1) > .5$ . This can be represented by the basic probability assignment  $m(s_1) = .5$ ,  $m(s_2) = 0$ ,  $m(s_1 \cup s_2) = .5$ . However, when there are more than two possible states of nature, ordinal information about probabilities can never be expressed by basic probability assignments.

Rounded probabilities can sometimes be represented by basic probability assignments, but not when the rounded probabilities add up to less than 1.0. For example, probabilities of .33, .33, and .34 would be rounded to .3, .3, and .3. The knowledge that the true probability distribution is somewhere in the region of probability space that rounds to (.3, .3, .3) would provide a useful approximation to the true probabilities, but it cannot be expressed as a basic probability assignment. When probabilities are forced to sum to 1.0, none of the resulting regions of probability space can be represented by basic probability assignments.

All the above cases, and many others, can be expressed by systems of linear constraints on probabilities. In such a case, the available information restricts the probability to lie within a particular region in probability space.

## Partial Second Order Ignorance

If a decision maker receives enough information to determine a precise (objective or subjective) probability assessment, the probability region reduces to a single point and the recipient faces a problem of decision making under pure risk. On the other hand, if the recipient can derive no information about the sender's subjective probabilities, the probability region is the whole of probability space, constrained only by the ordinary axioms of probability. In this case, the recipient's problem is equivalent to decision making under pure ignorance.

In the general case, the decision maker knows that the probability distribution over the  $n$  states of nature is somewhere within a constrained region  $\tau$  in the probability space. Each point in  $\tau$  specifies an ordinary probability distribution over the states of nature relevant to the original decision problem. This probability distribution together with the payoff matrix for (state-action) pairs in turn specifies an expected value for each action. Thus each point in the region of possible probability distributions specifies an expected utility for each action. The decision maker knows that the true probability distribution over states of nature corresponds to one

of the points in  $\tau$ , but has no information about the relative likelihood of the points within the region.

This is equivalent to a second order problem of decision making under ignorance. In the second order formulation, the  $n$  discrete states of nature are replaced by a continuum of second order "states," where each second order state is a probability distribution over first order states. If the set of second order states includes the full  $n$ -nomial probability space, then second order ignorance is equivalent to first order ignorance. In partial second order ignorance, the set of possible second order states equals the region  $\tau$  (probability distributions that satisfy the constraints arising from partial knowledge about the probabilities).

The payoff for a particular alternative action under a particular second order state equals the expected payoff for that action under the probability distribution over first order states specified by the second order state in question. The decision maker must choose an alternative action in the absence of any information about the second order probability distribution, except that it is within the set of distributions specified by. Thus, it is necessary to rely upon some other consideration to weight the expected return or regret of each probability distribution, in the same way as in ordinary decision making under ignorance.

It is relatively straightforward to find the corner points of a region in probability space defined by a system of linear constraints and to calculate the expected return arising from each alternative action at each corner point. For any possible probability distribution, the expected return for an action is a linear combination of the expected returns of that action at these corner points. Therefore the maximum and minimum expected return for each alternative action can be found by examining only these corner points.

### **Graphical Analysis When $n=3$**

Suppose that the uncertainty of a decision problem concerns just three possible states of nature. The space of possible probability distributions with respect to these three events forms a planar triangle bisecting the unit cube, as shown in Figure 1. This fact enables us to graph any trinomial probability as a point on a set of triangular coordinates. The three corners of the triangle represent respectively the three trivial probability distributions which assign a probability of 1 to the corresponding states of nature.

Figure 2 shows the 66 regions of probability space that arise from rounding the probability distribution to the nearest decile probability distribution that sums to 1.0. The hexagonal regions represent cases where none of the three rounded probabilities equal zero. The small triangles at the three corners represent the cases when one probability is rounded to 1.0 and the other two are rounded to zero. The pentagons represent cases where one probability is rounded to zero and the other two rounded probabilities are both nonzero.

Figure 3 shows the 166 different regions of probability space that arise from separately rounding each of the three probabilities to the nearest tenth. The hexagonal regions represent cases where the three rounded probabilities add up to 1.0. The small triangles at the three corners represent the cases when one probability is rounded to 1.0 and

the other two are rounded to zero. The trapezoids represent cases where one probability is rounded to zero and the other two rounded probabilities add up to 1.0. The upwards pointing triangles contain probability distributions such as (.86,.06,.08) which when rounded add up to more than 1.0. Finally, the downwards pointing triangles contain probability distributions such as (.84,.03,.13) or (.94,.03,.03) which when rounded add up to less than 1.0.

## Decision Criteria

A logical first step in making a decision under uncertainty is dominance screening. Potter & Anderson [1980] discuss dominance screening in the context of linearly constrained Bayesian priors. Ordinary linear programming can find the maximum and minimum values of the difference between the expected utility (EU) of one alternative and that of another. One alternative decision dominates another if the maximum and the minimum difference have the same sign. (A common error is to assume that the maximum EU of the dominated act must be less than the minimum EU of the act that dominates it. In fact two utility ranges can overlap even if one action always has greater EU than the other for each particular feasible probability distribution.)

Typically, more than one nondominated alternative will remain. To reach a final decision, it is helpful to calculate a figure of merit to represent the attractiveness of each action by a single number. When each state's probability is fully determined, expected utility is the figure of merit. When the probability is underdetermined, there are two approaches to calculating a figure of merit. One approach first evaluates the range of expected utilities possible for an action and then reduces this range to a single representative expected utility. The other approach first reduces the range of probability distributions to a single distribution and then calculates just one expected utility using this representative probability distribution.

## Representative Utility Approaches

The two most common ways to reduce a range of utilities to a single figure of merit are the maximin criterion and the midpoint criterion. Both are special cases of the Hurwicz family of criteria, which use a general weighted average of the minimum and maximum possible utility: maximin uses a weight of 1.0 for the lower bound and midpoint uses a weight of .5. The maximin criterion expresses conservatism in decision making, while the midpoint criterion seeks to optimize average performance.

The extended Hurwicz criterion selects the action for which

$$\alpha * (\max(E(\text{return}))) + (1 - \alpha) * (\min(E(\text{return})))$$

is greatest, where max and min are taken over the set of admissible probability distributions and expectation is taken over states of nature according to each particular distribution. In particular, when the optimism coefficient  $\alpha$  equals zero the extended Hurwicz criterion becomes extended maximin. Assuming that the observed decision maker's probability assessment is correct and remains constant for many iterations of the observing decision maker's action, the long-run average return of the extended maximin criterion's selected action cannot possibly fall below the indicated value, while that of other actions might be below this value for some possible probability distribution.

Similarly, when  $\alpha=.5$  the extended Hurwicz criterion becomes the extended midpoint criterion, while when  $\alpha=1$  it reduces to the extended maximax criterion.

## Representative Probability Approaches

On the other hand, many authors [Jaynes, 1968; Gottinger, 1990] argue that uncertainties about probabilities ought to be resolved as objectively as possible; in other words, without reference to utilities. If this principle is accepted, Gottinger has shown that the only reasonable choice for a representative probability distribution from a range is the distribution whose entropy is highest (the Laplace criterion). These arguments are convincing, but their direct application to the probabilities of states of nature can lead to discarding most or all of the available information. For example, the standard maximum entropy (Laplace) form for a complete order over probabilities is equivalent to the maximum entropy form for total ignorance!

This dilemma can be resolved using a second order maximum entropy concept that preserves more real information while satisfying the requirements that motivate the original maximum entropy concept. [Whalen & Brönn, 1990] Rather than considering the probability distribution over the original set of states, we consider a second-order probability distribution over points in probability space (see Figures 1-3). Applying the maximum entropy principle to this distribution implies that all points in probability space should be considered equally likely. Thus the representative point for a region of probability space is the mean point of that region.

Geometrically, the ordinary maximum entropy distribution for a region in probability space (as in Figures 1 & 2) is the point in the region closest to the center of the entire probability space. The second-order maximum entropy distribution for a region is the center of that region itself. Under total ignorance, the region in question is the entire probability space, and both versions of maximum entropy select the same representative point; i.e. the center of the space.

## Simulation Experiments

[Whalen, 1991] reports a series of simulation experiments that compared the four methods of determining a figure of merit (Maximin, Midpoint, Standard Laplace, and Extended Laplace) using six different information systems:

- (1) the null information system in which the decision maker has no information about probability,
- (2) an ordinal information system in which the decision maker can rank the 3 probabilities from lowest to highest (6 possible messages),
- (3) an information system that informs the decision maker which probability, if any, is above .5 (four possible messages),
- (4) an information system that informs the decision maker which probability, if any, is above 1/3 (6 possible messages),
- (5) an information system that informs the decision maker which probability, if any, above .25 (7 possible messages), and
- (6) the perfect information system in which the decision maker knows the exact probabilities of the three states.

Ten thousand trinomial distributions were generated according to a uniform second-order distribution:  $p_1 = 1-R^2$ ,  $p_2 = S*(1-p_1)$ ,  $p_3 = 1-p_1-p_2$  where R and S are uniformly distributed random fractions. Ten thousand 3X3 utility matrices were randomly generated; the highest utility in each matrix was 100 and the lowest zero, with other utilities uniformly distributed. Each pairing of a criterion with an information system selected an action, and the expected utility of that action was recorded for a total of ten thousand iterations. The lowest mean expected value was 64.255 (maximin criterion, null information system), and the highest mean expected value was 71.748 (perfect information system).

In the present research, the same benchmark set of 10,000 probability distributions and utility matrices was used to examine the performance of the decision criteria using the richer information provided by probabilities rounded to the nearest tenth. The label "Round:1.0" refers to the information system in which rounded probabilities are forced to sum to 1.0, while the "Round:.9-1.1" label refers to the information system which rounds the three probabilities separately. For these two information systems, a fifth decision criterion is also shown; in this criterion, the expected value is simply calculated using the three rounded probabilities. (In the "Round:.9-1.1" system, rounded probabilities are used without regard to whether they sum to 0.9, 1.0, or 1.1.)

Table 2 summarizes the findings of [Whalen, 1991] together with the new experiment (the rows labeled "Round:.9-1.1" and "Round:1.0"). The table shows the mean expected utility of each combination of one of the seven information systems with one of the four decision criterion, expressed as a percentage of the range of mean expected utility from the lowest to the highest; 0% means the lowest observed utility (64.255) and 100% means the highest observed utility (71.745). Thus, the percentages represent the proportion of the maximum benefit that can be derived from probability information.

TABLE 2

	# of Messages	Standard Laplace	Maximin	Midpoint	Extended Laplace	As Rounded
None	(1)	48.0%	0.0%	33.9%	48.0	
Ordinal	(6)	48.0%	81.1%	89.7%	88.6%	
Threshold=1/2	(4)	80.9%	78.0%	86.4%	88.6%	
Threshold=1/3	(6)	48.0%	84.7%	92.4%	92.2%	
Threshold=1/4	(7)	79.0%	85.2%	91.6%	92.3%	
Round:1.0	(66)	95.8%	97.7%	98.56%	98.57%	98.47%
Round:.9-1.1	(166)	98.6%	98.8%	99.1%	99.5%	99.4%
Perfect	(10000)	100.0%	100.0%	100.0%	100.0%	

Several interesting observations can be made based on these results. Not surprisingly, there is a general tendency for the performance of the various techniques to increase with increasing richness of information as measured by the number of alternative messages. But there are some noteworthy exceptions.

The Ordinal information system always leads to poorer performance than the probability threshold  $1/3$  even though both have six messages; furthermore, in the two representative probability approaches (Standard Laplace and Extended Laplace), the six-message Ordinal information system is actually inferior to the four-message information system with probability threshold  $.5$ ! Under the Midpoint criterion, the seven-message information system with threshold  $.25$  is inferior to the six-message information system with threshold  $1/3$ , while under the Standard Laplace criterion the four-message information system with probability threshold  $.25$  outperforms both six-message information systems and the seven-message information system. The only decision criterion which comes close to consistently rewarding richer information with better performance is the Extended Laplace, although even here the performance with ordinal information is very slightly poorer than the performance with information based on a probability threshold of  $.5$ .

Comparing decision criteria under a given information system, the Extended Laplace consistently outperforms the others except in the case of the Ordinal information system, in which it is not quite as good as the Midpoint criterion. Despite strong theoretical endorsements (Jaynes, 1968; Gottinger, 1990), the Standard Laplace is consistently the worst except in the case of the information system with probability threshold =  $.5$ , in which it is better than the maximin criterion. These results seem to imply that the Extended Laplace is the correct way to apply the principle of maximum entropy to problems of this type.

The relationships among the decision criteria are summarized in Figure 4 for the three probability threshold information systems and the two rounded probability information systems. (The horizontal axis, labeled "bandwidth," is the logarithm to the base 2 of the number of messages in the information system, ranging from 2 bits for the four-message system to 7.375 bits for the 166-message system.)

#### REFERENCES

- Gottinger, H. W. (1990), Decision problems under uncertainty based on entropy functionals, Theory & Decision 28:2 p.143-172.
- Ho, Y-C. and Chu, K.C. (1972) Team decision theory and information structure in optimal control problems (Parts I & II). IEEE Trans Autom. Control AC-17, 15-38.
- Jaynes, E. T. (1968), Prior probabilities, IEEE Trans Systems Science & Cybernetics SSC4:3 p.227-241.
- Potter, J. & Anderson, B. (1980), Partial prior information and decisionmaking, IEEE Trans. on Systems, Man, and Cybernetics SMC10:3 p.125-133.
- Resnik, M. (1987) Choices: An Introduction to Decision Theory. U. of Minnesota Press, Minneapolis, Mn.
- Shafer, G. (1976), A Mathematical Theory of Evidence, Princeton Univ. Press.
- Whalen, T. & Brnn, C. (1990), Hurwicz and regret criteria extended to decisions with ordinal probabilities, Proc. 1990 NAFIPS Conf. p.219-222.
- Whalen, T. (1991), Decisions With Underdetermined Probabilities, Proc. 1991 NAFIPS Workshop p.8-12.

Figure 1  
Probability Space ( $n = 3$ )

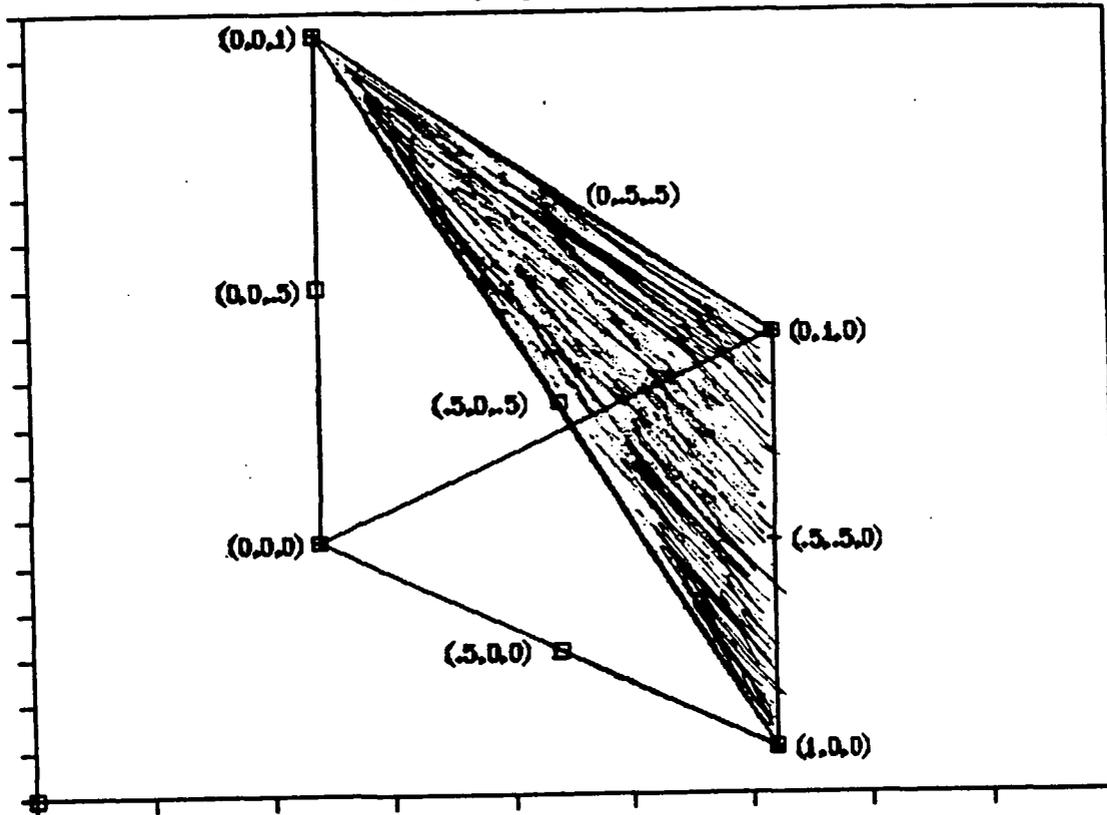


Figure 2  
Decile Probabilities Summing to 1.0

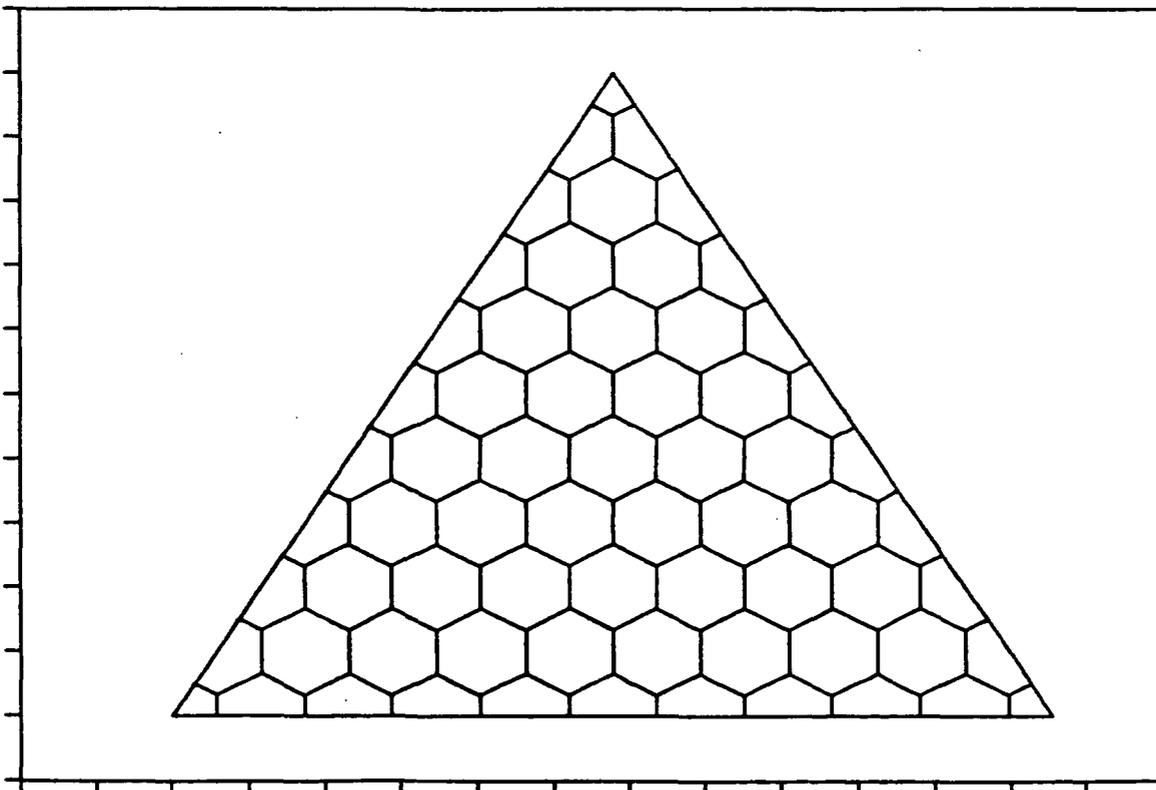
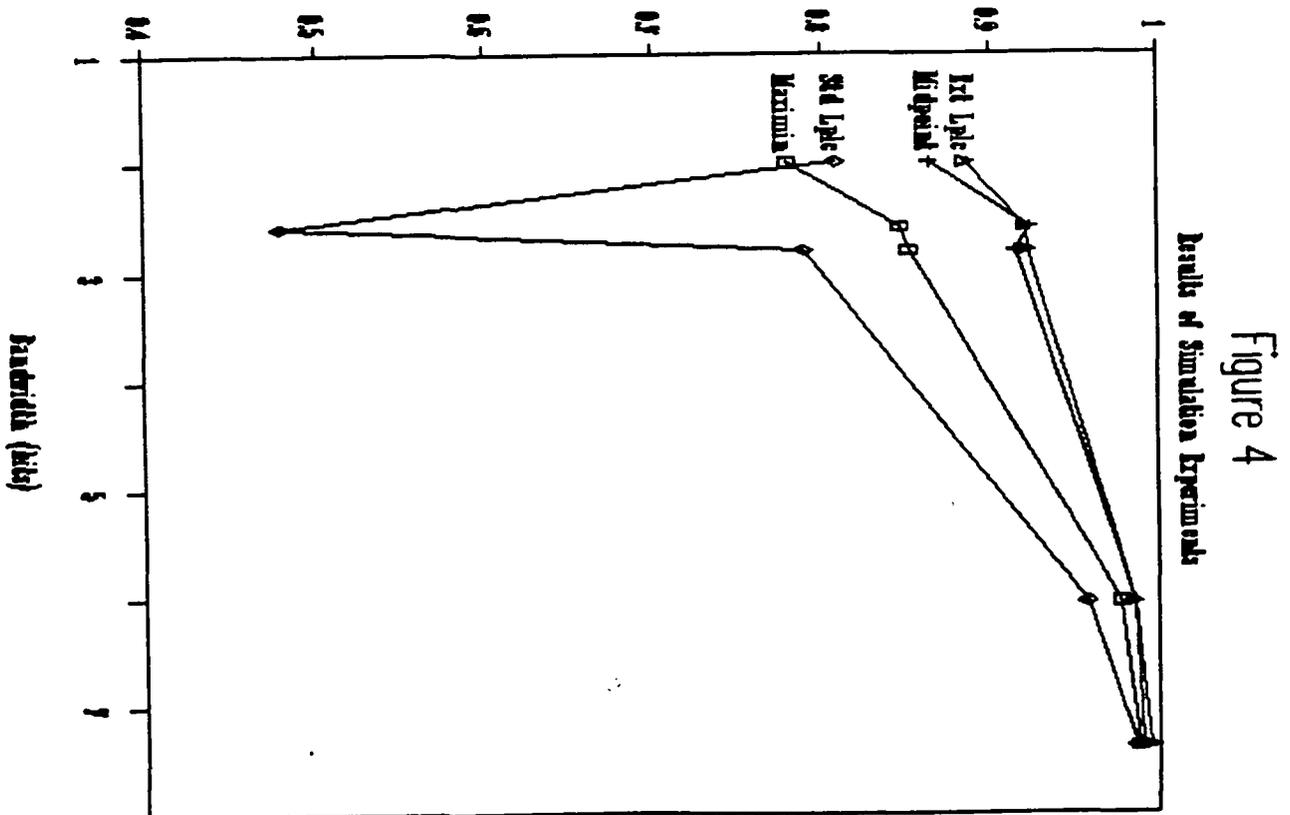
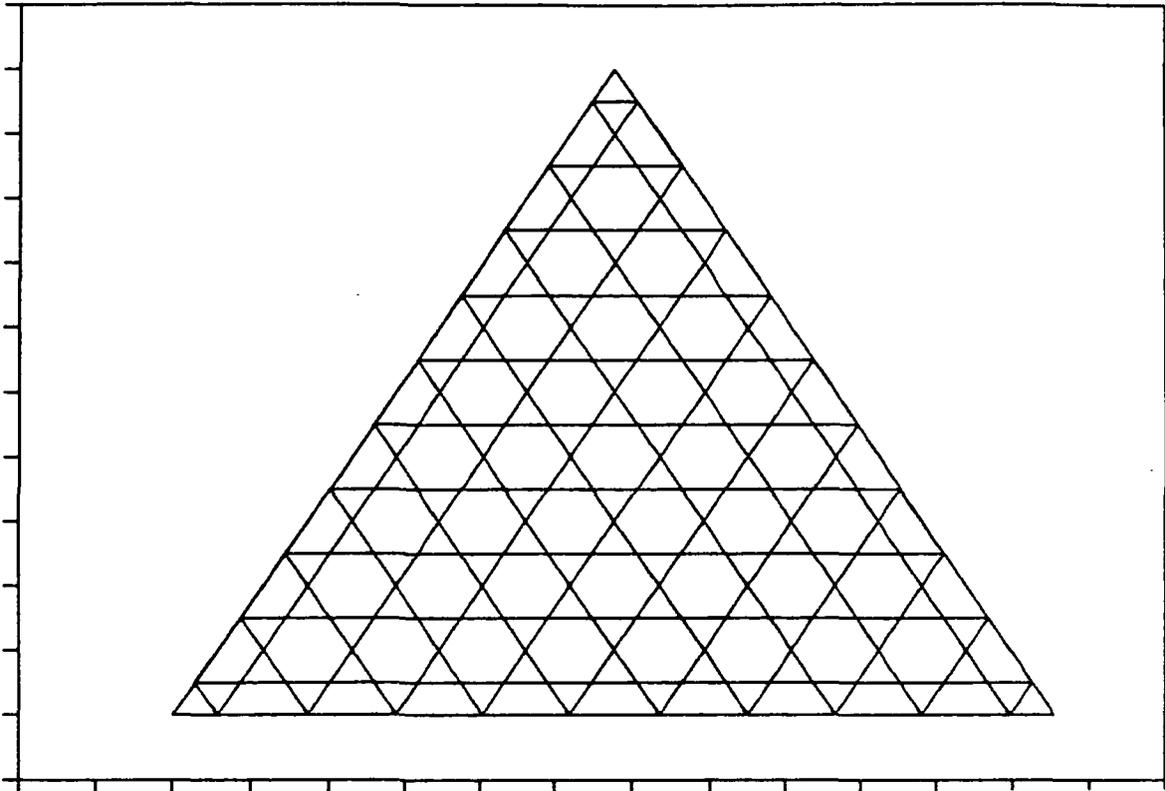


Figure 3  
 Probabilities Rounded to Nearest Tenth



## DISTRIBUTED TRAFFIC SIGNAL CONTROL USING FUZZY LOGIC

Stephen Chiu

Rockwell International Science Center  
1049 Camino Dos Rios  
Thousand Oaks, CA 91360, USA

### ABSTRACT

*We present a distributed approach to traffic signal control, where the signal timing parameters at a given intersection are adjusted as functions of the local traffic condition and of the signal timing parameters at adjacent intersections. Thus, the signal timing parameters evolve dynamically using only local information to improve traffic flow. This distributed approach provides for a fault-tolerant, highly responsive traffic management system.*

*The signal timing at an intersection is defined by three parameters: cycle time, phase split, and offset. We use fuzzy decision rules to adjust these three parameters based only on local information. The amount of change in the timing parameters during each cycle is limited to a small fraction of the current parameters to ensure smooth transition. We show the effectiveness of this method through simulation of the traffic flow in a network of controlled intersections.*

### 1. INTRODUCTION

With the steady increase in the number of automobiles on the road, it has become ever more important to manage traffic flow efficiently to optimize utilization of existing road capacity. High fuel cost and environmental concerns also provide important incentives for minimizing traffic delays. To this end, computer technology has been widely applied to optimize traffic signal timing to facilitate traffic movement.

Traffic signals in use today typically operate based on a preset timing schedule. The most common traffic control system used in the United States is the Urban Traffic Control System (UTCS), developed by the Federal Highway Administration in the 1970's. The UTCS generates timing schedules off-line on a central computer based on average traffic conditions for a specific time of day; the schedules are then downloaded to the local controllers at the corresponding time of day. The timing schedules are typically obtained by either maximizing the bandwidth on arterial streets or minimizing a disutility index that is generally a measure of delay and stops. Computer programs such as MAXBAND [1] and TRANSYT-7F [2] are well established means for performing these optimizations.

The off-line, global optimization approach used by UTCS cannot respond adequately to unpredictable changes in traffic demand. With the availability of inexpensive microprocessors, several real-time adaptive traffic control systems were developed in the late 70's and early 80's to address this problem. These systems can respond to changing traffic demand by performing incremental optimizations at the local level. The most notable of these are SCATS [3,4,5], developed in Australia, and SCOOT [5,6], developed in England. SCATS is installed in several major cities in Australia, New Zealand, and parts of Asia; recently the first installation of SCATS in the U.S. was completed near Detroit, Michigan. SCOOT is installed in over 40 cities, of which 8 are outside of England.

Both SCATS and SCOOT incrementally optimize the signals' cycle time, phase split, and offset. The cycle time is the duration for completing all phases of a signal; phase split is the division of the cycle time into periods of green signal for competing approaches; offset is the time relationship between the start of each phase among adjacent intersections. SCATS organizes groups of intersections into subsystems. Each subsystem contains only one critical intersection whose timing parameters are adjusted directly by a regional computer based on the average prevailing traffic condition for the area. All other intersections in the subsystem are always coordinated with the critical intersection, sharing a common cycle time and coordinated phase split and offset. Subsystems may be linked

to form a larger coordinated system when their cycle times are nearly equal. At the lower level, each intersection can independently shorten or omit a particular phase based on local traffic demand; however, any time saved by ending a phase early must be added to the subsequent phase to maintain a common cycle time among all intersections in the subsystem. The basic traffic data used by SCATS is the "degree of saturation", defined as the ratio of the effectively used green time to the total available green time. Cycle time for a critical intersection is adjusted to maintain a high degree of saturation for the lane with the greatest degree of saturation. Phase split for a critical intersection is adjusted to maintain equal degrees of saturation on competing approaches. The offsets among the intersections in a subsystem are selected to minimize stops in the direction of dominant traffic flow. Technical details are not available from literature on exactly how the cycle time and phase split of a critical intersection are adjusted. It seems that SCATS does not explicitly optimize any specific performance measure, such as average delay or stops.

SCOOT uses real-time traffic data to obtain traffic flow models, called "cyclic flow profiles", on-line. The cyclic flow profiles are then used to estimate how many vehicles will arrive at a downstream signal when the signal is red. This estimate provides predictions of queue size for different hypothetical changes in the signal timing parameters. SCOOT's objective is to minimize the sum of the average queues in an area. A few seconds before every phase change, SCOOT uses the flow model to determine whether it is better to delay or advance the time of the phase change by 4 seconds, or leave it unaltered. Once a cycle, a similar question is asked to determine whether the offset should be set 4 seconds earlier or later. Once every few minutes, a similar question is asked to determine whether the cycle time should be incremented or decremented by a few seconds. Thus, SCOOT changes its timing parameters in fixed increments to optimize an explicit performance objective.

It is problematic that a specific performance objective will be appropriate for all traffic conditions. For example, maximizing bandwidth on arterial streets may cause extended wait time for vehicles on minor streets. On the other hand, minimizing delay and stops generally does not result in maximum bandwidth. This problem is typically addressed by the use of weighting factors; the TRANSYT optimization program provides user-selectable link-to-link flow weighting, stop weighting factors, and delay weighting factors. A traffic engineer can vary these weighting factors until the program produces a good (by human judgement) compromise solution. Perhaps a performance index should be a function of the traffic condition; it may be appropriate to emphasize an equitable distribution of movement opportunities when traffic volume is low and emphasize overall network efficiency when the traffic is congested. In view of the uncertainty in defining a suitable performance measure, the reactive type of control provided by SCATS, where there is no explicit effort to optimize any specific performance measure, appears to have merit. We believe implementing this type of control using fuzzy logic decision rules can further enhance the appropriateness of the control actions, increase control flexibility, and produce performance characteristics that more closely match human's sensibility of "good" traffic management.

In past work performed by Pappis and Mamdani [7], fuzzy logic was applied to control an intersection of two one-way streets. It was assumed that vehicle detectors were placed sufficiently upstream from the intersection to inform the controller about future arrival of vehicles at the intersection. It is then possible to predict the the number of vehicles that will cross the intersection and the size of the queue that will accumulate if no change to the the signal state takes place in the next  $N$  seconds, for  $N = 1, 2, \dots, 10$ . The predicted outcomes are evaluated by fuzzy decision rules to determine the desirability of extending the current state for  $N$  more seconds. Each of the possible extensions is assigned a degree of confidence by the rules, and the extension with maximum confidence is selected for implementation. Before the extended period ends, the rules are applied again to see if further extensions are desirable.

Here we apply fuzzy logic to the general problem of controlling multiple intersections in a network of two-way streets. We propose a highly distributed architecture in which each intersection independently adjusts its cycle time, phase split, and offset using only local traffic data collected at the intersection. This architecture provides for a fault-tolerant traffic management system where traffic can be managed by the collective actions of simple microprocessors located at each intersection; hardware failure at a small number of intersections should have minimal effect on overall network performance. By requiring only local traffic data for operation, the controllers can be installed individually and incrementally into an area with existing signal controllers. Each intersection uses an identical set of fuzzy decision rules to adjust its timing parameters. The rules for adjusting the cycle time and phase split follow the same general principles used by SCATS: cycle time is adjusted to maintain a good degree of saturation and phase split is adjusted to achieve equal degrees of saturation on competing approaches. The offset at each intersection is adjusted incrementally to coordinate with the adjacent upstream intersection to minimize stops in the direction of dominant traffic flow. Through simulation of a small network of streets, the distributed fuzzy control system has shown to be effective in rapidly reducing delay and stops.

## 2. FUZZY RULE-BASED CONTROL

For completeness, a brief introduction to fuzzy rule-based control is presented in this section. At the basis of fuzzy logic is the representation of linguistic descriptions as *membership functions* [8]. The membership function indicates the degree to which a value belongs to the class labeled by the linguistic description. For example, the linguistic description BIG may be represented by the membership function BIG(x) shown in Fig. 1, where the abscissa is an input value and the ordinate is the degree to which the input value can be classified as BIG. In this example, the degree to which the number 80 is considered BIG is 0.5, i.e.,  $BIG(80) = 0.5$ .

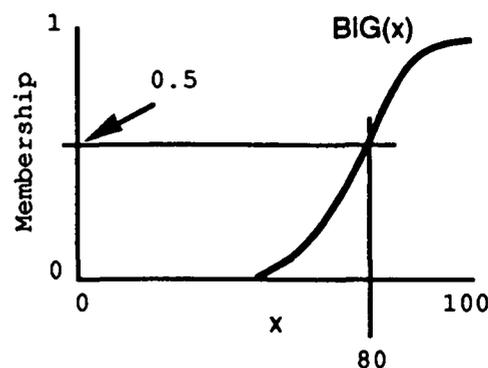


Fig. 1. Membership function defines a linguistic description.

Fuzzy decision rules are typically expressed in the following form:

If  $X_1$  is  $A_{i,1}$  and  $X_2$  is  $A_{i,2}$  then  $U$  is  $B_i$ .

where  $X_1$  and  $X_2$  are the inputs to the controller,  $U$  is the output,  $A$ 's and  $B$ 's are membership functions, and the subscript  $i$  denotes the rule number. For example, a rule for engine control may state "If the speed\_error is negative\_small and the speed\_error\_change is positive\_big, then the throttle\_change is positive\_small." Given input values of  $x_1$  and  $x_2$ , the *degree of fulfillment* (DOF) of rule  $i$  is given by the minimum of the degrees of satisfaction of the individual antecedent clauses, i.e.,

$$DOF_i = \text{Min} \{A_{i,1}(x_1), A_{i,2}(x_2)\}.$$

We compute the output value by

$$u = \frac{\sum_{i=1}^n (DOF_i) B_i^d}{\sum_{i=1}^n (DOF_i)}.$$

where  $B_i^d$  is the *defuzzified* value of the membership function  $B_i$ , and  $n$  is the number of rules. The defuzzified value of a membership function is the single value that best represents the linguistic description; typically, we take the abscissa of a membership function's centroid as its defuzzified value. In essence, each rule contributes a conclusion weighted by the degree to which the antecedent of the rule is fulfilled. The final control decision is obtained as the weighted average of all the contributed conclusions. Although there are several variant methods of fuzzy inference computation, the above method has gained popularity in control applications due to its computational and analytical simplicity.

## 3. TRAFFIC CONTROL RULES

A set of 40 fuzzy decision rules was used for adjusting the signal timing parameters. The rules for adjusting cycle time, phase split, and offset are decoupled so that these parameters are adjusted independently; this greatly simplifies the rule base. Although independent adjustment of these parameters may result in one parameter change working against another, no conflict was evident in simulations under various traffic conditions. Since incremental adjustments are made at every phase change, a conflicting adjustment will most likely be absorbed by the numerous successive adjustments.

### 3.1 CYCLE TIME ADJUSTMENT

Cycle time is adjusted to maintain a good degree of saturation on the approach with highest saturation. We define the degree of saturation for a given approach as the actual number of vehicles that passed through the intersection during the green period divided by the maximum number of vehicles that can pass through the intersection during that period. Hence, the degree of saturation is a measure of how effectively the green period is being used. The primary reason for adjusting cycle time to maintain a given degree of saturation is not to ensure efficient use of green periods, but to control delay and stops. When traffic volume is low, the cycle time must be reduced to maintain a given degree of saturation; this results in short cycle times that reduce the delay in waiting for phase changes. When the traffic volume is high, the cycle time must be increased to maintain the same degree of saturation; this results in long cycle times that reduce the number of stops.

The rules for adjusting the cycle time are shown in Fig. 2 and the corresponding membership functions are shown in Fig. 5. The inputs to the rules are: (1) the highest degree of saturation on any approach (denoted as "highest\_sat" in the rules), and (2) the highest degree of saturation on its competing approaches (denoted as "cross\_sat"). The output of the rules is the amount of adjustment to the current cycle time, expressed as a fraction of the current cycle time. The maximum adjustment allowed is 20% of the current cycle time. The rules basically adjust the cycle time in proportion to the deviation of the degree of saturation from the desired saturation value. However, when the highest saturation is high and the saturation on the competing approach is low, we can let the phase split adjustments alleviate the high saturation. It should be noted that the "optimal" degree of saturation to be maintained by the controller is only 0.55, whereas SCATS typically attempts to maintain a degree of saturation of 0.9. This discrepancy arises from the method of calculating the maximum (saturated) flow value. We derive the maximum flow value based on a platoon of vehicles with no gaps moving through the intersection at the speed limit, while SCATS uses calibrated, more realistic values.

```
if highest_sat is none then cycl_change is n.big;
if highest_sat is low then cycl_change is n.med;
if highest_sat is slightly low then cycl_change is n.sml;
if highest_sat is good then cycl_change is zero;
if highest_sat is high & cross_sat is not high then cycl_change is p.sml;
if highest_sat is high & cross_sat is high then cycl_change is p.med;
if highest_sat is saturated then cycl_change is p.big;
```

*Fig. 2. Rules for adjusting cycle time.*

### 3.2 PHASE SPLIT ADJUSTMENT

Phase split is adjusted to maintain equal degrees of saturation on competing approaches. The rules for adjusting the phase split is shown in Fig. 3 and the corresponding membership functions are shown in Fig. 5. The inputs to the rules are: (1) the difference between the highest degree of saturation on the east-west approaches and the highest degree of saturation on the north-south approaches ("sat\_diff"), and (2) the highest degree of saturation on any approach ("highest\_sat"). The output of the rules is the amount of adjustment to the current east-west green period, expressed as a fraction of the current cycle time. Subtracting time from the east-west green period is equivalent to adding an equal amount of time to the north-south green period. When the saturation difference is large and the highest degree of saturation is high, the green period is adjusted by a large amount to both reduce the difference and alleviate the high saturation. When the highest degree of saturation is low, the green period is adjusted by only a small amount to avoid excessive reduction in the degree of saturation.

### 3.3 OFFSET ADJUSTMENT

Offset is adjusted to coordinate adjacent signals in a way that minimizes stops in the direction of dominant traffic flow. The controller first determines the dominant direction from the vehicle count for each approach. Based on the next green time of the upstream intersection, the arrival time of a vehicle platoon leaving the upstream intersection can be calculated. If the local signal becomes green at that time, then the vehicles will pass through the local

```

if sat_diff is p.big & highest_sat is saturated then green_change is p.big;
if sat_diff is p.big & highest_sat is high then green_change is p.big;
if sat_diff is p.big & highest_sat is not high then green_change is p.med;
if sat_diff is n.big & highest_sat is saturated then green_change is n.big;
if sat_diff is n.big & highest_sat is high then green_change is n.big;
if sat_diff is n.big & highest_sat is not high then green_change is n.med;
if sat_diff is p.med & highest_sat is saturated then green_change is p.med;
if sat_diff is p.med & highest_sat is high then green_change is p.med;
if sat_diff is p.med & highest_sat is not high then green_change is p.sml;
if sat_diff is n.med & highest_sat is saturated then green_change is n.med;
if sat_diff is n.med & highest_sat is high then green_change is n.med;
if sat_diff is n.med & highest_sat is not high then green_change is n.sml;
if sat_diff is p.sml then green_change is p.sml;
if sat_diff is n.sml then green_change is n.sml;
if sat_diff is zero then green_change is zero;

```

*Fig. 3. Rules for adjusting phase split.*

intersection unstopped. The required local adjustment to the time of the next phase change is calculated based on this target green time. Fuzzy rules are then applied to determine what fraction of the required adjustment can be reasonably executed in the current cycle. The rules for determining the allowable adjustment are shown in Fig. 4 and the corresponding membership functions are shown in Fig. 5. The inputs to the rules are: (1) the normalized difference between the traffic volume in the dominant direction and the average volume in the remaining directions ("vol\_diff"); and (2) the required time adjustment relative to the adjustable amount of time ("req\_adjust"), e.g., the amount by which the current green phase is to be ended early divided by the the current green period. The output of the rules is the allowable adjustment, expressed as a fraction of the required amount of adjustment. These rules will allow a large fraction of the adjustment to be made if there is a significant advantage to be gained by coordinating the flow in the dominant direction and that the adjustment can be made without significant disruption to the current schedule.

```

if vol_diff is none then allow_adjust is none;
if req_adjust is very.high then allow_adjust is none;
if vol_diff is very.high & req_adjust is none then allow_adjust is very high;
if vol_diff is very.high & req_adjust is low then allow_adjust is very high;
if vol_diff is very.high & req_adjust is medium then allow_adjust is high;
if vol_diff is very.high & req_adjust is high then allow_adjust is medium;
if vol_diff is high & req_adjust is none then allow_adjust is very high;
if vol_diff is high & req_adjust is low then allow_adjust is very high;
if vol_diff is high & req_adjust is medium then allow_adjust is high;
if vol_diff is high & req_adjust is high then allow_adjust is low;
if vol_diff is medium & req_adjust is none then allow_adjust is very high;
if vol_diff is medium & req_adjust is low then allow_adjust is high;
if vol_diff is medium & req_adjust is medium then allow_adjust is medium;
if vol_diff is medium & req_adjust is high then allow_adjust is low;
if vol_diff is low & req_adjust is none then allow_adjust is high;
if vol_diff is low & req_adjust is low then allow_adjust is medium;
if vol_diff is low & req_adjust is medium then allow_adjust is low;
if vol_diff is low & req_adjust is high then allow_adjust is low;

```

*Fig. 4. Rules for adjusting offset.*

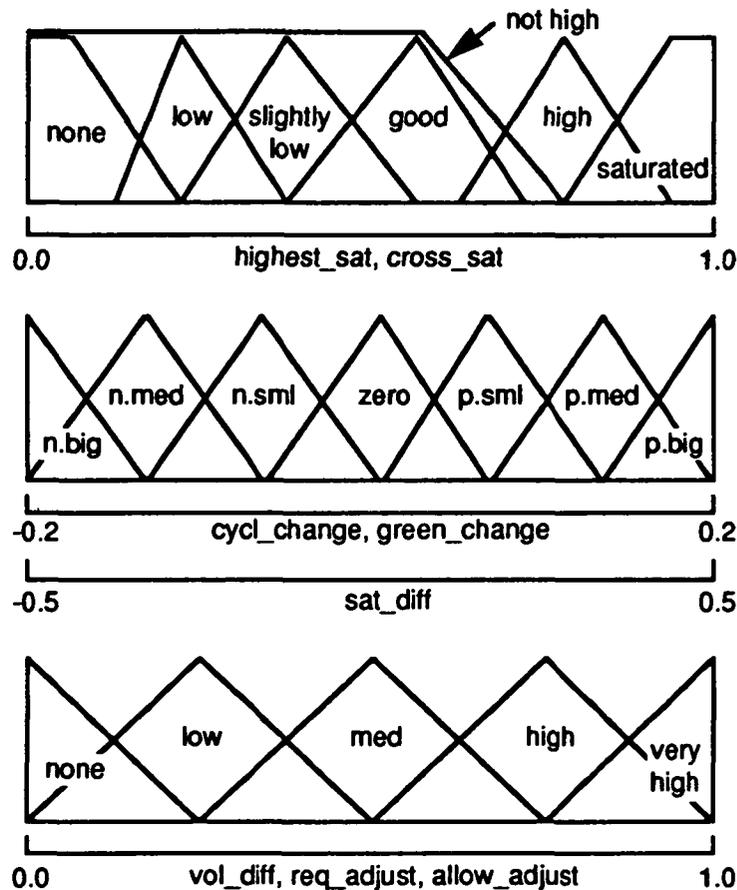


Fig. 5. Membership functions used in rules.

#### 4. SIMULATION RESULTS

Simulation was performed to verify the effectiveness of the distributed fuzzy control scheme. We considered a small network of intersections formed by six streets, shown in Fig. 6. A mean vehicle arrival rate is assigned to each end of a street. At every simulation time step, a random number is generated for each lane of a street and compared with the assigned vehicle arrival rate to determine whether a vehicle should be added to the beginning of the lane. Some simplifying assumptions were used in the simulation model: (1) unless stopped, a vehicle always moves at the speed prescribed by the speed limit of the street, (2) a vehicle cannot change lane, and (3) a vehicle cannot turn. Vehicle counters are assumed to be installed in all lanes of a street at each intersection. When the green phase begins for a given approach, the number of vehicles passing through the intersection during the green period is counted. The degree of saturation for each approach is then calculated from the vehicle count and the length of the green period. At the start of each phase change, the controller computes the time of the next phase change using its current cycle time and phase split values. The fuzzy decision rules are then applied to adjust the time of the next phase change according to the offset adjustment rules; the adjusted cycle time and phase split values are used only in the subsequent computation of the next phase change time.

Figure 7 shows the average waiting time per vehicle per second spent in the network as a function of time. Figure 8 shows the number of stops per minute encountered by all vehicles. For the first 30 minutes of this simulation, all intersections have a fixed cycle time of 40 seconds, a green duration of 20 seconds, and start their phases at the same time. At the end of 30 minutes, intersections A, B, and C shown in Fig. 6 were allowed to adapt their timing parameters according to the fuzzy decision rules. At the end of 60 minutes, all intersections were allowed to adapt. We see that the improvement in waiting time is minimal when only 3 intersections are adaptive. Furthermore, when only 3 intersections are adaptive, the minor improvement in waiting time was obtained at the expense of

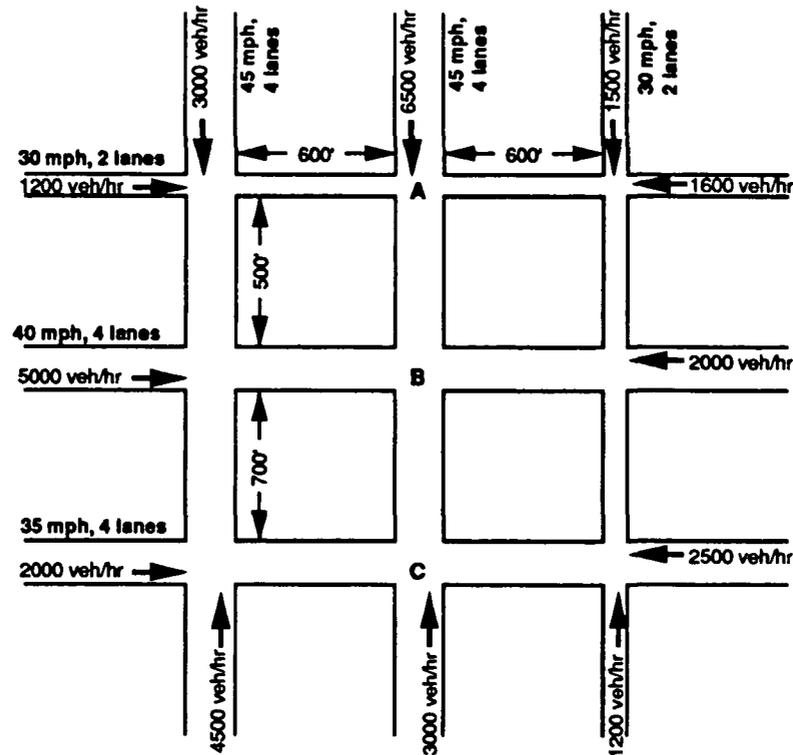


Fig. 6. Network of streets used in simulation.

greatly increased number of stops. This is because the cycle time chosen by the adaptive intersections (around 20 sec) is widely different from the cycle time for the fixed intersections (40 sec). The mismatch of cycle times resulted in a complete lack of coordination between the adaptive intersections and the fixed intersections, where timing adjustments to facilitate local traffic movement can adversely affect the overall traffic movement. When all intersections were allowed to adapt, all intersections attained similar cycle times (around 20 sec), and significant reductions in both waiting time and number of stops were achieved.

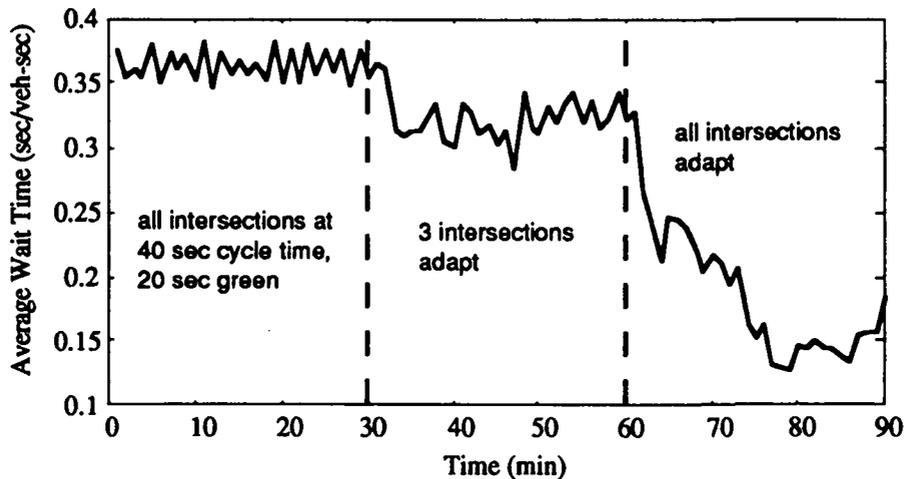


Fig. 7. Average waiting time for the case in which all intersections have an initial cycle time of 40 seconds.

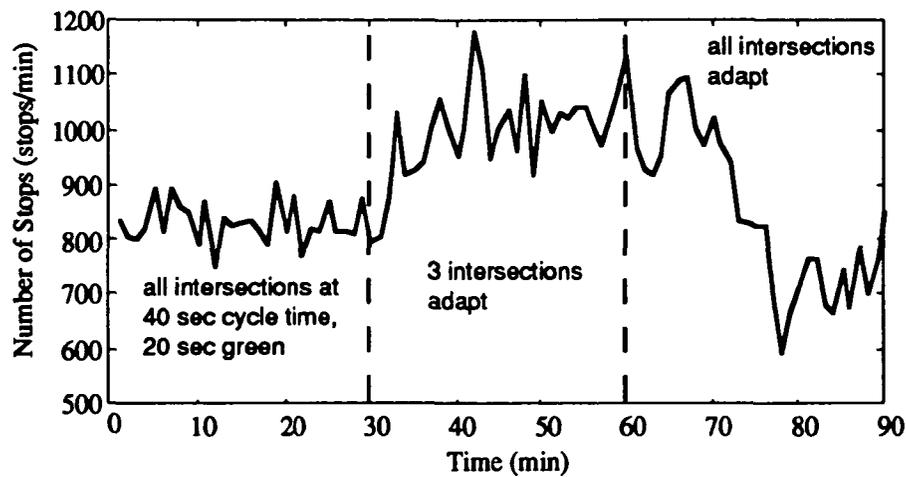


Fig. 8. Number of stops for the case in which all intersections have an initial cycle time of 40 seconds.

Figures 9 and 10 show the results of a simulation performed using the same sequence of events, but with an initial cycle time of 20 seconds and green duration of 10 seconds for all intersections. In this case, significant reductions in both waiting time and number of stops were achieved even when only 3 intersections are adaptive. This is because the cycle time for the fixed intersections closely matches that chosen by the adaptive intersections. Sharing a common cycle time has enabled the 3 adaptive intersections to have immediate positive effect on overall system performance.

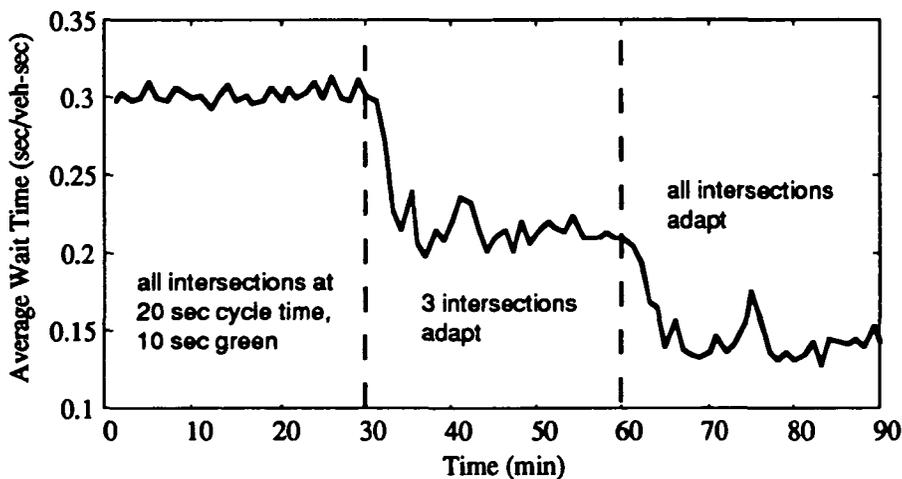


Fig. 9. Average waiting time for the case in which all intersections have an initial cycle time of 20 seconds.

## 5. CONCLUDING REMARKS

We have investigated the use of fuzzy decision rules for adaptive traffic control. A highly distributed architecture was considered, where the timing parameters at each intersection are adjusted using only local information and coordinated only with adjacent intersections. Although this localized approach simplifies incremental integration of the fuzzy controller into existing systems, simulation results show that the effectiveness of a small number of "smart"

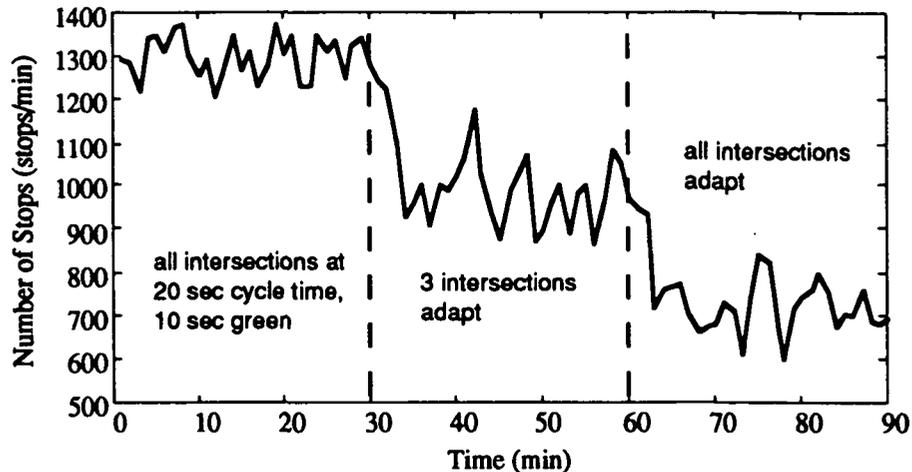


Fig. 10. Number of stops for the case in which all intersections have an initial cycle time of 20 seconds.

intersections is limited if they operate at a cycle time widely different from the rest of the system. In this case, constraining the controller to maintain a fixed cycle time that matches the existing system may provide better overall performance. For the case in which all intersections are adaptive, we need to investigate whether better performance is achieved by constraining all intersections to share a common variable cycle time.

There is much that can be done to further improve the present fuzzy controller, such as including queue length as an input and using trend data for predictive control. The flexibility of fuzzy decision rules greatly simplifies these extensions.

## REFERENCES

- [1] J. Little, M. Kelson, and N. Gartner, "MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks," *Transportation Research Record 795*, National Research Council, Washington, D.C., pp. 40-46, 1981.
- [2] C. Wallace, et. al., *TRANSYT-7F User's Manual (Release 6)*, prepared for FHWA by the Transportation Research Center, University of Florida, Gainesville, FL, 1988.
- [3] A. Sims, "The Sydney Coordinated Adaptive Traffic System," *Proc. ASCE Engineering Foundation Conference on Research Priorities in Computer Control of Urban Traffic Systems*, pp. 12-27, 1979.
- [4] P. Lowrie, *SCATS - A Traffic Responsive Method of Controlling Urban Traffic*, sales information brochure published by Roads & Traffic Authority, Sydney, Australia, 1990.
- [5] J. Luk, "Two traffic-responsive area traffic control methods: SCATS and SCOOT," *Traffic Engineering and Control*, pp. 14-20, 1984.
- [6] D. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time - the SCOOT method," *IEEE Trans. on Vehicular Technology*, vol. 40, no. 1, pp. 11-15, February 1991.
- [7] C. Pappis and E. Mamdani, "A fuzzy logic controller for a traffic junction," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 10, October 1977.
- [8] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28-44, 1973.

528-67

161828

N93-29544

**INTELLIGENT VIRTUAL REALITY**  
**in the**  
**SETTING OF FUZZY SETS**

by

John Dockery  
Center for Excellence in C3I<sup>1</sup>  
George Mason University, Fairfax, Virginia 22030

and

David Littman  
Computer Science Department  
George Mason University, Fairfax, Virginia 22030

**Abstract**

*The authors have previously introduced the concept of virtual reality worlds governed by artificial intelligence. Creation of an intelligent virtual reality was further proposed as a universal interface for the handicapped. This paper extends consideration of intelligent virtual reality to a context in which fuzzy set principles are explored as a major tool for implementing theory in the domain of applications to the disabled.*

**Introduction and Motivation**

This paper is intended as part of an exercise in the generation of requirements from an emergent system design. Following directly upon a brief sketch of the design proposal, design requirements, previously identified, are interpreted in the context of fuzzy sets as potential applications of said subject.

Recently the potential benefits of virtual reality for the disabled have begun to be explored. [CSUN, 1992; Weghorst, 1991] Independent consideration of the potential use of virtual reality to aid the handicapped is being developed by the authors. [Dockery and Littman, 1992] We have proposed the implementation of what we call intelligent virtual reality as a universal interface for the handicapped. The intelligent aspect emerges from what we see as a requirement to wrap such an interface virtual world in an artificial intelligence shell. We shall begin by reviewing the need for such a requirement. Embedded in an end-to-end systems design, it would yield a *total prosthetic environment*. However, before spinning out requirements for a very advanced virtual world, a reality check on virtual worlds may be in order.

What exists presently? The first applications of virtual reality have been primarily in the entertainment field although scientific uses are beginning to be reported in connection with data visualization. For example theoretical chemists use virtual reality to "dock" large molecular species. [Anon., 1992] Likewise NASA is experimenting with telerobotics and remote handling of hazardous materials. Regardless it remains difficult to separate the true promise from the hype.

---

<sup>1</sup> On an Intergovernmental Personnel Act assignment from the Defense Information Systems Agency at GMU on a part time basis.

What is the current technology of virtual reality? With virtual reality the user becomes a participant in the computer display being observed. In virtual reality the user is surrounded by the display. The technology is a mating of high speed graphics, sensors, and fast computing. Introduction of special input devices has opened up the technology to a steadily expanding group of users. Some of these devices include the "data glove", with which the user can control his interaction with the virtual world, or the "eyephones" [sic] by which the user achieves stereo-optical observation of the display.

What differentiates a virtual from a real world? From an environmental viewpoint the essential feature of any virtual world is the designer's ability to suspend the conventional laws of nature and replace them with his own. Thus, users can fly; objects can shrink and expand at will; things can fall up instead of down. In such a world, a handicapped person could reach across a room to pick up an object without ever leaving his place. Why not, then, build a virtual world which compensates for a particular disability by replacing troublesome laws of nature? This is clearly the answer, but on reflection only part of it. That latter portion of the answer lies with the conception of the intelligent virtual reality interface as primarily a device for *intent amplification*. We anticipate an interface communication language which is strongly metaphoric in design. For example, consider an intelligent virtual reality action of "pulling the blinds". It could mean just that. But as a metaphor for controlling intensity, it could mean shutting down a reactor--in an extreme case--depending on context.

The authors also ran a reality check on themselves and their proposal for intelligent virtual reality for the handicapped. Is such a system currently practical? The answer: it can not be done with current technology. Why then propose such a system? The answer: without a conceptual framework for such a design the best the disabled can hope for is some kind of trickle down technology from the entertainment applications which are here now. If everything is so preliminary, then why focus on fuzzy sets? The answer: we will need a strong conceptual framework for stating requirements and for system modelling both of which are amenable to transcription into fuzzy sets as we shall shortly argue.

### **Intelligent Virtual Reality and the Disabled**

For purposes of initial theory development we have assumed the disabled person to have a full and intact cognitive map although this is not an inherent limitation on what we propose. The problem with even a tailored virtual world for the disabled lies with the question of manipulation of that virtual world to some end. Given a limited repertoire of physical moves, a limit on manipulation of a virtual world is anticipated. In fact it could become a further barrier if badly designed. We may set the design situation as follows. Imagine someone with extensive physical handicaps but effectively functioning cognitive and sensory capacities. That is, the person can plan, set goals, monitor the unfolding of a plan, etc., but has great difficulty executing and controlling the motor movements necessary to achieve goals.

Now imagine that the person's environment is populated with intelligent objects, whose purpose is to identify and to carry out the person's intentions. The person communicates intentions to the intelligent objects through an artificially intelligent interface. The latter gives the person access to a combination of (1) computer-generated artificial reality and (2) information captured from the person's environment. The user projects himself into the interface and commands the intelligent objects to do his bidding. In Figure 1 we illustrate the logical flow from which a requirements analysis can begin.

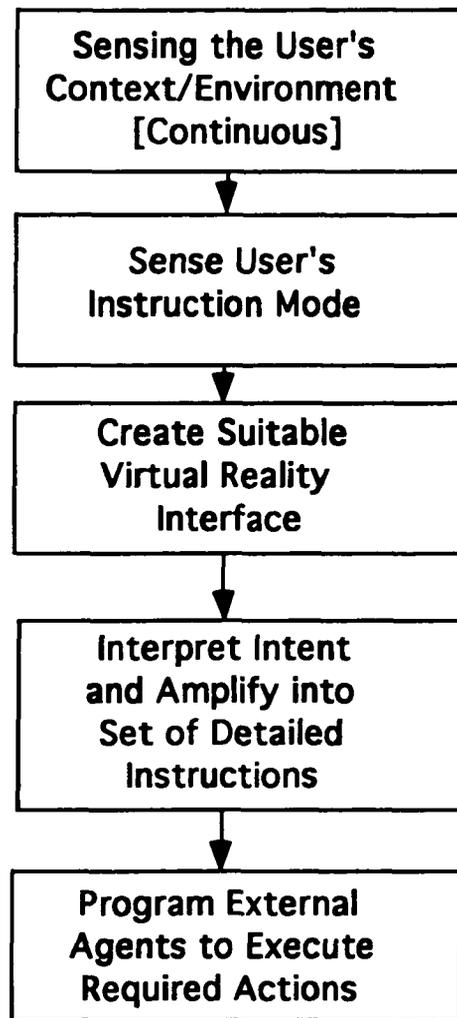


Figure 1: Sequence of Events Necessary to Effect Interaction with the Proposed Intelligent Virtual Reality Interface

The first two boxes in Figure 1 seem straightforward enough, but they mask considerable complexity. One might continue to argue for non-fuzzy implementations if the tasks were simple. When either, or both, the user environment and instruction mode get complex, fuzzy set implementations seem indicated from the outset. The case for fuzzy design principles becomes, if possible, stronger when we remove the restriction for an intact cognitive repertoire. Consider for instance the loss of short term memory. The intelligent virtual reality interface would then have to extract the missing information from records or the environment (real and virtual) after first sensing that amplification of the divined intent required such information.

The second set of two boxes in Figure 1 call for a formal model of intent amplification. We are currently working on an evidence based model. [Dockery and Littman, 1993]. The last box could be considered controversial since robotics has not developed in this direction. However, this paper is an exercise in the statement of requirements; and "smart" external agents are necessary to the concept.

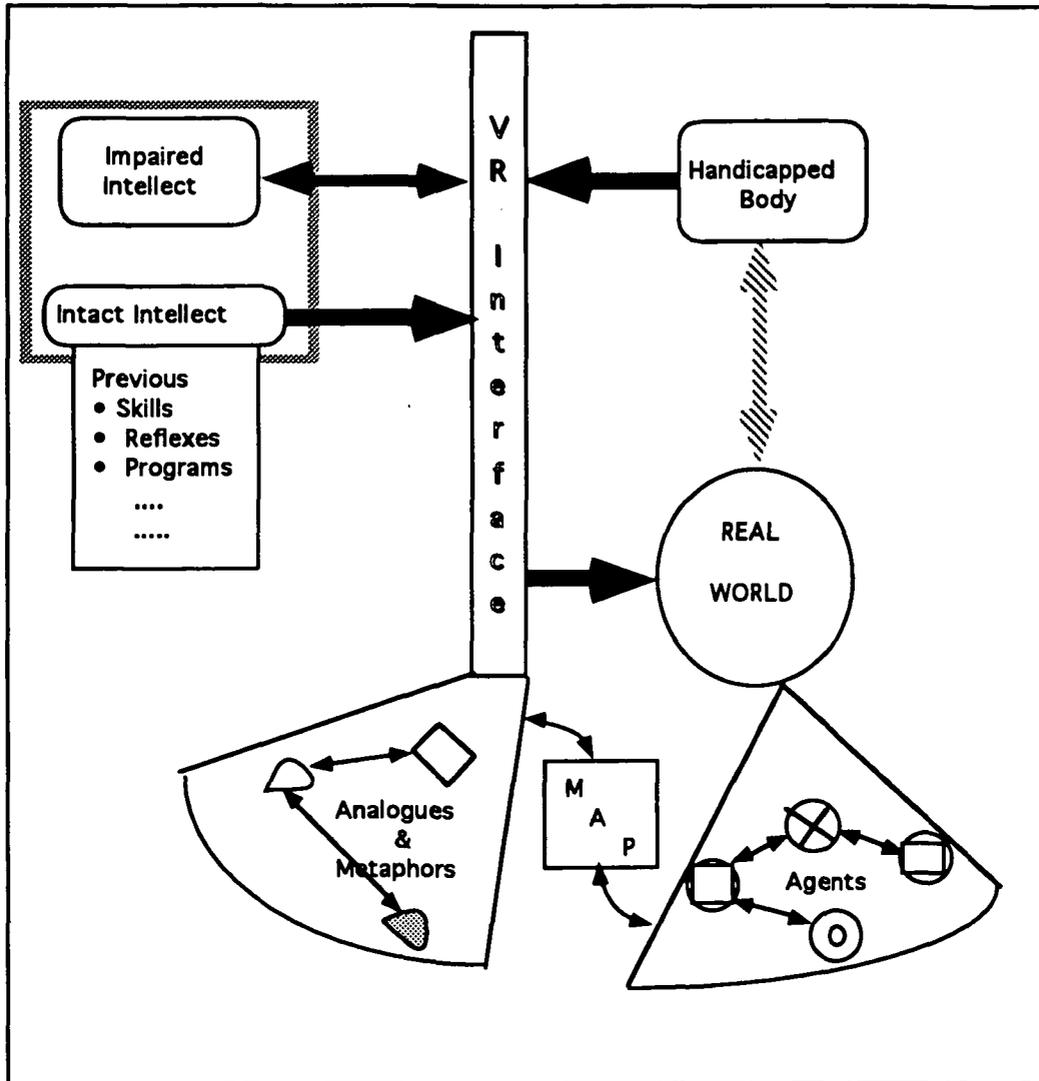


Figure 2: Dynamics of the Interactions between the User and the External World via the Intelligent Virtual Reality Interface.

Seen from a systems engineering viewpoint things are a bit more complicated. Figure 2 above from Dockery and Littman [1992] summarizes the linked intelligent virtual reality interface in more dynamical terms than Figure 1. Attention is called to the reliance on analogue reasoning and metaphorical communication. Both of these are well handled by fuzzy sets. The hatched arrow between the handicapped body and the real world is meant to indicate an impaired and fuzzy communications channel between stated intent and requisite implementation. We turn now to a systematic overview of all the possible requirements which may possibly be met within a fuzzy sets framework.

### Emergent Requirements for Fuzzy Sets Implementation

We have done some preliminary analysis of the required network of technologies necessary to bring about an intelligent virtual reality interface for the handicapped. A fragment

of such analysis can be seen in Figure 3. It shows some possible relationships between fuzzy sets and other technologies.

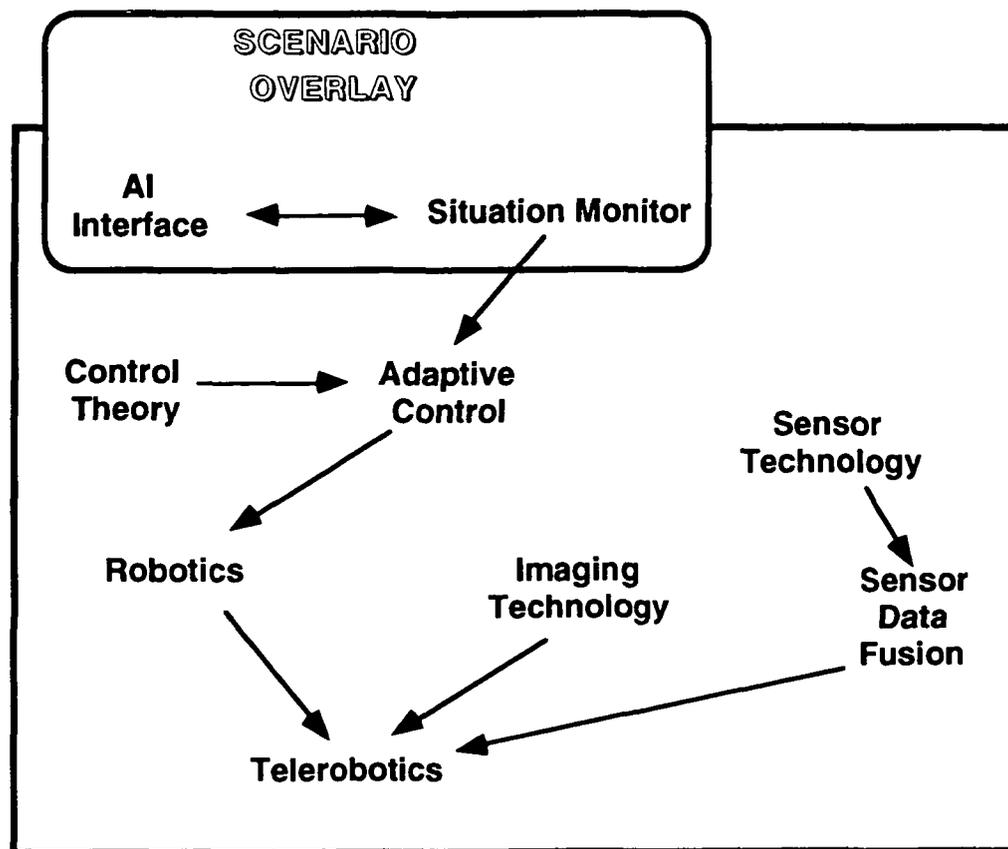


Figure 3: Example of Networked Technologies Needed to Implement an Intelligent Virtual Reality Interface for the Handicapped

To see why fuzzy set theory can be expected to play such an important role in the implementation of intelligent virtual reality for the handicapped, we look first to the assumption of intent amplification. We have already asserted that the signaling and interpretation of intent is basically a fuzzy process. Where else might the fundamental interactions be best described with the help of fuzzy sets? Including the aforementioned relationships with intent, they arise from at least the set of design foci, which are first listed in Table II, and then discussed. But first we assert that there are globally valid reasons for expecting fuzzy sets to play an important role in design of an intelligent virtual reality for the handicapped. They are summarized in Table I below.

**Table I**

**Global arguments for use of fuzzy set theory in design of an intelligent virtual reality for the handicapped.**

- Both input, e.g. intent and output, e.g. telerobotic commands are inherently multi-valued. Moreover, depending on context that may be inherently imprecise.
- Virtual reality worlds are excellent examples of instantiated possibilities rather than probable variations on real worlds. [Although the latter can not be gainsaid, the emphasis in intelligent virtual reality is the possible.]
- In the interpretation of intent there are simply too many real life instances to write rules for them all. Therefore, fuzzy reasoning is suggested for interpolating between and/or extending the rule base.
- Similarity transforms and reasoning by analogy, both well treated by fuzzy sets, are required in dealing with goal determination from intent signaling.
- Reasoning under uncertainty will certainly be important.
- Soft computing recently proposed by Zadeh [1992] appears useful for interpolation requirements sure to be present.

Some candidate design issues, which incorporate one or more of the global arguments follow in Table II. We turn finally to a series of brief expositions on emergent applications.

### **Commentary on Emergent Applications to Intelligent Virtual Reality**

#### **REASONING**

Above all the reasoning about intent and translation into overt action by agents is a hierarchical process. The process in all but the most trivial examples is non-monotonic. At the highest level is the requirement for an overall awareness function related to the imputed goal. Although crisp logic may actually drive the agents behavior, the choice of which crisp logic that is appropriate in a given time interval has been shown in simulation to be well treated by fuzzy logic. Likewise the choice of reasoning method seems to require a cross between deduction and intuition of the sort typically referred to as abductive reasoning methods. Aspects of abductive inference may benefit from fuzzy algorithms.

Adoption of various pairs of norms and co-norms effectively creates hierarchically arranged models of the decision maker operating through the intelligent virtual reality interface. Thus, the user could chose between risk taking and risk adverse solutions to goal satisfaction, itself a fuzzy concept.

**Table II**

**Candidate elements of the intelligent virtual reality for fuzzy set applications**

- Sensor fusion of real world data.
- Signaling of intent.
- Interpretation of intent by the artificial intelligence shell.
- Planning, execution, monitoring as well as replanning and adaptation.
- Design of a "forgiving" implementation of external actions resulting from interpretation of intent.
- Description of virtual reality metaphors via linguistic variables.
- Design of a virtual reality world according to fuzzy laws of nature; or equivalently, a physics with fuzzy equations.
- Incorporation of an "awareness" function at the top level of design to answer the question: "How am I [the interface system] doing."
- Strong requirement for learning which suggests linking fuzzy set controllers with neural net hardware.
- Fuzzy logic controllers for the smart robotic agents.

OPERATION OF A VIRTUAL REALITY WORLD

There appears to be a requirement for a fuzzy qualitative physics such as that discussed by Demtchenko [1991] by which to describe some of the laws of nature in the intelligent virtual reality. For example given an interpretation by the intelligent virtual reality artificial intelligence shell that force need be applied, it is anticipated that the appropriate statement is not the classical  $F = ma$  but rather "some moderate force" is required to accelerate an approximate mass to a modest velocity adequate to carry out the task, as for example forcing open a stuck door. One is reminded in this instance of why super tankers can't dock--1/4 mile per hour times a loaded super tanker mass equals trouble.

Building a world based on fuzzy qualitative physics may signal a real application for fuzzy differential equations in which both coefficients and variables are fuzzy entities. In general we will be dealing with fuzzy dynamical systems. [Buckley, 1991]

COMMUNICATION BETWEEN THE INTELLIGENT VIRTUAL REALITY ENVIRONMENT AND USER

As has already been stated, the anticipated mode of communication is by metaphor. This would almost certainly involve complex sets of similarity transforms. Since you can write your own rules in a virtual reality world, consideration of communication leads to a concept of a fuzzy

self-adaptive interface. There is no reason why the color intensity in the intelligent virtual reality could not indicate the quality of the data input to take a solution which has its answer in another design dimension. That dimension is user adaptation to some rather alien virtual reality implementation.

Whatever fuzziness appears in the interface or in the controller logic of the smart agents results in crisp actions. However, the evaluation of the crisp action in terms of movement toward the goal derived from the signaled intent is still fuzzy.

### DATA INPUT INTO THE INTELLIGENT VIRTUAL REALITY INTERFACE

Generation of events in the intelligent virtual reality interface are controlled by data fusion of real world data and possibly stored data as well. A possible military analogue is collection and evaluation of intelligence information. Real world data are by nature fuzzy since some of them will be derived by inference. Even stored data about objects in the environment are perhaps better stored as possibility functions. For example even without weighing it, it is not very possible that a book will weigh more than a couple of pounds. The question of fuzzy input data also involves practical limitations on number and precision of external sensors driving the creation and operation of the intelligent virtual reality interface world.

### RELATED TO ALLIED APPLICATIONS

Two decision science areas which may be very successfully allied to fuzzy set implementation come to mind early in the requirements generation phase.

- Neural Nets combined with fuzzy control logic to tackle questions of learning and adaptation.
- Bayesian inference net applications.

### **Summary**

We have introduced the possibility that fuzzy set theory applications could play a significant role in the design and implementation of a universal interface for the handicapped. That interface and the total system concept in which it is embedded does not exist. Therefore, this paper addressed top level requirements for such a system and interface in terms of opportunities for fuzzy set mathematics and logic.

### **References**

Anonymous, "A Visit to a Virtual World", *Science* **256**, 3 April 1992, p. 45.

Buckley, J. J., "Fuzzy Dynamical Systems I: The Dynamic Case", Proceedings of the International Fuzzy Systems Association Conference [IFSA '91] held in Brussels, Belgium, July 1991 at the Vrij Universiteit.

CSUN, Mini-conference presentations on the subject of virtual reality and the handicapped at the March 19-21, 1992 California State University at Northridge [CSUN] Conference on "Technology and Persons with Disabilities" in Los Angeles. As yet unpublished. Information c/o the Office of Disabled Student Services, U. of California at Northridge

Demtchenko, A., "Expert Systems Based on Fuzzy Qualitative Physics", Proceedings of the International Fuzzy Systems Association Conference [IFSA '91] held in Brussels, Belgium, July 1991 at the Vrij Universiteit.

Dockery, J and Littman, D., "Intelligent Virtual Reality as a Universal Interface for the Handicapped", Proceedings of a Conference on Virtual Reality held in Montpellier, France, March 1992. Sponsors were Informatique 92, c/o EC2, 269 rue de la Garenne, 92024 Nanterre Cedex, France.

Dockery, J. and Littman, D., "Fuzzy Logic Implementation of Intent Amplification in Virtual Reality", (submitted to Fuzzieee, 1993).

Weghorst, S, "Virtual Reality: Tailoring Worlds to Individuals", Keynote address to the Conference of the Media and Technology Division of the Council for Exceptional Children (Reston, Va.), Albuquerque, NM, Dec. 1991.

Zadeh. L., An address at Goddard Spaceflight Center to a Workshop on Fuzzy Set Logic, Greenbelt, Maryland, May 7, 1992.

# Comparison of Crisp and Fuzzy Character Networks in Handwritten Word Recognition

Paul Gader, Magdi Mohamed, and Jung-Hsien Chiang

*Department of Electrical and Computer Engineering  
University of Missouri-Columbia  
Columbia, MO 65211*

529-63

161829

p. 9

---

This research was supported by the Environmental Research Institute of Michigan and the United States Postal Service Office of Advanced Technology

---

## ABSTRACT

Experiments involving handwritten word recognition on words taken from images of handwritten address blocks from the United States Postal Service mailstream are described. The word recognition algorithm relies on the use of neural networks at the character level. The neural networks were trained using crisp and fuzzy desired outputs. The fuzzy outputs were defined using a fuzzy k-nearest neighbor algorithm. The crisp networks slightly outperformed the fuzzy networks at the character level but the fuzzy networks outperformed the crisp networks at the word level.

## INTRODUCTION

Handwritten word recognition by computer is a very difficult task. Although considerable research has been performed in character recognition, not much has been done in word recognition. Interest has picked up lately, as can be seen by viewing the contents of the proceedings of recent conferences in these areas [1,2,3,4]. Even in the machine-printed case, word recognition consists of more than just reading the individual characters in the word [5,6,7]. People are able to read words with illegible and ambiguous characters. Many alphabetic characters are ambiguous when read out of context. In fact, the same pixel pattern can represent different characters in different words. Furthermore, multiple characters can look like characters. For example, the "l" in the image of the word "Portland" in Figure 8 could be an "H".

The implication of this is that high recognition rates may not be the ultimate goal of an alphabetic character classifier that is to be used in word reading. Accurate representation of ambiguity is more important. Thus if a certain character in the training set is called a "u" but could be either a "u" or "v", then the desired output of a classifier for that sample should reflect the ambiguity. That is, the notion of fuzzy set membership of characters is very natural and important in the development of character classifiers to be used in word recognition.

In this paper, we discuss a handwritten word recognition algorithm that uses neural network classifiers on the character level to attempt to read a word. The algorithm is designed to read words that are amenable to segmentation-based approaches; handprinted and well-formed cursive words. We discuss experiments involving the using of assigning desired outputs in the training of the neural networks using a fuzzy k-nearest neighbor algorithm. We compare the use of such networks with crisply trained networks at the character level and at the word level. Our experimental results indicate that the fuzzy output networks do not perform as well on the character level but perform better at the word level.

## CHARACTER TRAINING

### FEATURES

Currently, a character image is size and skew normalized to size 24 x 16. In the first stage of processing a normalized image is input and a set of eight feature images are generated as output. Each feature image corresponds to one of four directions (east, northeast, north, and northwest) in either the foreground or the background. Each feature image has an integer value at each location that represents the length of the longest bar that fits at that point in that direction. An example of the background and foreground feature images corresponding to the east-west directions for an upper-case "B" is shown in Figure 1.

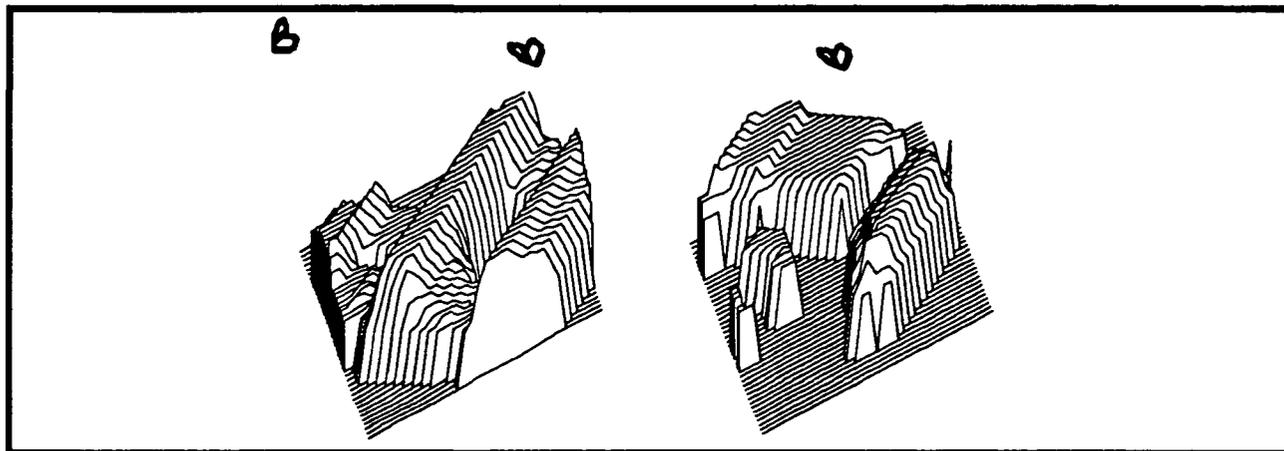


Figure 1. An upper case "B" and the foreground and background bar-feature images corresponding to the east/west directions.

The next stage of processing consists of generating feature vectors from the feature images using the technique of overlapping zones. Fifteen zones are being used; each zone is of size 8 x 8. The zones are maximally overlapping. Zone 1 has its upper left hand corner at position (1,1), zone 2 at position (1,5), ..., zone 4 at position (5,1), etc.

The values in each zone in each feature image are summed. The resulting sums are then normalized between 0 and 1 by dividing by the maximum possible sum in a zone. Thus, the resulting feature vector is of dimension  $15 \times 8 = 120$  and has values between 0 and 1

### NETWORK STRUCTURE

We trained separate networks for upper and lower case characters. The networks are four-layered, fully connected, back-propagation networks. Each has input, output, and two hidden layers. Each hidden and output unit has a bias. In this experiment we used 120 input units, 65 units for the first hidden layer, 39 for the second hidden layer, and 26 outputs units. The SAIC neurocomputer [9] was used for training.

### COMPUTATION OF DESIRED OUTPUTS

The desired outputs for the crisp networks were set by setting the desired output for the true class to 0.4 and the desired outputs for all other classes to -0.4. The desired outputs for the fuzzy networks were set using a fuzzy k-nearest neighbor algorithm described below.

The fuzzy K-nearest neighbor algorithm we used to assign desired outputs to the characters in the training sets was suggested by Keller et al [10]. The idea is to assign membership based on the percentage of characters in each class among the neighbors of a training sample. Each of our training samples has a true class associated with it, that is, what character the original writer intended to form when writing the character. We do not allow the desired output for the true class to be lower than the desired output for any other class.

We chose to use the twenty nearest neighbors of a training sample using Euclidean distance. The samples

were represented by their feature vectors, thus the distance is being measured in 120-dimensional space. Some examples of desired outputs computed using this Keller's algorithm are shown in Figure 2.

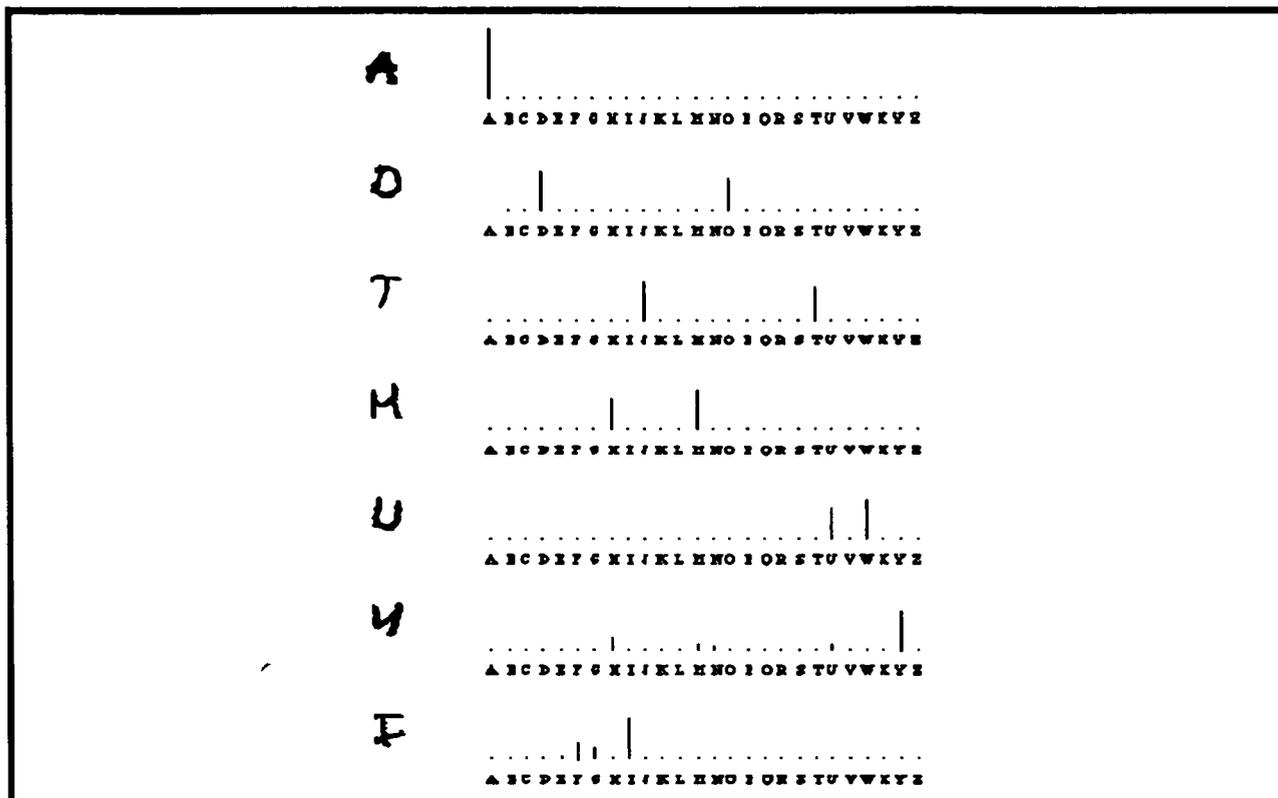


Figure 2. Some uppercase characters and their fuzzy set memberships as determined by the fuzzy k-nearest neighbor algorithm.

## TRAINING DATA

Two sources of data were used to construct our training and test sets: characters from the NIST data set and characters from images of handwritten address blocks obtained from the USPS, which we refer to as HWAB data. Both sets of characters were extracted from images using automatic and manual extraction techniques[11].

We are interested in reading words in address blocks and therefore the HWAB data is more important to us. Some classes of characters are not well represented in the HWAB data. For example, we were only able to find seven lower case "j"'s in a set of 3000 address blocks. We used the NIST data to fill in the "gaps" in the HWAB data.

Specifically, the training and testing sets for the normalized neural networks consisted of 250 characters from each class. We used as many characters as possible from each class using HWAB data. Thus, if only 300 characters were available from a given class, then we would use 150 in the training set and 150 in the test set. The difference between the number of characters available from the HWAB data and 250 was made up using NIST data. The results are shown in Table 1.

Table 1 HWAB character data correct classification rates.

UPPER CASE	Training Set	Test Set	RMS error	Learning Cycles
Crisp	92.98%	85.91%	0.063959	2158
Fuzzy	88.12%	84.03%	0.050164	1778
LOWER CASE				
Crisp	88.43%	82.15%	0.071576	3255
Fuzzy	86.91%	80.72%	0.050958	2014

At first glance, it may seem that the networks trained with fuzzy outputs are not doing as well. However, several interesting points can be made concerning the above results.

The networks trained with fuzzy outputs converged to lower RMS errors than their crisp counterparts in far fewer learning cycles. Furthermore, the drop in performance was somewhat less for the fuzzy output networks than for the crisp output networks, indicating that the fuzzy output networks may be more robust.

Another view of the results at the character level is given by the sequence of graphs in Figures 3 - 6. We have translated the output values linearly between 0 and 10 and quantized them to integer values. We then constructed histograms of the number correct and incorrect in each bin for the crisp and fuzzy case (Figures 3 and 4) and for the percentage of answers correct and incorrect in each bin (Figures 5 and 6). The number of answers in the high value bins is smaller for the fuzzy case. However, the percentage of answers that are correct in the higher value bins is higher for the fuzzy case. This indicates one can trust answers with higher values more in the fuzzy case.

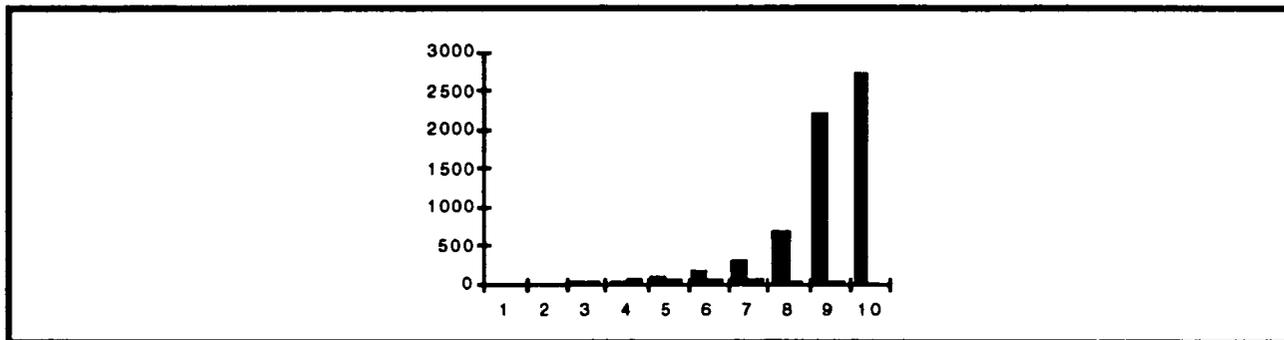


Figure 3. Number Correct/Incorrect by confidence bin - crisp case

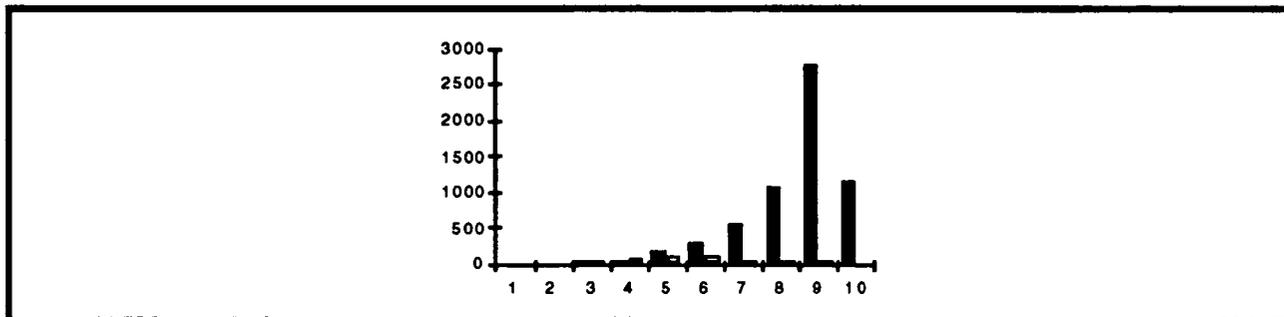


Figure 4. Number Correct/Incorrect by confidence bin - fuzzy case

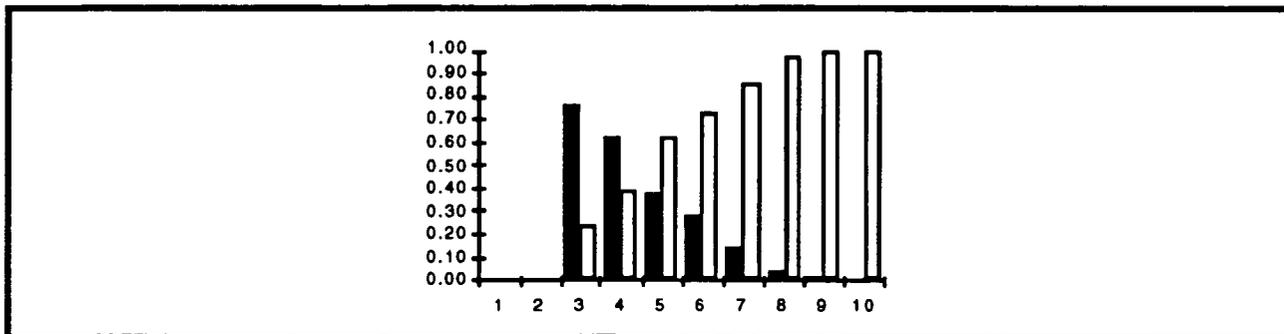


Figure 5. Percent Correct/Incorrect by confidence bin - crisp case

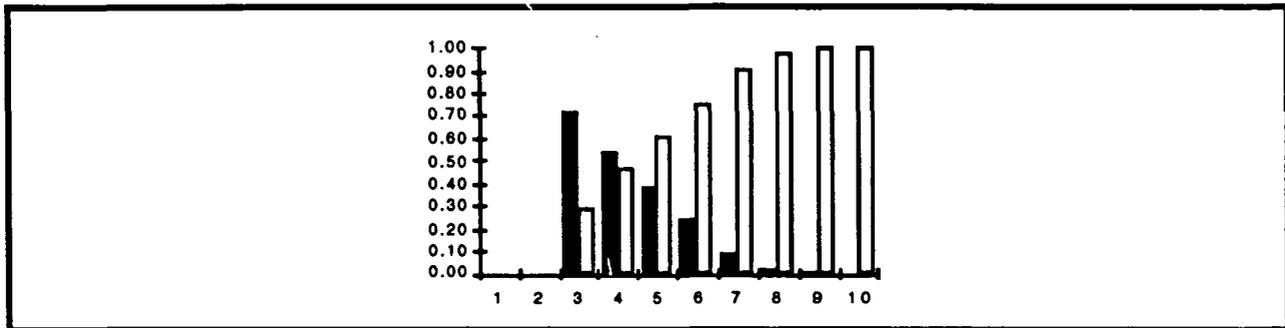


Figure 6. Percent Correct/Incorrect by confidence bin - fuzzy case

## WORD RECOGNITION ALGORITHM

### OVERVIEW

The word recognition algorithm used is based on image segmentation and dynamic programming matching. The inputs to the algorithm are a binary image of a word and a lexicon. The lexicon is a list of strings representing all possible candidate words for the image.

The approach is based on segmenting the word image into character images, matching the character images against the characters in the word strings in the lexicon, and assigning a confidence to each string in the lexicon based on an aggregation of the confidence of each of the character segments. Unfortunately, it seems impossible to correctly segment a word image into characters without the use of recognition because of the ambiguity of characters and multiple characters mentioned in the introduction above. We therefore need to generate multiple segmentation hypotheses.

The image of the word is segmented into primitive segments. Each primitive segment is generated from a subimage of the original word and ideally consists of either a character or a part of a character. The correct segmentation can be thought of as a path through the space consisting of all primitive segments and their legal unions. Dynamic programming is used to find the best cost path. The cost of a path is currently defined to be the sum of the character confidence of each segment along the path. A more detailed description is given in the following sections. An overview of the system is shown in Figure 7.

As noted in the introduction, this system is being designed to read words that are mainly handprinted; segmentation-based techniques do not seem appropriate for cursive words. Thus, our system contains a module to filter out words that are look too much like cursive words.. The filter is set loosely so that we do process a significant number of cursive words.

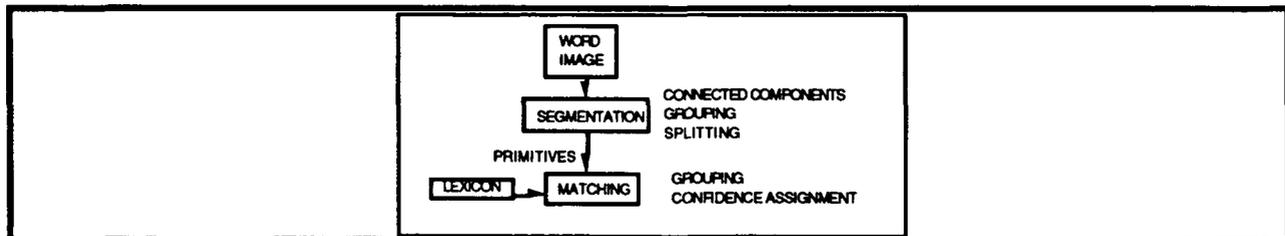


Figure 7. Overview of word recognition system.

### SEGMENTATION

The segmentation process is a refinement of that described in [12]. We describe it briefly here. The connected components in the image are computed. Punctuation is detected and removed. Some simple grouping of horizontal bars is performed. The result is an initial segmentation. The initial segments generally consist of images of one or more characters. Those that consist of more than one character need to be split.

Each segment in the initial segmentation is passed through a splitting module. The splitting module uses the distance transform to detect possible locations to split initial segments into characters. The distance transform

encodes each pixel in the background using the distance from the stroke. Roughly speaking, split paths are formed that stay as far from the stroke as possible without turning too much. Thus, the distance transform can be thought of as a cost function and the process of splitting one of finding an optimal path. Heuristics are used to define starting points for the paths based on the shape of the image. Heuristics are also used to modify the distance function; for example, holes are encoded as uniformly high cost and "fat" strokes are encoded as low cost.

The result of splitting and initial segmentation is a sequence of subcharacter images which are postprocessed to correct for images that are very small or very complex. This yields the primitive segments as in Figure 8. Unions of primitives are not formed unless required to match strings in the lexicon.

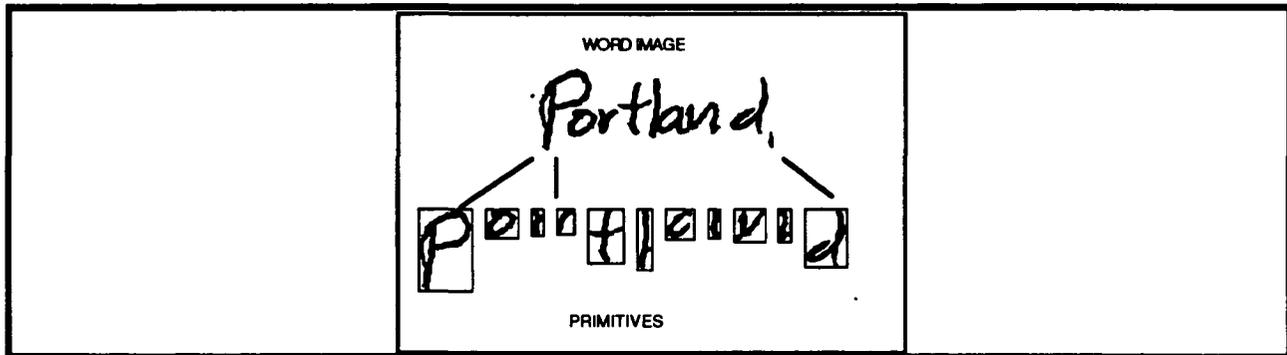


Figure 8. A word image and its primitive segments.

#### DYNAMIC PROGRAMMING MATCHING

The core of the dynamic programming algorithm is a module that takes a word image, a string, and a list of the primitives from the word image and returns a confidence value between 0 and 1 that indicates the confidence that the word image represents the string. Dynamic programming is used to find the best path through the space of primitives and legal unions of primitives. The best path depends upon the method to evaluate each node in the path. The value of each node here is currently provided solely by the neural networks described above. The value of a path is computed by averaging the values of the nodes.

The algorithm is implemented using a matrix approach. For each string in the lexicon, an array is formed. The rows of the array correspond to the characters in the string. The columns of the array correspond to primitive segments. The  $ij$  element in the array is the value of the best match between the first  $i$  characters in the string and the first  $j$  primitive segments. This value may be  $-\infty$  if there is no legal match.

Let the primitive segments of the image be denoted by  $S_1, S_2, \dots, S_p$ . Let the characters in the string be denoted by  $C_1, C_2, \dots, C_w$ . Let  $m(c,s)$  be a function that takes a character  $c$  and a segment image  $s$  computes the confidence that  $s$  represents  $c$ . The  $ij$  element of the array is computed as follows:

If  $i = 1$ , (matching against the first character) Then

$$v(1,j) = \begin{cases} m\left(\bigcup_{h=1}^j S_h, C_1\right) & \forall j \text{ such that } \bigcup_{h=1}^j S_h \text{ is legal} \\ -\infty & \text{otherwise} \end{cases}$$

If  $i > 1$  Then

$$v(i,j) = \begin{cases} \max_k (v(i-1,k) + m\left(\bigcup_{h=k}^j S_h, C_i\right)) & \forall k,j \text{ such that } \bigcup_{h=k}^j S_h \text{ is legal} \\ -\infty & \text{otherwise} \end{cases}$$

The match value is currently computed by running the upper and lower case neural networks on the segment and retrieving the output value associated with the character for each. Currently the maximum value is taken (except if the character is the first character in a word, in which case a capital letter is much more likely in our application). The neural network can be either the crisp output or the fuzzy output type.

A union of two segments is considered legal if it the two segments pass a sequence of tests. The tests measure closeness and complexity.

## WORD TESTING RESULTS

The data used in these experiments were obtained from images of real mail pieces from the United States Postal Service mailstream. They consist of binary images of city and state names.

A test was run on 500 of the images as described above. The lexicon size for our results was 457. The results are shown in tables 2 and 3. The tables require some interpretation. Recall that, since this system is not designed to read all words, there is a check that rejects some words. The response percentage in the tables indicates the percentage of the 500 words that the system decided to process. The word recognition value returns a confidence value between 0 and 1 for each string in the lexicon. We can further decrease the number of responses by thresholding this confidence value.

For each value of a threshold, we compute the number of times the correct string was the top choice, the second highest choice, etc. The rows labeled 0 - 9 indicate these statistics. For example, In the column labeled Thresh = 0.25 of the fuzzy output table, 66.04% of the words for which there was a response were among the top three choices, etc.

Table 2. Results of word recognition with fuzzy output neural network

	Thresh = 0.0	Thresh = 0.25	Thresh = 0.5	Thresh = 0.75
	response = 57%	response = 53%	response = 39%	response = 27%
Rank	% at rank	% at rank	% at rank	% at rank
0	54.74	55.85	69.19	75.37
1	61.40	62.64	75.14	81.34
2	64.56	66.04	77.30	83.58
10	75.09	76.23	84.32	88.06

Table 3. Results of word recognition with crisp output neural network

	Thresh = 0.0	Thresh = 0.25	Thresh = 0.5	Thresh = 0.75
	response = 57%	response = 51%	response = 34%	response = 27%
Rank	% at rank	% at rank	% at rank	% at rank
0	52.98	57.25	69.01	73.68
1	62.11	67.06	79.53	82.71
2	64.91	69.02	79.53	82.71
10	78.25	81.57	87.72	89.47

There are several interesting points here. The fuzzy output network was usually higher in top choice and percentage of answers above the thresholds. Thus, the network with fuzzy output values got a higher percentage of answers correct at the top rank and answered on more words at each level than the crisp output network. It was expected that the fuzzy network would yield a higher percentage correct at the top choice. It was not expected that the network would answer more often at the higher confidence values. Also note that the crisp network had a consistently higher percentage among the top ten choices.

The percentage differences are not large between the two networks and the test set is too small to be conclusive. The experiment described here supports the use of the character network using fuzzy output values over that using crisp output values if the ultimate application is word recognition, but not if the application is isolated character recognition.

Examples of correctly and incorrectly read words are shown in figures 9 and 10.

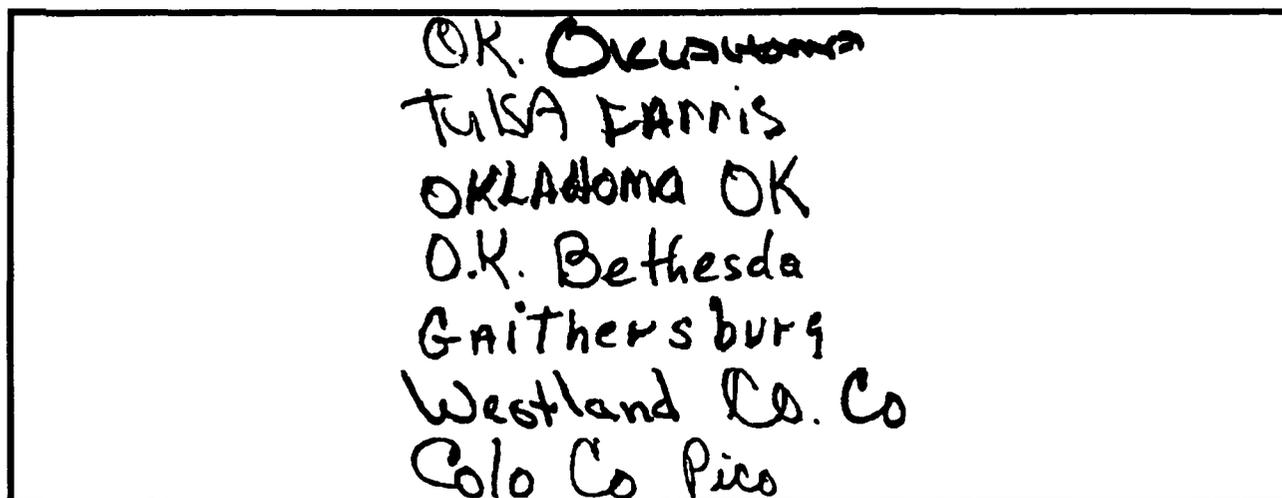


Figure 9. Some words correctly recognized by our system.

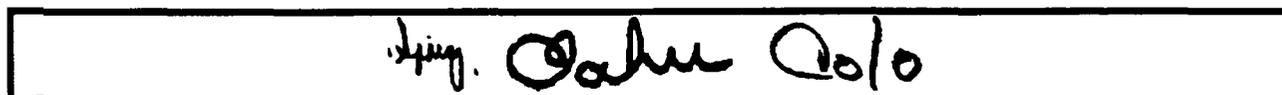


Figure 10. Some words incorrectly recognized by our system.

## CONCLUSIONS

We have described an approach to word recognition that relies heavily on the use of neural networks at the character level. We described experiments involving networks trained with crisp output and with fuzzy outputs. The networks with crisp outputs performed better at the character recognition level. The networks with fuzzy outputs performed better at the word recognition level.

## ACKNOWLEDGEMENTS

This research was supported by the Environmental Research Institute of Michigan (ERIM) and the United States Postal Service Office of Advanced Technology. Several researchers at ERIM had significant inputs to the word described here. We would like to acknowledge Andrew Gillies, Dan Hepp, Michael Whalen, and Margaret Ganzberger for their contributions to this work.

## REFERENCES

- [1] *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, Chateau de Bonas, France, Sept 1991.
- [2] *Proceedings of the International Conference on Document Analysis and Recognition*, Saint-Malo France, Oct. 1991.
- [3] *Proceedings of the United States Postal Service Advanced Technology Conference*, Washington DC, November 1990.
- [4] R. M. Bozinovic and S. N. Srihari, "Off-line Cursive Script Word Recognition", *IEEE Trans PAMI*, Vol. 11, No. 1, Jan 1989.
- [5] T. K. Ho, J. J. Hull, S. N. Srihari, "Word recognition with multi-level contextual knowledge", *Proceedings of the*

*International Conference on Document Analysis and Recognition*, Saint-Malo France, Oct. 1991, pp. 905 - 915.

- [6] T. K. Ho, "A Word Shape Analysis Approach to Recognition of Degraded Word Images", *Proceedings of the United States Postal Service Advanced Technology Conference*, Washington DC, November 1990, pp. 207 - 233.
- [7] A. M. Gillies, "Word Verification for the Contextual Analysis of Address Block Images", *Proceedings of the United States Postal Service Advanced Technology Conference*, Washington DC, November 1990, pp. 247 - 255.
- [8] R. A. Duderstadt, "Isolated Word Recognition for Postal Address Processing", *Proceedings of the United States Postal Service Advanced Technology Conference*, Washington DC, November 1990, pp. 233 - 247.
- [9] Manual for ANSim Version 2.30, Science Applications International Corporation, 1989.
- [10] J. M. Keller, M. R. Gray, J. A. Givens, "A Fuzzy K-Nearest Neighbor Algorithms", *IEEE Trans. Systems, Man, Cybernetics*, Vol SMC-15, No. 4, July -August 1985.
- [11] P. D. Gader and M. P. Whalen, "Advanced Research in Handwritten ZIP Code Recognition, Final Technical Report to USPS Office of Advanced Technology, Dec. 1990.
- [12] M. P. Whalen, M. Ganzberger, P. D. Gader, "Handprinted Word Recognition II: Segmentation and Verification", submitted to *Machine Vision and Applications*.

530-52  
161830

N93-29546

p-10  
**FUZZY NEURAL NETWORK METHODOLOGY  
APPLIED TO MEDICAL DIAGNOSIS**

Marian B. Gorzalczany\* and Mary Deutsch-McLeish

Department of Computing and Information Science, University of Guelph  
Guelph, Ontario, CANADA, N1G 2W1

\*On a leave from the Technical University of Kielce, Department of  
Electrotechnics and Automation, Al. 1000-lecia P. P. 7,  
25-314 Kielce, POLAND

### Abstract

*This paper presents a technique for building expert systems which combines the fuzzy-set approach with artificial neural network structures. This technique can effectively deal with two types of medical knowledge: a nonfuzzy one and a fuzzy one which usually contribute to the process of medical diagnosis. Nonfuzzy numerical data is obtained from medical tests. Fuzzy linguistic rules describing the diagnosis process are provided by human expert. The proposed method has been successfully applied in veterinary medicine as a support system in the diagnosis of canine liver diseases.*

## 1. Introduction

Medical expert systems have relied heavily upon the use of expert opinions and textbook information to form rules or protocols for making decisions, see e.g. [3]. Expert opinions usually take the form of qualitative knowledge and very frequently can be represented as a family of linguistic conditional rules of the type: IF "symptoms" THEN "diagnoses". The "symptoms" often have the form of linguistic statements, like "cholesterol is significantly increased" or "blood pressure is normal". The "diagnoses" can have the form of possibility distributions over some set of diseases, indicating that - for given "symptoms" - disease  $x$  is highly possible while disease  $y$  is less possible and disease  $z$  shouldn't be taken into account. Unfortunately, sometimes a variability between experts in a given domain exists which decreases the quality of the systems obtained.

On the other hand, an increasing number of hospital data base systems have been installed which collect on-line the results of many medical tests. This kind of data represents quantitative medical knowledge. A study of this data collected over time and the incorporation of the knowledge acquired would significantly enhance the quality of medical expert systems.

In a paper [9] considering the current state of medical expert systems, the authors suggest that the time has come to enhance programs which were based on a study of the problem-solving behaviour of clinicians, with knowledge obtained from numerically based methods. They also recognize the difficulty of this approach when they state that "an extensive research effort is required before all these techniques can be incorporated into a single program".

The purpose of this paper is to present a new technique for the construction of medical expert systems. This technique combines neural network structures [7] with some elements

of the theory of fuzzy sets [10]. The proposed technique can effectively cope with two types of medical knowledge: linguistic conditional IF-THEN rules which are expressed by a human expert and numerical data which are collected as a result of medical tests. That is, both mentioned types of medical knowledge can be formalized and incorporated into the expert system. Moreover, both the qualitative and quantitative data can also be processed by the expert system when decision making processes are performed. Some other applications of neural networks to the design of expert systems for medical diagnosis can be found in [1, 5].

The structure combining neural networks and fuzzy sets is called a fuzzy neural network. First, its learning phase is presented during which the network builds a formal representation for the available medical knowledge from a given domain. Then, the inference phase of the network is described. In this phase the network functions as a decision making system. In turn, the application of the proposed methodology in a veterinary medical field, where it has been used as a support system in the diagnosis of the canine liver diseases, is presented.

## 2. Fuzzy neuro-computational scheme for medical knowledge representation

The general procedure for the construction of an expert system which is based on fuzzy neural networks has the following stages:

- a) the choice of the expert system structure in terms of its inputs and outputs and the definition of the primary fuzzy sets for inputs,
- b) the derivation of the linguistic conditional rules representing a human expert's knowledge in a given medical domain as well as collecting available numerical medical data supporting the diagnosis process,
- c) the development of a fuzzy neural-network-based scheme which - during the process of learning - builds a formal representation for the available qualitative and quantitative medical knowledge,
- d) the assessment of the expert system quality against learning data and, if available, test data.

In a general case, the expert system has  $n$  inputs  $x_1 \in X_1, x_2 \in X_2, \dots, x_n \in X_n$  and one output  $y$ . Each input  $x_i$  represents one medical parameter which takes values from the set  $X_i$ . Output set  $Y = \{y_1, y_2, \dots, y_m\}$  is a set of potential diseases. The collections of the primary fuzzy sets represent the aggregations for the masses of numerical data from inputs. These aggregations or clusters are verbally described by means of linguistic labels and form the level at which learning and inference processes are then carried out. Primary fuzzy sets also establish the perception level for the classical neural network which is a part of the proposed fuzzy neural network. The collections of the primary fuzzy sets can be defined in a twofold way. If the qualitative medical knowledge (usually given by the domain human-expert) prevails in the overall description of the system then the primary fuzzy sets can also be defined by the human expert. For instance, many medical parameters can often be characterized by three basic verbal labels: "normal", "decreased", and "increased". These labels, in a natural way, can be formally represented by three fuzzy sets whose membership functions can be readily sketched by a human expert. In turn, these sets can be used as a collection of primary fuzzy sets. If three verbal labels (three fuzzy sets) do not create a sufficiently adequate representation of a given medical parameter, then one has to introduce a respectively higher number of them. On the other hand, if the quantitative medical data prevails in the system description then the primary fuzzy sets can either be defined by a human expert or generated by a formal algorithm of fuzzy clustering [2]. We assume that

for each input  $x_i$  a collection  $A_{i1}, A_{i2}, \dots, A_{in_i} \in F(X_i)$  of  $n_i$  primary fuzzy sets is defined.  $F(X_i)$  denotes a family of all fuzzy sets defined on  $X_i$ .

The second stage of the construction of a fuzzy neural-network-based expert system consists in the derivation and formal representation of available qualitative and quantitative medical knowledge in a given domain. The qualitative knowledge is usually a set of  $K$  linguistic rules representing a human expert's knowledge. The rules have the form:

$$\begin{aligned} & \dots \text{ALSO} \\ & \text{IF } x_1 \text{ is } A'_{1k} \text{ AND } x_2 \text{ is } A'_{2k} \text{ AND } \dots \text{ AND } x_n \text{ is } A'_{nk} \text{ THEN } B'_k \quad (1) \\ & \text{ALSO} \dots \end{aligned}$$

$$k = 1, 2, \dots, K,$$

where  $A'_{ik}$  are the linguistic labels such as "increased", "normal", etc. and  $B'_k$  is a corresponding possibility distribution defined over the set  $Y$  of potential diseases. Linguistic labels  $A'_{ik}$  are formally represented by fuzzy sets which - for simplicity - are also called  $A'_{ik}$ ;  $A'_{ik} \in F(X_i), i = 1, 2, \dots, n$ . Analogously, the possibility distribution  $B'_k$  is represented by fuzzy set  $B'_k \in F(Y)$ . The possibility distribution assigns to each disease  $y_j$  from the set  $Y$ , a number from the interval  $[0, 1]$ , indicating how possible is occurrence of a disease  $y_j$  given the "symptoms" represented by input data in (1). Number 0 assigned to disease  $y_j$  means that  $y_j$ , according to an expert, can not occur. Number 1 - means that  $y_j$  certainly occurs. Regarding the earlier discussion of the primary fuzzy sets, one can notice that the input fuzzy sets  $A'_{ik}$  can also be used as the primary fuzzy sets.

The available quantitative medical data can also be presented in a rule-like form (we have  $L$  rules):

$$\begin{aligned} & \dots \text{ALSO} \\ & \text{IF } x_1 \text{ is } x_{1l} \text{ AND } x_2 \text{ is } x_{2l} \text{ AND } \dots \text{ AND } x_n \text{ is } x_{nl} \text{ THEN } B_l \quad (2) \\ & \text{ALSO} \dots \end{aligned}$$

$$l = 1, 2, \dots, L,$$

where  $x_{il}$  is a numerical value of medical parameter  $x_i$  and  $B_l$  is a corresponding possibility distribution as in (1). In order to unify the formal representation of rules (2) and (1), numerical values  $x_{il}$  of (2) are described by degenerate fuzzy sets  $\tilde{x}_{il}$  called fuzzy singletons whose membership functions are of the form:

$$\mu_{\tilde{x}_{il}} = \begin{cases} 1, & \text{for } x_i = x_{il}, \\ 0, & \text{for } x_i \neq x_{il}. \end{cases} \quad (3)$$

It is also possible that certain rules may contain both qualitative and quantitative data, that is some inputs of a given rule are described by linguistic terms (represented by fuzzy sets) and the other inputs are described by numbers (represented by fuzzy singletons) taken from medical tests.

The third stage of the proposed methodology for the expert system construction consists in the development of a fuzzy neural network which - through the learning process - builds an internal formal representation for both qualitative (linguistic, fuzzy) and quantitative (numerical) medical knowledge described by (1) and (2). Fig. 1 presents a structure of the proposed fuzzy neural network in the learning phase. Symbols  $A'_i, i=1,2,\dots,n$  denote fuzzy sets  $A'_{ik}$  from (1) or fuzzy singletons  $\tilde{x}_{il}$  for (2). Analogously,  $B'$  denotes a corresponding fuzzy set  $B'_k$  from (1) or fuzzy set  $B_l$  from (2). Since the collections of primary fuzzy sets establish the perception level for the classical neural network of Fig. 1, it means that both

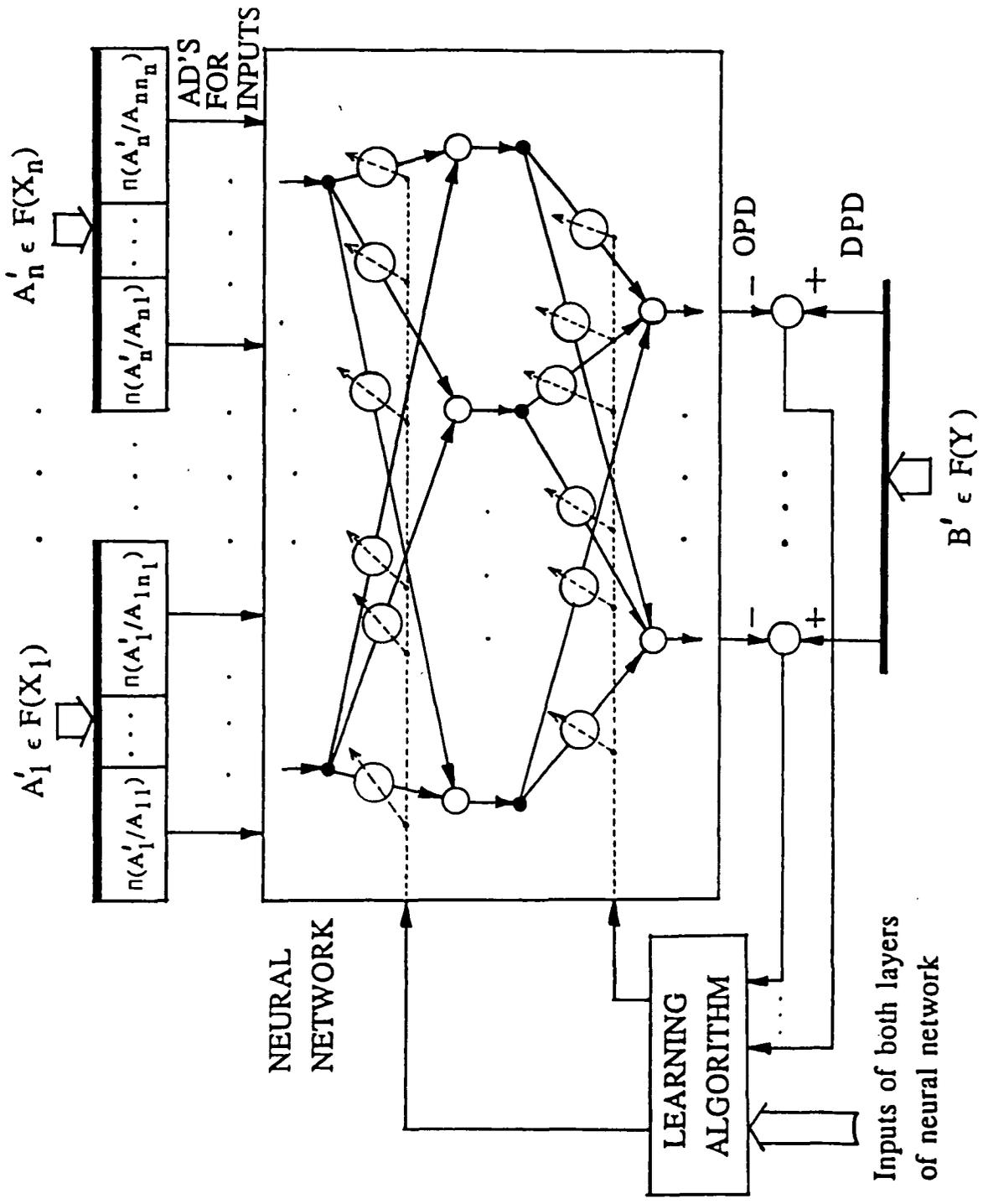


Figure 1: Structure of the fuzzy neural network in the learning phase

nonfuzzy and fuzzy data which are to be processed by the fuzzy neural network, first must be "transferred" to that perception level. The representations of the input transferred data are called activation degrees of primary fuzzy sets for particular inputs (AD's for inputs, for short - see Fig. 1). The AD's are calculated using the notion of a possibility measure [11] that is for input  $x_i$  the AD of a given primary fuzzy set  $A_{ij}$  induced by input fuzzy set  $A'_i$  is expressed by:

$$\Pi(A'_i/A_{ij}) = \sup_{x_i \in X_i} \{ \min[\mu_{A'_i}(x_i), \mu_{A_{ij}}(x_i)] \} \quad (4)$$

In a special case of nonfuzzy numerical data  $x_i^0 \in X_i$ , fuzzy set  $A'_i$  is reduced to a fuzzy singleton  $\tilde{x}_i^0$  and then the expression (4) has the following form:

$$\Pi(\tilde{x}_i^0/A_{ij}) = \mu_{A_{ij}}(x_i^0). \quad (5)$$

The AD's for inputs are then processed by the classical neural network (see Fig. 1), which generates at its outputs an output possibility distribution (OPD - for short). OPD's are in turn compared with corresponding desired possibility distributions (DPD's - for short) coming from the rules (1) and (2). The differences between the DPD's and OPD's are then processed by the learning algorithm which adjusts the neural network weights in such a way as to minimize these differences. As for the classical neural network of Fig. 1, we use a two-layer perceptron [7] because of its universal properties [7,6]. The new back-propagation learning algorithm [8] will be used as a training technique for this network. The overall cost function which is being minimized during the learning process has the form:

$$Q_1 = \frac{1}{mP} \sum_{p=1}^P \sum_{j=1}^m (v_j^p - d_j^p)^2, \quad (6)$$

where:  $v_j^p$  are the OPD's ( $j = 1, 2, \dots, m$ ) generated by the neural network for the  $p$ -th sample of training data; there are K samples of training data of the type (1) and L samples of training data of the type (2), thus  $P=K+L$ ,

$d_j^p$  are the corresponding DPD's coming from the  $p$ -th sample of training data.

The network is trained by initially selecting small random weights and then presenting all available training data repeatedly until the weights converge and the quality index is reduced to an acceptable value - see [7,8] for the details.

### 3. Inference scheme

After the learning process is performed and the optimal values for the weights are stored, modifying slightly the scheme of Fig. 1, the structure of a fuzzy neuro-computational inference engine can be obtained. It is presented in Fig. 2. Symbols  $A_i^0$ ,  $i = 1, 2, \dots, n$  represent the input data ("symptoms") describing the condition of a new patient. If this data results from laboratory tests, it has a numerical form and is represented by a fuzzy singleton - cf(3). Input data may also result from the assessment made by the physician. In this case, very often they have the form of linguistic terms which are represented by fuzzy sets.

The structure of Fig.2 processes the input data and generates the corresponding possibility distribution PD over the set Y of diseases. PD indicates, given the input data, the possibility of occurrence of each disease from the set Y.

The assessment of the expert system quality remains yet to be done. Initially, this assessment should be done with regard to the training data. The cost function  $Q_1$  represented by (6) is also the quality index describing the accuracy of the mapping of the training data by the formal fuzzy-neural-network-based system. The other quality index is the averaged error between the possibility distributions generated by the system and the desired possibility distributions taken from the training data:

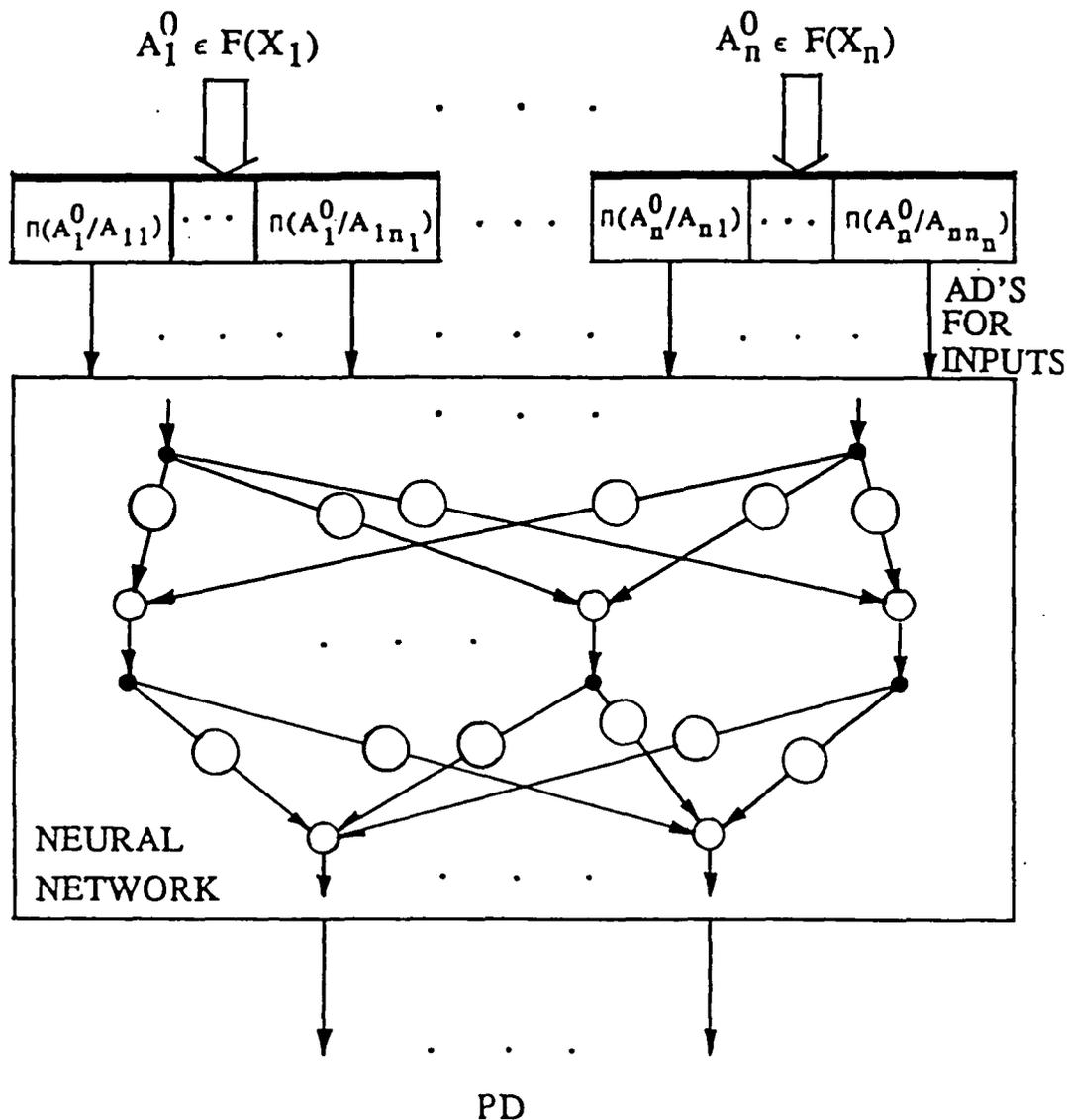


Figure 2: Structure of the fuzzy neural network in the inference phase

$$Q_2 = \frac{1}{mP} \sum_{p=1}^P \sum_{j=1}^m |v_j^p - d_j^p|, \quad (7)$$

and the variance corresponding to  $Q_2$ .  $v_j^p$  and  $d_j^p$  in (7) are the same as in (6). The expert system quality can also be assessed against test data if it is available. In this case, an analogous index to  $Q_2$  and the variance corresponding to it can be used.

#### 4. Application to veterinary medical diagnosis

Now the entire methodology leading to the development of the expert system based on a fuzzy neural network will be illustrated for a domain from the veterinary medical field, that is the diagnosis of liver diseases in small animals and in particular canine liver diseases, cf[4]. Clinicians can accurately diagnose whether or not liver disease is present in about 75 %

of all cases. They can only predict in about 15 % cases the specific type of liver disease. The diagnostic process involves physical examination and laboratory tests; often either a biopsy or a necropsy is performed. The cost of doing laboratory tests is about 20 times cheaper than that of performing a biopsy or necropsy. The latter ones provide more valuable information but, on the other hand, there are some risks in performing biopsy [4]. The aim of this paper is to build an expert system which uses mainly laboratory data and, in a limited range, also verbal rules formulated by a human expert, to determine specific types of liver disease. The expert system will produce a possibility distribution over the set of liver diseases indicating the possibility of occurrence of each of those diseases for a given set of input data.

According to the general procedure for the construction of such an expert system, first, we have to determine its structure in terms of inputs and outputs. Overall, there are 40 medical (biochemical and hematologic) parameters used in the liver disease diagnosis. After a detailed analysis of the correlations between particular parameters finally a subset of 15 biochemical and hematologic parameters has been chosen - see [4] for details. These parameters are listed in Appendix A; they are used as the inputs of our system. The output of the system is a set of 14 liver diseases; they are listed in Appendix B. For each input a set of 3 primary fuzzy sets has been defined using a fuzzy clustering technique [2]. As for the classical neural network of Fig. 1, a two-layer perceptron has been used. It has  $15 \times 3 = 45$  inputs and 14 outputs. After some experimentation, 30 nodes in a hidden layer have been set. As a result of the training process, cost function  $Q_1$  (6) - after 1500 iterations - has been reduced to 0.0004 - see Fig. 3. Switching to the inference phase - see Fig. 2 - the assessment of the expert system against training data resulted in the averaged error  $Q_2$  equal to 0.0088 and the corresponding variance equal to 0.0003. For the training data, the system never produces a response which is essentially contradictory to the desired one. An example of the assessment of the system against training data is presented in Fig. 4. There were also available two sets of test data, not used during the training process. For them we obtained, respectively,  $Q_2$  equal to 0.0357, variance equal to 0.0100, and  $Q_2$  equal to 0.0506, variance equal to 0.0134. They show a high level of correctness of the expert system responses.

Fig. 5 shows an exemplary response of the expert system in the inference phase.

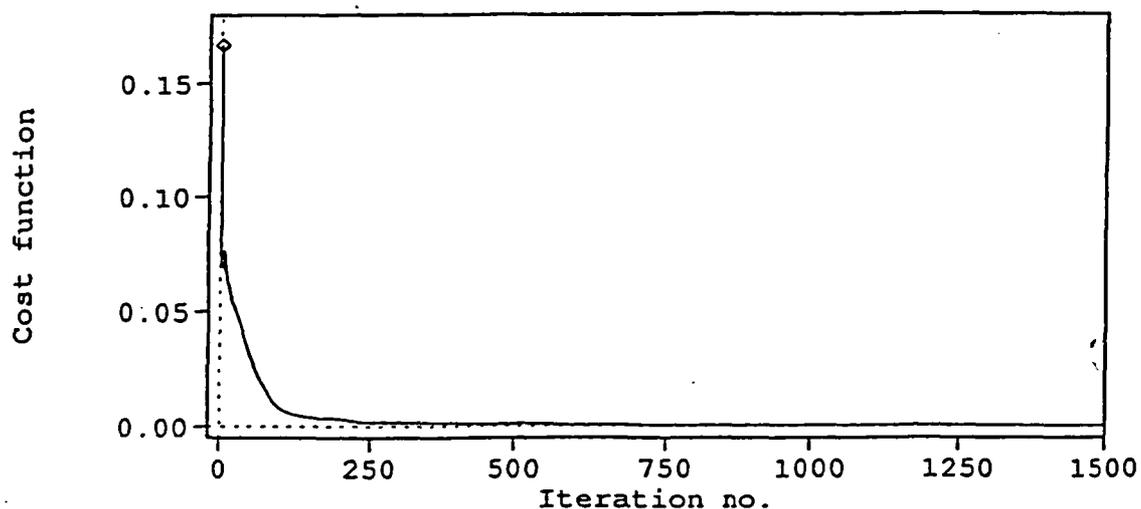


Figure 3: Cost function  $Q_1$  versus number of iterations plot

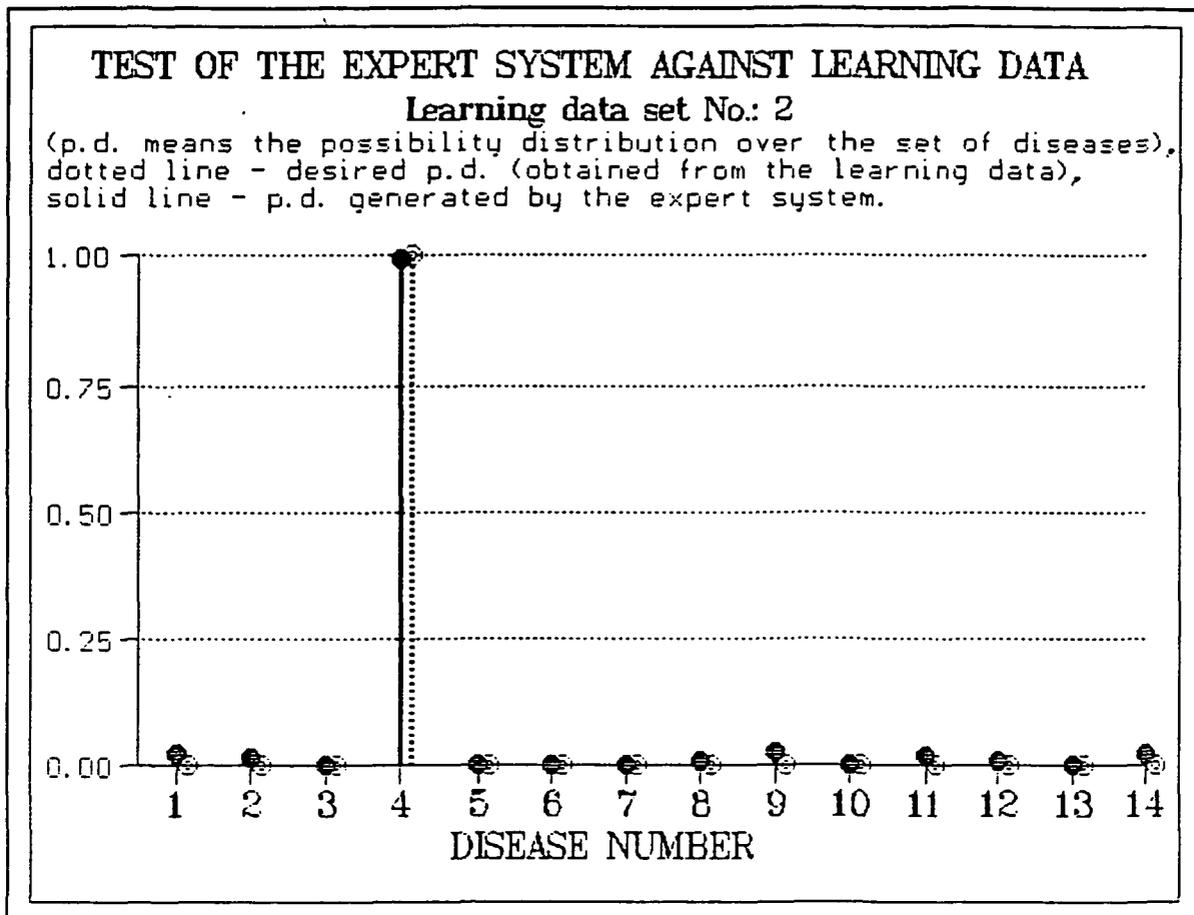


Figure 4: Assessment of the expert system against learning data - an example

## 5. Conclusions

In this paper we have introduced a method for building expert systems which can effectively deal with two main types of medical knowledge: a) a nonfuzzy one (numerical data from medical tests) and b) a fuzzy one (linguistic rules provided by a human expert). The proposed technique combines the fuzzy-set approach with neural network structures, which are characterized by high learning and adaptive capabilities. The proposed method has been successfully applied in the veterinary medical field as a support system in the diagnosis of canine liver diseases.

## References

1. W.G. Baxt, "Use of an artificial neural network for data analysis in clinical decision-making: the diagnosis of coronary occlusion". *Neural Computation*, 2, pp. 480-489, 1990.
2. J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.
3. B. Buchanan and E. Shortcliffe, "Rule-Based Expert Systems: The MYCIN Experiment of the Stanford Heuristic Programming Project", Addison-Wesley, 1986.

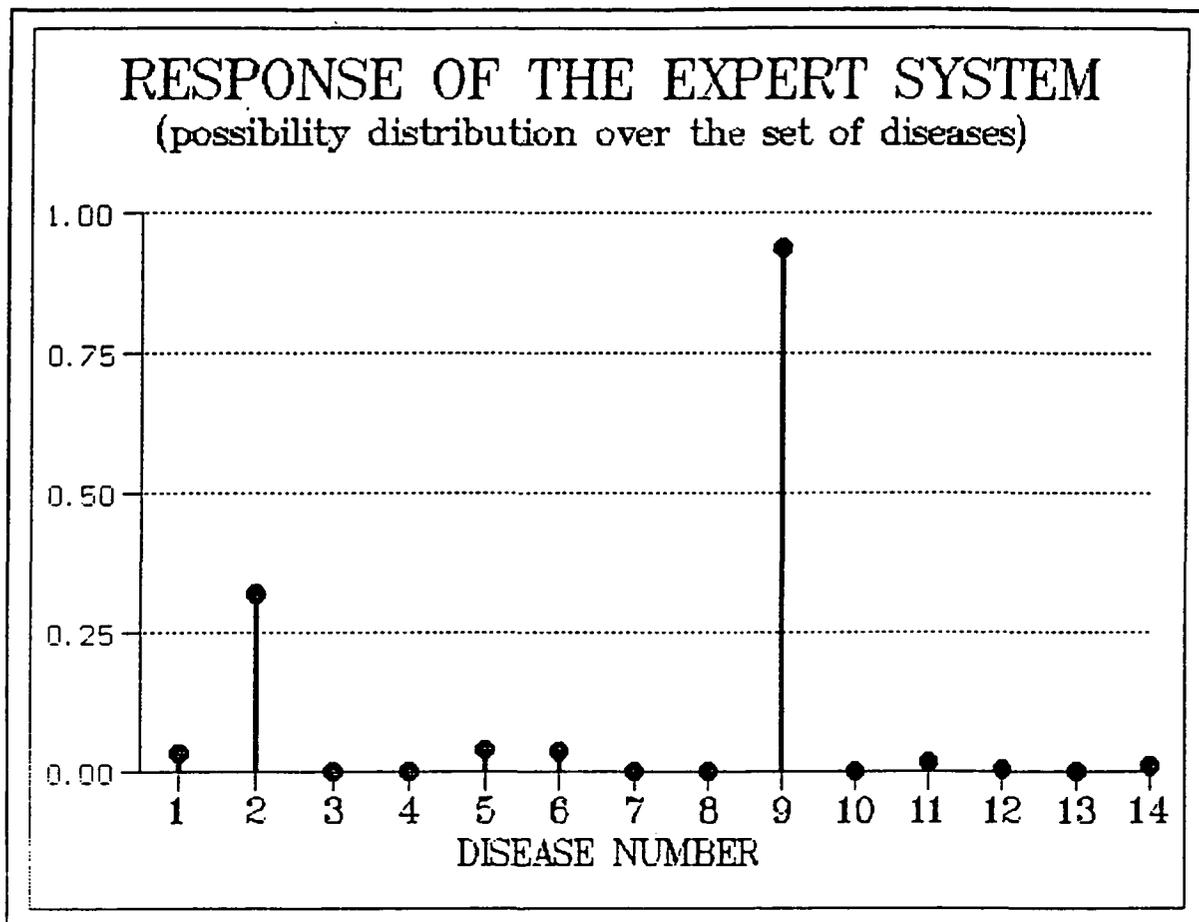


Figure 5: Exemplary response of the expert system

4. M. Deutsch-McLeish, P. Yao and T. Stirtzinger, "A study on the use of belief functions for medical expert systems", *Journal of Applied Statistics*, vol. 18, no. 1, pp. 155-174, 1991.
5. S.I. Gallant, "Connectionist Expert Systems", *Comm. of the ACM*, 31(2), pp. 152-169, 1988.
6. K.Hornik, M. Stinchcombe, H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.
7. R.P. Lippmann, "An introduction to computing with neural nets", *IEEE ASSP Magazine*, pp. 4-22. April 1987.
8. D.E. Rumelhart et al., "Learning internal representations by error propagation", in: "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", Vol. 1: "Foundations", D.E. Rumelhart and J.L. McClelland (Eds), MIT Press, Cambridge, MA, 1986.
9. W. Schwartz, R. Patil and P. Szolovits, "Sounding board, artificial intelligence in medicine", *New England Journal of Medicine*, vol. 16, no. 11, pp. 685-688, 1987.

10. L.A. Zadeh, "Fuzzy sets", Information and Control, vol. 8, pp. 338-353, 1965.
11. L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility". Fuzzy Sets and Systems, vol. 1, pp. 3-28, 1978.

**Appendix A.** Medical parameters which are the inputs of the expert system

Biochemical parameters		Hematologic parameters	
AALB	Albumin	HHCT	Hematocrit
AALKP	Alkaline phosphatase	HIMMAT	Immature leucocytes
AALT	Alanine aminotransferase	HLYMPH	Lymphocytes
AAST	Aspartate aminotransferase	HMCV	Mean corpuscular volume
ACBILI	Conjugated bilirubin	HRETIC	Reticulocytes
AGLUC	Glucose	HRBC	Red blood cells
ATPROT	Total protein	HSEGS	Neutrophil segmentations
AUREA	UREA		

**Appendix B.** Set of liver diseases - the output of the expert system

- |  |                                    |
|--|------------------------------------|
| 1 = Primary and Metastatic Tumors      | 2 = Hepatocellular Necrosis        |
| 3 = Hepatic Congestion                 | 4 = Hepatic Failure                |
| 5 = Hepatomegaly                       | 6 = Hepatic Fibrosis and Cirrhosis |
| 7 = Infectious Hepatocellular Necrosis | 8 = Traumatic Injury Hepatic       |
| 9 = Hepatic Atrophy and Hypoplasia     | 10 = Hepatic Fatty Infiltration    |
| 11 = Steroid Hepatopathy               | 12 = Hepatocellular Dissociation   |
| 13 = Hepatic Encephalopathy            | 14 = Hepatic Torsion               |

531-67  
161831

N93-29547

P-10  
**AN EXPERIMENTAL METHODOLOGY FOR A FUZZY SET PREFERENCE MODEL**

**I. B. TURKSEN, IAN A. WILLSON**  
University of Toronto, Toronto ON Canada M5S 1A4

A flexible fuzzy set preference model first requires appropriate methodologies for implementation. Fuzzy sets must be defined for each individual consumer using computer software, requiring a minimum of time and expertise on the part of the consumer. The amount of information needed in defining sets must also be established. The model itself must adapt fully to the subject's choice of attributes (vague or precise), attribute levels and importance weights. The resulting individual-level model should be fully adapted to each consumer. The methodologies needed to develop this model will be equally useful in a new generation of intelligent systems which interact with ordinary consumers, controlling electronic devices through fuzzy expert systems or making recommendations based on a variety of inputs. The power of personal computers and their acceptance by consumers has yet to be fully utilized to create interactive knowledge systems that fully adapt their function to the user.

Understanding individual consumer preferences is critical to the design of new products and the estimation of demand (market share) for existing products, which in turn is an input to management systems concerned with production and distribution. The question of what to make, for whom to make it and how much to make requires an understanding of the customer's preferences and the trade-offs that exist between alternatives. Conjoint analysis is a widely used methodology which de-composes an overall preference for an object into a combination of preferences for its constituent parts (attributes such as taste and price), which are combined using an appropriate combination function (Green 1984). Preferences are often expressed using linguistic terms which can not be represented in conjoint models. Current models are also not implemented an individual level, making it difficult to reach meaningful conclusions about the cause of an individual's behavior from an aggregate model. The combination of complex aggregate models and vague linguistic preferences has greatly limited the usefulness and predictive validity of existing preference models. A fuzzy set preference model that uses linguistic variables and a fully interactive implementation should be able to simultaneously address these issues and substantially improve the accuracy of demand estimates. The parallel implementation of crisp and fuzzy conjoint models using identical data not only validates the fuzzy set model but also provides an opportunity to assess the impact of fuzzy set definitions and individual attribute choices implemented in the interactive methodology developed in this research. The generalized experimental tools needed for conjoint models can be also be applied to many other types of intelligent systems.

## **FUZZY SETS AND PREFERENCE MODELS**

### **Fuzzy Sets and Linguistic Variables**

The most important consideration in developing a preference model is to select an appropriate representation for preferences. Likert rating scales are the most commonly used measurement scales in conjoint analysis studies (Wittink & Cattin 1989). Since preferences are measured on a labelled rating scale, a representation is needed for linguistic ratings such as "good" and "somewhat good". Fuzzy sets are a good representation for the uncertainty or vagueness inherent in the definition of a linguistic variable (Zadeh 1975), such as a rating of a product (e.g. somewhat good). Since conjoint analysis is based on preferences, a fuzzy set preference model is uniquely suited to this domain. Consumer ratings such as "good" are inherently vague, with a gradient of membership as to which other

possible ratings belong, and a lack of sharp boundaries between ratings. Combinations of preferences, such as "good price AND somewhat good taste", are also expected to be fuzzy, in that classical logic does not adequately describe the combination operator "AND" (Turksen 1986). There has been substantial research in cognitive psychology in general, and categorization in particular, confirming that fuzzy sets are a good representation for linguistic variables. Conceptually, there is agreement on the gradient thesis and the concept of typicality in natural categories and fuzzy set theory (e.g. McCloskey and Glucksberg 1978). For preference models, fuzzy sets can be defined for the linguistic preferences on any labelled rating scale. A Likert scale labelled with 7 linguistic terms (very poor,...,very good) requires a fuzzy set definition for each of the 7 linguistic terms.

### Fuzzy Set Measurement

Fuzzy sets representing preference ratings can be defined either on an individual basis or an aggregate basis. The proposed experimental methodology can obtain and refine fuzzy set parameters (prototype and crossover values) interactively and apply these values to an algorithm for generating individual fuzzy set membership values. Fuzzy sets should ideally be determined on a completely individual basis using interactive software. If this is not practical, membership values may be also pre-defined based on expert assessment or analysis of previous values. Both approaches are useful and are tested in this research. The domain variable for all fuzzy sets is a numeric subjective evaluation on a 0 to 100 scale (100 is best). Seven subjective evaluations (0-100) are anchored to the linguistic terms on the Likert scale and are treated as prototypes. Six additional evaluations are assigned as crossover points and two additional evaluations are added as endpoints, creating a total of 15 domain elements. The rating 75 might be prototypical (i.e. have a membership of 1.0) of the set "good" for example, while 80 might represent the crossover value or point where an evaluation becomes "very good" instead of "good". The transition or crossover domain value is given a membership of .50, corresponding to a point of maximum entropy in a set. For each set there are 3 membership values directly derived from subject responses (prototype and adjacent crossovers). The remaining membership values for the 15 domain elements are assigned to each set based on the slope of the line segment connecting the prototype and crossover elements and their position on the 0-100 domain variable.

In order to study the effect of using a fuzzy set representation for subject ratings, four types of fuzzy sets are used. The crisp number for a rating is essentially a one element set with the prototype having membership of 1.0. A 7 element set is defined by omitting the crossover membership values between prototypes. The 15 element set just described is the main form of measurement used in this research, while a 29 element set is also created by assigning intermediate membership values between existing elements. This provides 1, 7, 15 and 29 element sets on which to assess the impact of the fuzzy set representation and to determine an appropriate set size. An example of a pre-defined set for each of these four sizes is shown in Figure 1, along with an example of an elicited set for a particular subject. The support for the 15 element set "good" in Figure 1 is: { (32,0.05), (40,0.10), (45,0.20), (50,0.30), (55,0.45), (60,0.60), (68,0.80), (75,1.0), (83,0.80), (90,0.60), (100,0.45) }. There are no assumptions about a functional form, a distribution or an axiom such as additivity in the definition, nor of any measurement properties beyond ordinality in the underlying subjective evaluations.

### Fuzzy Set Preference Models

It is not enough to have a good representation for preferences, since a model must estimate an overall preference for an object based on the preferences of its constituent attributes. The combination of linguistic preferences and the parameters associated with this function are a key component of any preference model. With fuzzy sets as attribute evaluations, a valid method of combining fuzzy sets must be found to produce an overall fuzzy set preference. The most effective and simplest crisp

combination function, the vector model, will be used. The vector model uses an importance weighted sum of attribute evaluations, where each estimated attribute importance weight is multiplied by the attribute evaluation and summed across attributes to calculate an overall evaluation. The same approach can be used with fuzzy sets, given measurement support for the operations involved.

The fuzzy set conjoint model represents linguistic ratings in the weighted sum structure of the vector preference model. The fuzzy conjoint model uses the same consumer ratings as conventional conjoint models to allow direct comparisons of the effect on predictive validity. Subject ratings are represented by the fuzzy set definition for the linguistic term, instead of the number associated with the rating ("good" instead of 6). These fuzzy sets are combined in a linear preference model using crisp attribute importance weights similar to the combination of crisp numbers in the vector model. The inputs to the fuzzy conjoint model are the fuzzy sets defined for the linguistic terms of each attribute rating ( $e_i(m)$ ). The membership of each domain element  $y_j$  in the calculated overall preference set ( $\mu_B(y_j, m)$ ) for product  $m$  is defined as

$$(1) \quad \mu_{B'}(y_j, m) = \sum_{i=1}^T \frac{w_i}{\sum_{k=1}^T w_k} \times \mu_{A_i}(x_j, m)$$

where  $\mu_{A_i}(x_j, m)$  is the membership degree of the subject's linguistic rating  $A_i$  for the  $i$ th attribute of product  $m$  for domain value  $x_j$ ,  $w_i$  is a crisp attribute importance weight (1-7), and  $T$  is the number of attributes. For example, the membership of the domain element "good" in the overall set is the weighted sum of the membership of the domain element "good" in each of the attribute evaluation sets. The attribute and overall evaluation domain variables  $x$  and  $y$  are both subjective evaluation scores from 0 to 100 anchored to the identical 7 linguistic terms and their intermediate crossover points. The weight  $w_i$  is a directly elicited subject rating of the attribute's importance from 1 to 7. Attribute importance weights are normalized to produce an overall fuzzy membership value between 0 and 1. The overall preference is a convex linear combination of fuzzy sets representing attribute evaluations.

The fuzzy conjoint model requires only ordinal measurement of the fuzzy sets representing attribute and overall evaluations. The fuzzy conjoint model and a general class of fuzzy set models (including approximate reasoning using min/max norms) have been proven to preserve monotonic weak ordering of inputs through fuzzy operations (Turksen 1991). The membership function must only establish a weak order relation, that of being connected, transitive and bounded. Given such a structure, Turksen has proven that there exists an ordinal scale for the convex linear combination of fuzzy sets. The fuzzy conjoint model requires only ordinal attribute evaluations, which are easily obtained in conjoint analysis using a rating scale. Since an accurate ordering of overall evaluations is sufficient for choice prediction, the minimal measurement requirements of the fuzzy conjoint model are suitable for preference data.

### Experimental Methodology

The individual fuzzy preference model requires an effective implementation, based on a sophisticated interactive computer program. For a product preference study for a given product category, the methodology must provide a wide range of attributes and attribute types from which the subject can pick, and then adjust the values and presentation of these attributes according to subject preferences. The individual options offered by the software are described in Table 1. Attributes and values are routinely pre-assigned in existing conjoint models, making it possible that the attribute is not important to the subject or that the values chosen are not meaningful or distinct (e.g. all 3 pre-specified levels

appear low or high), invalidating the data from that subject and distorting the results in aggregate models. The interactive adjustment to the subject, independent of the model, is critical to successfully understanding preferences and provides important advantages. Each subject is treated as completely independent of all others, yet can be combined, as needed, in a bottom-up rather than top-down process. For example, the effect of price on preferences can be examined for all subjects for which price is an important attribute. Since only a few attributes are actually important to each subject, model complexity can also be reduced from the set of all possible attributes to only a few attributes for each subject. The combination of a good representation for ratings, a model which can use such a representation and an appropriate individual-level implementation can realize the full potential of preference models.

## **THE TEST OF THE FUZZY PREFERENCE MODEL**

### **Experimental Design And Objectives**

The objective of this research is to test a suitable implementation for the fuzzy conjoint model, considering the effect on predictive validity relative to the crisp model. The predictive validity is measured in a cross-validation test for first choice prediction and for longer ordered sequences of preferences from among test stimuli. The experimental design should permit both the fuzzy and crisp conjoint models to be properly implemented at an individual-level, with sufficient estimation and holdout stimuli for a strong test of predictive validity. The experimental procedures should take advantage of computer software to fully adjust the attributes and values to be most appropriate for each subject. The results should also provide convincing evidence about potential applications of fuzzy set preference models and of the interactive methodology outside of conjoint analysis.

The experimental design allows the fuzzy and crisp models to be implemented from identical data, subject ratings of hypothetical products based on combinations of general attribute levels. The attributes and exact value of attribute levels are obtained from each subject by the computer software that administers the experiment. A key component of the experimental design is the stimulus design, which specifies a small number of combinations of general attribute levels (e.g. low/medium/high) designed to permit statistical estimation of crisp conjoint model parameters. Since the fuzzy conjoint model does not require estimation or other statistical techniques such as regression analysis, estimation stimuli are only necessary for the crisp conjoint model. Individual-level models require that all estimation and test (or holdout) stimuli be presented to each subject. A set of 9 stimuli, each having 4 attributes with 3 levels can provide a statistical estimate of the main effects of the 4 attributes on the dependent variable (Addelman 1962). An addition 9 estimation stimuli represent additional combinations and provide sufficient degrees of freedom for crisp estimation procedures. In addition to 18 estimation stimuli, 6 unique holdout or cross-validation stimuli are used which are realistic tests for the models, forcing subjects to trade-off values of attributes (no dominated alternatives).

Two product categories are tested, with each subject completing the conjoint experiment for the delivered pizza and compact car categories. For the pizza category, subjects are to pick a large, 4 item pizza to order for home delivery. For the car category, subjects are instructed to rate cars they are given information on in order to select a few to test drive. Almost any type of product or service could be tested, as long as it could be described to subjects using a computer screen in words or pictures. Some product attributes in any category are naturally vague and linguistic. For a pizza, taste, quality and consistency are described with linguistic terms. For the car category, linguistic attributes such as acceleration (adequate, moderate, strong) and interior space (limited, somewhat roomy, roomy) are provided. Subjects select their four most preferred attributes from a pre-tested set of equal numbers of linguistic and numeric attributes. In addition to subject-specific attribute selection, the software further customizes the levels of all numeric attributes to each subject. For example, the price values

used in product displays are directly elicited from each subject. The software combines a general stimulus design specifying attribute level combinations with a subject-specific set of attributes and attribute values to provide a unique and meaningful set of alternatives for each subject.

### Experiment Implementation

The experiment is implemented using custom-written computer software given to subjects on a diskette that can be run on any IBM-compatible personal computer. The software provides all of the information needed, validating responses and recording data and monitoring information on the distribution diskette, which is returned by the subject. The software is run at any time, without the need for supervision or written instructions. The program describes the product category, obtains the attributes and prototypical values for each subject and then presents hypothetical combinations of these values according to the stimulus design. Subjects rate the products based on the information presented, with the software providing help and carefully logging all subject responses and times for subsequent analysis. The measurement scale used for all subject preference ratings is a Likert rating scale labelled with 7 linguistic terms representing 3 positive, 3 negative and 1 neutral evaluation. The linguistic terms for the scale responses are: very poor (1), poor (2), somewhat poor (3), neutral (4), somewhat good (5), good (6) and very good (7). The subject is instructed to pick the linguistic term that best represents his or her evaluation and to respond using the corresponding number (1-7). This labelled scale makes it possible to use either numbers or fuzzy set definitions for the corresponding linguistic terms as inputs.

In order to implement the fuzzy set model, membership values must be defined for the 15 elements of each of the 7 fuzzy sets representing preference ratings. These values can be assigned based on subjective assessment (identical pre-defined sets for all subjects) or defined interactively using computer software, with both methods reported in the results. The automated set elicitation technique has been implemented successfully in previous experiments (Willson 1991) and is based on a modified reverse rating procedure (Turksen 1991). The subject is asked for the prototype values ("What rating best represents good?") of each of the 7 sets on the underlying 0-100 domain variable. The 6 crossover values ("At what value (0-100) does good become very good?") are then elicited between adjacent ratings, for a total of 13 parameters for the set definition algorithm. The interactive software carefully validates all responses, displays a partition of the domain variable and offers opportunities to change and refine values. The set elicitation procedure takes about 5 minutes to complete on average. Each complete product category takes 25 minutes to complete, making it possible to implement two categories with each subject without excessive demands on the subject. Volunteer subjects were recruited from an undergraduate subject pool at the University of Toronto and were given course credit for successful participation in the experiment. This test involved 70 subjects who took 64.05 minutes on average to complete the entire experiment (2 product categories each).

### Model Implementation and Testing

The experimental software provides relatively clean validated data from subjects. The data files are simply copied to a computer for use by the analysis software that implements the preference models. Specially written analysis modules automatically analyze subject data, determining choice predictions for crisp and fuzzy models. Other analyses can be done on the extensive monitoring data to determine how long subjects spent on each component of the experiment and how much assistance was required during the experiment, both important validation issues. The subject data can be automatically analyzed and input into other management systems, providing updates of demand estimates. This experimental and analytical software has been extensively refined based on previous experiments. The crisp vector conjoint model is implemented by estimating 4 attribute weight parameters using the ratings (implicit attribute and explicit overall evaluations) from the 18 estimation profiles. These crisp weights are

estimated using ordinary least squares regression for each subject and product separately. The fuzzy conjoint model uses the estimation data to refine the pre-defined sets for each subject.

Once the models are implemented, a method of comparing predictive validity is needed. The success of a model in predicting the ranking of overall preferences is the most important criteria. The task is to predict the ranking of the 6 holdout profiles based on the attribute evaluations for these profiles. The subject's actual overall evaluation is the dependent variable to predict. Marketing practitioners and researchers have used relatively weak prediction measures (e.g. prediction of top product from a pair), with few tests of prediction among multiple alternatives, and even then only the rate of prediction for the top ranked alternative. Green (1984) reports that the best first choice prediction rate among current conjoint models is 53 percent for 4 holdout products. A stronger prediction measure developed in this research is the number of correct ordered predictions for each subject, ranging from 0 to 6 (6 holdout stimuli). Due to a promotion, advertising or inventory situation, a consumer could easily purchase a second or third choice product, particularly if preferences are relatively close together. Thus it is important to consider more than just the "hit rate" for first choice.

To determine prediction for a subject, the attribute evaluations of each holdout profile are used together with any estimated parameters to predict the subject's overall evaluation. The overall evaluations are then compared to the model's calculated overall evaluation for each of the 6 holdout profiles in the order of ranking until the correspondence in ranking is broken. For the crisp models, the procedure uses the calculated crisp preference scores ( $y(m)$ ) and the subject's overall evaluations. For the fuzzy conjoint model, a fuzzy similarity measure is used to calculate the sum of the Euclidean distance between corresponding elements in the calculated and actual fuzzy sets, without first defuzzifying either set. The formula for the similarity of two sets is

$$(2) \quad SIM(B'(y_j, m), B(y_j, l)) = 1 / [ 1 + \sqrt{\sum_{j=1}^{15} (\mu_{B'}(y_j, m) - \mu_B(y_j, l))^2} ]$$

where  $B(y, l)$  is the fuzzy set for linguistic term  $l$  (subject's actual overall evaluation) and  $B'(y, m)$  is the fuzzy conjoint model output for product  $m$ . The squared difference of the degree of membership of the  $j$ th element of each set is summed for all elements in the two sets. The square root of this sum added to 1 and then divided by 1 defines the similarity measure. The similarity is computed for product  $m$  to each of the 7 possible linguistic terms  $l$ . The similarity score ranges from 0 to 1 and provides only ordinal information, which is sufficient to determine prediction.

To predict a subject's product preference, the most similar set must be the set representing the subject's actual overall evaluation. For the  $n$ th highest overall evaluation, the calculated preference score should be the  $n$ th highest among the six holdout profiles. If the top rated product has an overall evaluation of good, then the calculated fuzzy set from the fuzzy conjoint model must be most similar to good, compared to any of the other fuzzy sets. The prediction measure (0-6) for each subject for both crisp and fuzzy models is then aggregated across subjects in three summary measures along with the mean. The number of subjects with first choice predicted, the sum of the prediction measure and the weighted sum ( $n(n+1)/2$  where  $n$  is the number of correct ordered predictions) are reported in results and averaged as an overall comparison.

## RESULTS

### Prediction Results

The predictive validity of the crisp and fuzzy preference models is measured in terms of first choice

prediction (%), the sum and weighted sum measures, the average of these three and the mean prediction by subject. All of these measures are reported in Table 2 for the crisp conjoint model and for the fuzzy conjoint model using 7, 15 and 29 element pre-defined sets and using individually elicited sets with 15 elements (see Figure 1 for examples). The two product categories (pizza and car) are combined to provide a sample of 139 (1 product was incomplete). A naive model that randomly orders the 6 holdout profiles is also used to put the results in perspective. The prediction results will be discussed in this section for the standard 15 element pre-defined sets (column 3 of Table 1), with the different set sizes and elicited sets discussed in the following two sections.

The results show that the fuzzy conjoint model predicts the first choice of 76 percent of the subjects, compared to 48 percent for the crisp model and 17 percent for the naive model. This rate of first choice prediction from six holdouts is much higher than results reported in the literature. The crisp conjoint model, however, is still very good relative to the naive model and to the best current methods (e.g. Green 1984), which manage at best only similar prediction rates using much more complex models and aggregate estimation. Previous experiments using the fuzzy conjoint model and the pizza product add further support to these results, with an average first choice prediction rate of 78 percent over three previous tests (Willson 1991).

The prediction advantage of the fuzzy conjoint model over the crisp model increases substantially beyond first choice prediction, as reflected in the sum and weighted sum measures. The advantage increases from 58 percent for first choice prediction to 95 percent for the sum and weighted sum measures. The percentage of subjects for which the first  $n$  choices are correctly predicted declines quite slowly in the fuzzy conjoint model compared to the crisp conjoint model. The relative advantage over the naive model is even larger, starting with a 358 percent improvement in first choice prediction and increasing to 5700 percent for the first 5 choices in order. The overall improvement percentage is the average of the fuzzy conjoint model improvement over the crisp model for the first choice, sum of choices and weighted sum measures. The fuzzy conjoint model is 82.6 percent better than the crisp conjoint model overall. Comparing the mean prediction of the fuzzy conjoint model (FC-15) and the crisp conjoint model, the fuzzy conjoint model predicts the first two choices in order on average from the six holdout profiles, an event that would be expected by chance only 1 in 33 times. Comparing the mean predictions, the fuzzy conjoint mean is significantly better at 1.892 than the crisp conjoint mean of 0.971 with probability of error less than .001.

### Fuzzy Set Definitions

The number of elements used to define the pre-defined fuzzy sets is expected to influence the predictive validity of the fuzzy conjoint model, with more elements increasing prediction to a point and then providing little additional improvement. With 7 linguistic terms providing 7 anchored subjective evaluations (prototype of each set with membership 1.0), there are 3 useful fuzzy set sizes to consider; 7, 15 and 29 elements. The 7 element sets have membership values only for prototype elements in the sets, while 15 element sets add crossover membership elements between prototypes and 29 element sets add an additional intermediate element between each of the 15 elements. The prediction for the three set sizes is given in Table 2 in the second, fourth and fifth columns. Predictive validity improves significantly using 15 element sets compared to 7 element sets, but much less so between 15 and 29 element sets. First choice prediction increases 6 percent from 7 to 15 elements and not at all using 29 element sets. The improvements are larger for longer sequences of prediction, as indicated by the 11 and 19 percent improvements in the sum and weighted sum measures respectively from 7 to 15 elements. The overall advantage over the crisp model increases from 63 percent with 7 element sets to 83 percent with 15 element sets and to 85 percent with 29 element sets.

The mean prediction measure is used to test for significant differences in the results (t-value and its significance given in Table 2). All three set sizes of the fuzzy conjoint model are better than the

crisp conjoint model at a high level of significance (.001). The different set sizes can be compared to each other to see if the additional elements improve prediction significantly. The mean prediction using 15 and 29 element pre-defined sets is higher than the 7 element results, but at a lower level of significance (.05). The 29 and 15 element results are not statistically different. The results suggest that it is important to have an adequate number of set elements in defining membership in sets representing preference ratings. Clearly 7 element sets are not yet sufficient to represent ratings, with only one element covering each rating. Simply representing the crossover elements between adjacent sets using a total of 15 elements is sufficient to achieve very high levels of predictive validity, with little improvement gained by adding an additional 14 elements. This result is a strong confirmation of the notion of minimally sufficient measurement in defining fuzzy sets.

### Experimental Implementation

This section examines the results of the fuzzy conjoint model using elicited set definitions determined interactively. The predictive validity of the fuzzy conjoint model using the elicited set definitions is shown in the third column of Table 2 (FC-EL). The results are somewhat better than the 7 element pre-defined sets and somewhat worse than the 15 element sets. Compared to the crisp results, the elicited results are 65 percent better overall, with a first choice prediction rate of 74 percent, a 54 percent improvement. The mean prediction of 1.712 is 76 percent better than the crisp mean, a difference significant at the .001 level. The large improvement over the crisp model and the comparable results to the pre-defined sets are a strong indication of the value of eliciting fuzzy set information from subjects. Using individual-level models it is most desirable to be able to also have individual fuzzy set definitions, and to do so easily and with a minimal loss of predictive validity compared to aggregate methods. This simple set elicitation procedure involving only 13 parameters for all 7 sets and requiring only 5 minutes appears to meet this goal. The subjects in this research are not told about fuzzy sets and do not examine graphed sets or refine membership functions. This is an important criteria for the future use of fuzzy set methods in management and with consumers, who can not be expected to use engineering-oriented set definition software.

One final aspect of the results is the success of the fully interactive experimental method implemented using computer software. Earlier tests of the crisp and fuzzy models using identical 15 element pre-defined sets and the pizza product allow a direct comparison of the effect of using the interactive methodology. The first two tests of the fuzzy conjoint model used written questionnaires with only pre-assigned attributes and attribute values and otherwise identical rating scales and data. The fuzzy conjoint model results improve using the experimental software. The average first choice prediction rate for all written tests is 71.5 percent, compared to 76 percent for the computerized studies (Willson 1991). Extensive analysis of subject comments and responses demonstrates that subjects can easily use the preference software without the need for prior training, providing meaningful responses sufficient to implement both fuzzy and crisp models from their preference ratings. This is a clear demonstration that fuzzy sets and linguistic preferences can be easily obtained from subjects in an automated methodology based on interactive computer software. The result is a fuzzy conjoint study that is easier to implement and requires fewer subjects than existing crisp conjoint methods, providing much better information for management at a lower cost.

## CONCLUSIONS & DISCUSSION

The results clearly demonstrate the improvements due to an appropriate preference representation and an individual-level model implemented with fully adaptive computer software. The prediction improvement over existing models in a realistic comparison test using identical data is 83 percent, with

the largest improvements for the more difficult task of predicting longer sequences of choices. The first choice prediction rate of 76 percent for 6 holdout products is much higher than crisp model, while the crisp model equals the best results in the literature. The results also demonstrate the value of using a fully interactive computer program to implement individual-level models, adjusting attribute choices and values to each subject. Individual fuzzy set definitions can be obtained from any subject in a few minutes with large improvements in predictive validity compared to crisp models. Since the fuzzy preference model does not require statistical estimation, it is much easier to implement. The adjustment of attributes and values to each consumer improves both crisp and fuzzy set models enough for the crisp model to surpass existing aggregate models with only individual estimation. Since all subjects selected at least one linguistic attribute (2 on average) in their top 4 choices, it is important to accommodate both numeric and linguistic information in a preference model. The fuzzy model output provides information on each individual's preferences and how attribute values are traded-off. Optimal product(s) can be customized to particular market segments, created by grouping similar individuals together. The resulting aggregate market segment will be based on meaningful interpretations of the behavior of individual consumers, unlike existing approaches which rely on imaginary aggregates of consumers.

The success of the fuzzy set elicitation procedure (compared to crisp results and pre-defined sets) may have important implications for many types of intelligent systems. Almost any fuzzy expert system that involves individual tastes or perceptions (picture quality, microwave cooking, car performance) can benefit from the individual definition of fuzzy sets used in the inference process. The machine intelligence expected in the next generation of consumer products will require an ability to individually adapt to consumer preferences for attributes and to differences in the definition of linguistic terms. It is hard to imagine an optimal television picture or microwave cooking cycle in the abstract, without regard to a particular individual's preferences. A microwave should interact with an individual to learn what "well-done" means and to learn when this degree of cooking is desired. A television picture controller must consider the particular visual characteristics of its viewers, such as relative colour sensitivity and particular picture preferences (e.g. strong or weak flesh tones). Such intelligent devices can directly utilize the methods demonstrated in this research to improve performance and ultimately to better adapt the characteristics of the product to its user.

## REFERENCES

- Addelman, Sidney (1962), "Orthogonal Main-Effect Plans for Asymmetrical Factorial Experiments," Technometrics, 4 No. 1 (February), 21-46.
- Green, Paul E. (1984), "Hybrid Models for Conjoint Analysis: An Expository Review," Journal of Marketing Research, 21 (May), 155-69.
- McCloskey, Michael and Sam Glucksberg (1978), "Natural Categories: Well defined or fuzzy sets?," Memory and Cognition, 6 (4), 462-72.
- Turksen, I.B. (1986), "Interval Valued Fuzzy Sets Based On Normal Forms," Fuzzy Sets and Systems, 20, 191-210.
- Turksen, I.B. (1991), "Measurement of Membership Functions and their Acquisition," Fuzzy Sets and Systems, 40 (1), 5-38.
- Willson, Ian A. (1991), "Fuzzy Sets and Uncertain Choice Processes: An Empirical Test of Models," Doctoral dissertation, Faculty of Management, University of Toronto.
- Wittink, Dick R. and Philippe Cattin (1989), "Commercial Use of Conjoint Analysis: An Update," Journal of Marketing, 53 (July), 91-96.
- Zadeh, Lofti A. (1975), "The Concept of a Linguistic Variable and its Application to Approximate Reasoning," Parts 1 and 2, Information Sciences, 9, 199-249, 301-357.

**TABLE 1: INDIVIDUAL-LEVEL MODEL OPTIONS**

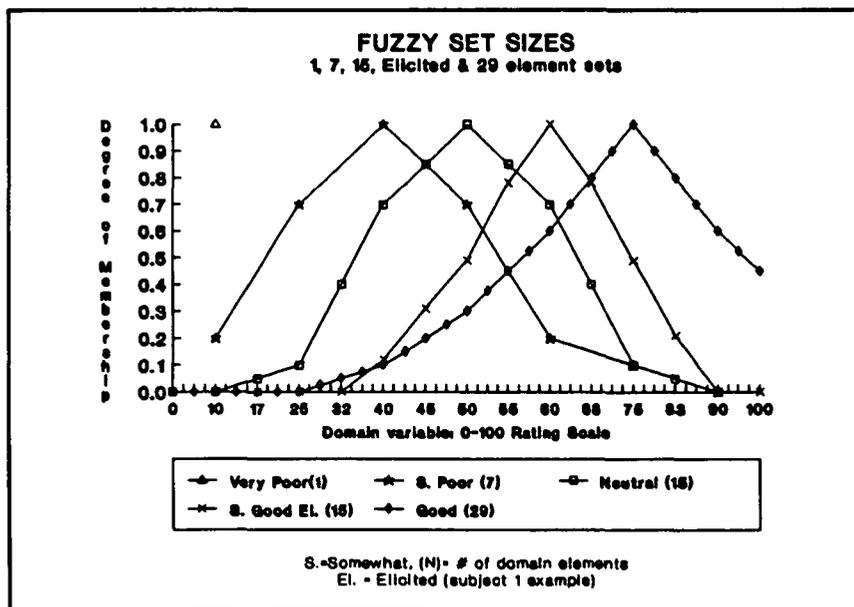
<u>Option</u>	<u>Adjustment to each subject</u>
Attribute Types	Both numeric and linguistic attributes must be available (e.g. linguistic acceleration OR numeric engine horsepower for a car)
Attributes	Only the most important attributes should be used, selected by the subject from a larger list of possible product attributes
Attribute Order	Attributes used in order of importance as ranked by subject
Attribute Values	Attribute values customized for each subject to ensure relevance. (E.g. elicit low/medium/high prices from the subject)
Fuzzy Set Definition	Individually elicit fuzzy set parameters from each subject (prototypes and crossovers) to define their fuzzy sets

**TABLE 2: PREDICTION RESULTS**

<u>Prediction Measure:</u>	<u>Crisp</u>	<u>FC-7</u>	<u>FC-EL</u>	<u>FC-15</u>	<u>FC-29</u>
1st choice rate:	48.2%	71.9%	74.1%	76.3%	76.3%
Sum of choices measure:	135	236	238	263	267
Weighted sum measure:	287	470	471	559	567
Overall % Advantage (fuzzy/crisp):		62.6%	64.7%	82.6%	84.5%
Mean prediction:	0.971	1.698	1.712	1.892	1.921
t-value of mean differences:					
FC-___ - Crisp		0.73 <sup>a</sup>	0.74 <sup>a</sup>	0.92 <sup>a</sup>	0.95 <sup>a</sup>
FC-___ - FC-7			0.01	0.19 <sup>b</sup>	0.22 <sup>b</sup>
FC-29 - FC-15					0.03

<sup>a</sup> p < .001. <sup>b</sup> p < .05. <sup>c</sup> p < .10. FC-7/15/29 = Fuzzy Conjoint with 7/15/29 element pre-defined sets, FC-EL = Fuzzy Conjoint using 15 element elicited sets.

**FIGURE 1: FUZZY SETS FOR "GOOD"**



p-10

**A FUZZY SET PREFERENCE MODEL FOR MARKET SHARE ANALYSIS**

I. B. TURKSEN, IAN A. WILLSON

University of Toronto, Toronto ON Canada M5S 1A4

Consumer preference models are widely used in new product design, marketing management, pricing and market segmentation (Green and Srinivasan 1990, Wittink and Cattin 1989). The success of new products depends on accurate market share prediction and design decisions based on consumer preferences. The vague linguistic nature of consumer preferences and product attributes combined with the substantial differences between individuals creates a formidable challenge to marketing models. The most widely used methodology is conjoint analysis. Conjoint models as currently implemented represent linguistic preferences as ratio or interval-scaled numbers, use only numeric product attributes and require aggregation of individuals for estimation purposes. It is not surprising then that these models are costly to implement, are inflexible and have rather poor predictive validity not substantially better than chance, which in turn affects the accuracy of market share estimates.

A fuzzy set preference model can easily represent linguistic variables either in consumer preferences or product attributes with minimal measurement requirements (ordinal scales), while still estimating overall preferences suitable for market share prediction. This approach results in flexible individual-level conjoint models which can provide more accurate market share estimates from a smaller number of more meaningful consumer ratings. Fuzzy sets can be incorporated within existing preference model structures, such as a linear combination, using the techniques developed for conjoint analysis and market share estimation. The purpose of this article is to develop and fully test a fuzzy set preference model which can represent linguistic variables in individual-level models implemented in parallel with existing conjoint models. The potential improvements in market share prediction and predictive validity can substantially improve management decisions about what to make (product design), for whom to make it (market segmentation) and how much to make (market share prediction).

**A FUZZY SET PREFERENCE MODEL****A General Preference Model**

The underlying theory of conjoint measurement is that an overall preference for a product or service can be de-composed into a combination of preferences for its constituent parts (attributes such as taste and price), which are combined using an appropriate combination function. An example is a weighted sum of  $T$  attribute preferences, where the preference for alternative  $m$  is defined as

$$(1) \quad y(m) = \sum_{i=1}^T (w_i \times e_i(m)) + w_0$$

where a numeric value ( $e_i(m)$ ) is used for the linguistic evaluation of the  $i$ th attribute (e.g. "good" represented by 6 on a 1 to 7 scale). These attribute evaluations are the independent variables that are combined to calculate the dependent variable; an estimated overall preference. Crisp attribute importance weights  $w_i$  are statistically estimated using subject ratings of both attribute and overall evaluations on a separate group of "estimation" products. The weights are then used to predict overall preferences for a second group of test products ("holdouts"), which are compared to the subject's actual ratings to assess predictive validity in a cross-validation test.

Since the attribute evaluations given by subjects are often linguistic terms on a labelled rating scale,

a preference model should represent linguistic preferences accurately. The combination function should be appropriate for linguistic variables, producing an overall linguistic preference. Since the meaning of linguistic terms also varies among subjects, it is particularly important to use individual-level models. The appeal of using fuzzy sets in preference models comes from representing linguistic variables in a mathematical structure that closely corresponds to subject preferences.

### Fuzzy Sets and Linguistic Variables

Fuzzy sets are a good representation for the uncertainty or vagueness inherent in the definition of a linguistic variable (Zadeh 1975). Linguistic variables are prevalent in describing products ("large") and in expressing preferences ("somewhat good"). Since conjoint analysis is based on preferences, a fuzzy set preference model is uniquely suited to this situation. Consumer ratings such as "good" are inherently vague, with a gradient of membership as to which other ratings belong, and a lack of sharp boundaries between ratings. Combinations of ratings, such as "good price AND somewhat good taste", are also expected to be fuzzy, in that classical logic does not adequately describe the combination operator "AND" (Turksen 1986).

Fuzzy sets are defined for each of the 7 linguistic ratings on a Likert scale. The domain variable for these sets is a numeric subjective evaluation on a 0 to 100 scale. Seven subjective evaluations (0-100) are anchored to linguistic terms as prototypes, with 6 additional evaluations assigned as crossover points and 2 additional endpoints, for a total of 15 domain elements in each set. The rating 75 might be prototypical (i.e. have a membership of 1.0) of the set "good" for example. The fuzzy sets for the linguistic terms good and very good are shown in Figure 1. The sets are graphed for subjective ratings above 50, where they have the highest membership. The fuzzy set "good" is shown as: { (50,0.30), (55,0.45), (60,0.60), (68,0.80), (75,1.0), (83,0.80), (90,0.60), (100,0.45) }. There are no assumptions about a functional form, a distribution or an axiom such as additivity in the fuzzy set definition, nor of any measurement properties beyond ordinality in the underlying subjective evaluations.

### The Fuzzy Set Preference Model

A fuzzy set preference model is developed to represent linguistic ratings ( $e_i(m)$ ) in the vector preference model. This new model is, in effect, a "fuzzified" vector conjoint model from Equation 1. Subject ratings are represented by the fuzzy set definition for the linguistic term applicable to each rating (e.g. "good" for 6), instead of the number associated with the rating (1-7). These fuzzy sets are combined in a linear preference model using attribute weights in a manner similar to the combination of "crisp" (non-fuzzy) numbers in the vector model. The inputs to the fuzzy conjoint model are the fuzzy sets defined for each attribute rating ( $e_i(m)$ ). The membership of each domain element  $y_j$  in the calculated overall preference set ( $\mu_{B'}(y_j, m)$ ) for product  $m$  is defined as

$$(2) \quad \mu_{B'}(y_j, m) = \sum_{i=1}^T \frac{w_i}{\sum_{k=1}^T w_k} \times \mu_{A_i}(x_j, m)$$

where  $\mu_{A_i}(x_j, m)$  is the membership degree of the subject's linguistic rating  $A_i$  for the  $i$ th attribute of product  $m$  for domain element  $x_j$ ,  $w_i$  is a crisp attribute importance weight (1-7), and  $T$  is the number of attributes. For example, the membership of the domain element "good" in the overall calculated set  $B'$  is the weighted sum of the membership of the domain element "good" in each of the attribute evaluation sets. The attribute and overall evaluation domain variables  $x$  and  $y$  are both subjective

evaluations from 0 to 100. The crisp weight  $w_i$  is a directly elicited subject rating of the attribute's importance from 1 to 7. Attribute importance weights are normalized to produce an overall fuzzy membership value between 0 and 1. The overall preference is a convex linear combination of fuzzy sets representing attribute evaluations.

The fuzzy conjoint model requires only ordinal measurement of the fuzzy sets representing attribute and overall evaluations. There are no assumptions about interval or ratio scale properties, avoiding the need for extensive diagnostic procedures which are often required by crisp preference models. The fuzzy conjoint model and a general class of fuzzy set models (including approximate reasoning using min/max norms) have been proven to preserve monotonic weak ordering of inputs through fuzzy operations (Turksen 1991). The membership function must only establish a weak order relation, that of being connected, transitive and bounded. Given such a structure, Turksen has proven that there exists an ordinal scale for the convex linear combination of fuzzy sets. The fuzzy conjoint model requires only ordinal attribute evaluations, which are easily obtained in a conjoint scenario using a rating scale. The membership values for each linguistic term  $\mu_{A_i}(x_i, -)$  can be pre-specified or can be based on an elicitation procedure which obtains set parameters from each subject. Since an accurate ordering of overall evaluations is sufficient for choice prediction, the minimal measurement requirements of the fuzzy conjoint model are well suited to preference data.

## THE TEST OF FUZZY AND CRISP CONJOINT MODELS

### Hypotheses

The purpose of this research is to test the fuzzy conjoint model, both in terms of predictive validity and market share estimation. The experimental design should permit both the fuzzy and crisp conjoint models to be properly implemented at an individual-level, with sufficient estimation and holdout stimuli for a strong test of predictive validity. The experimental procedures should take advantage of computer software to fully adjust the attributes and values to be most appropriate for each subject. Two specific hypotheses tested are:

H1: The fuzzy conjoint model will predict the first choice and even more of the ordered sequences of choices of more subjects than the crisp vector conjoint model.

H2: Improvements in predictive validity should be related to the representation of vagueness in product attributes and fuzzy set definitions.

### Experimental Design

The experimental design has two main factors: the product category, which varies within subject, and the vagueness of the attribute information, which varies according to the linguistic or numeric attributes selected by each subject. Vagueness is defined as the number of linguistic attributes selected, ranging from 0 to  $n$ , where  $n$  is the number of attributes in the stimulus design. Since most products have some characteristics that are linguistic, it is important to permit the subject to select this type of attribute. The experimental design allows both fuzzy and crisp conjoint models to be implemented from the same subject ratings of full profile stimuli collected in a computer-assisted conjoint task.

A key component of the experimental design in conjoint analysis is the stimulus design, which specifies a small number of combinations of attribute levels designed to permit effective estimation of crisp conjoint parameters (attribute weights). A total of 24 hypothetical stimuli are used in this

experiment to provide 18 estimation stimuli and 6 cross-validation holdout stimuli on which to test each model at an individual level. Since the fuzzy conjoint model does not require estimation and does not use regression analysis or other statistical techniques, the 18 estimation stimuli are only necessary to estimate the *crisp conjoint model*. The 6 holdout stimuli are designed to be realistic, requiring trade-offs between attributes (no dominated alternatives).

### Product Categories

Two product categories are tested in this research, with each subject completing the conjoint experiment for the delivered pizza and compact car categories. For the pizza category, subjects are to pick a large, 4 item pizza to order for home delivery. For the car category, subjects are instructed to rate cars they are given information on in order to select a few to test drive. Some product attributes in any category are naturally vague and linguistic. For a pizza, taste, quality and consistency are attributes described with linguistic terms. For the car category, linguistic attributes such as acceleration (adequate, moderate, strong) and interior space (limited, somewhat roomy, roomy) are provided. Subjects select their four most preferred attributes from a pre-tested set of equal numbers of linguistic and numeric attributes. In addition to subject-specific attribute selection, the software further customizes the levels of all numeric attributes to each subject. For example, the price values used in product displays are directly elicited from each subject. The software combines a general stimulus design specifying attribute level combinations with a subject-specific set of attributes and attribute values to provide a unique and meaningful set of alternatives for each subject.

### Measurement

The measurement scale used for attribute and overall preferences is a Likert scale labelled with 7 linguistic terms representing 3 positive, 3 negative and 1 neutral evaluation. The linguistic terms are: very poor (1), poor (2), somewhat poor (3), neutral (4), somewhat good (5), good (6) and very good (7). The subject is instructed to pick the linguistic term that best represents his or her evaluation and to respond using the corresponding number (1-7). This labelled scale makes it possible to use either numbers or fuzzy set definitions for the corresponding linguistic terms as inputs to the conjoint models (Equations 1 and 2). For the fuzzy set model, membership values are assigned to the 15 elements of each of the 7 fuzzy sets based on either experimental data and expert assessment (pre-defined) or based on each subject's response to an interactive set definition module in the conjoint analysis software. The later technique has been implemented successfully (Willson 1991), demonstrating that preference models can obtain fuzzy set parameters from ordinary subjects with relative ease. The results in this paper use only pre-defined set definitions in order to assess the success of the model separate from the issue of interactively defining individual sets. The same pre-defined sets have been used in all research involving the fuzzy conjoint model to allow an overall assessment across product categories, subject types and preference model implementations.

### Experimental Procedures

The preference experiment is administered by custom written computer software given to subjects on a diskette (Willson 1991). Subjects can run the software on any IBM-compatible personal computer without any installation. The software provides all of the information needed, validating responses and recording data and monitoring information on the distribution diskette, which is returned by the subject. Volunteer subjects were recruited from marketing classes at the University of Toronto and paid \$10 (CAN) for completing the study. Each product category takes about 25 minutes to complete, with the entire experiment taking 58.9 minutes on average. The software describes the product

category, obtains the attributes and prototypical values for each subject and then presents hypothetical combinations of these values according to the stimulus design. Subjects rate the products based on the information presented, with the software providing help and carefully logging all subject responses and times for subsequent analysis. Analysis software then automatically validates subject data, determining choice predictions and market shares for crisp and fuzzy set models.

### Prediction Tests

Practitioners and researchers have used relatively weak prediction measures in conjoint analysis. Paired comparisons of alternatives, for example, result in prediction rates for the first choice of each pair that are only 40 percent better than chance (Currim and Sarin 1984). Relatively few results are reported for prediction among multiple alternatives, and even then only the rate of prediction for the top ranked alternative. Green (1984) reports that the best first choice prediction rate among current conjoint models is 53 percent for 4 holdout products. A stronger prediction measure to use is the number of correct ordered predictions for each subject, ranging from 0 to 6 in this research. Due to a promotion, advertising or inventory situation, a consumer could easily purchase a second or third choice product, particularly if preferences are relatively close together. Since all models are individual-level, the prediction test is a cross-validation test using each subject's six holdout profiles, with model estimation based only on the ratings given by that subject for the 18 estimation profiles.

To determine prediction for a subject, the attribute evaluations of each holdout profile are used together with any estimated parameters to predict the subject's overall evaluation. The subject's actual overall evaluations are then compared to the model's calculated overall evaluation for each of the six holdout profiles in the order of preference. The comparison continues until the correspondence in ranking between subject ratings and estimated model ratings is broken. For the crisp models, the procedure uses the calculated crisp preference scores ( $y(m)$ ) and the subject's overall evaluations. For the fuzzy conjoint model, a fuzzy similarity measure is used to calculate the sum of the Euclidean distance between corresponding elements in the calculated and actual fuzzy sets, without first defuzzifying either set. The formula for the similarity of two sets is

$$(3) \quad SIM(B'(y_j, m), B(y_j, l)) = 1 / [ 1 + \sqrt{\sum_{j=1}^{15} (\mu_{B'}(y_j, m) - \mu_B(y_j, l))^2} ]$$

where  $B(y_j, l)$  is the fuzzy set for linguistic term  $l$  (subject's actual overall evaluation) and  $B'(y_j, m)$  is the calculated set for product  $m$  from Equation 2. The squared difference of the degree of membership of the  $j$ th element of each set is summed for all elements in the two sets. The square root of the sum added to 1 and then divided by 1 defines the similarity measure. The similarity is computed for product  $m$  to each of the 7 possible linguistic terms  $l$ . The similarity score ranges from 0 to 1 and provide only ordinal information, which is sufficient to determine prediction.

To predict a subject's product preference, the most similar set must be the set representing the subject's actual overall evaluation. For the  $n$ th highest overall evaluation, the calculated preference score should be the  $n$ th highest among the six holdout profiles. If the top rated product has an overall evaluation of good, then the calculated fuzzy set from the fuzzy conjoint model must be most similar to good, compared to any of the other fuzzy sets. The prediction measure (0-6) for each subject for both crisp and fuzzy models is then aggregated across subjects in three summary measures as well as the mean. The number of subjects with first choice predicted, the sum of the prediction measure and the weighted sum ( $n(n+1)/2$  where  $n$  is the number of correct ordered predictions) are reported in results and averaged as an overall comparison.

## Preference Model Implementation

The vector conjoint model (Equation 1) is an appropriate crisp conjoint model for comparison, since it can be "fuzzified" by using fuzzy sets for linguistic ratings and since its 4 linear parameters are easily estimated from the 18 estimation products in the stimulus design. Crisp model attribute weights are estimated using ordinary least squares regression for each subject and product separately. The fuzzy conjoint model does not need estimation in the conventional sense, but to be comparable to the crisp model, the estimation data can be used to refine the pre-defined sets for each subject. A cut-off level or alpha-cut for fuzzy set membership is found by discrete testing of the estimation ratings. The goal is to consider only the more important higher membership elements in the calculated overall fuzzy set. For each subject, the cut-off level (among 10 tested) with the highest prediction for the estimation profiles is then automatically used for the holdout profiles in the prediction test. In previous research this procedure improved the quality of predictions somewhat (longer predictions), but did not affect the number of first choice predictions.

## RESULTS

### Prediction

The predictive validity of the crisp and fuzzy conjoint models is presented in this section, addressing both hypotheses. The prediction measures reported in Table 1 are the percentage of subjects for which first choice is predicted, the sum of choices and weighted sum of choices measures and the mean prediction used for statistical tests. The two product categories (pizza and car) are combined to provide a sample of 50. A naive model that randomly orders the six holdout profiles is given in the first column, with the crisp vector conjoint model shown in the second column and the fuzzy conjoint model in the third column. The results show that the fuzzy conjoint model predicts the first choice of 82 percent of the subjects, compared to 50 percent for the crisp model and 17 percent for the naive model. This rate of first choice prediction from six holdouts is much higher than crisp model results reported in the literature, with the crisp conjoint model results also very good relative to the best current models (e.g. Green 1984). Previous experiments using the fuzzy conjoint model and the pizza product add further support, with an average first choice prediction rate of 78 percent using the experimental software and 72 percent using written questionnaires over a total of 142 subjects (Willson 1991). The improvement of the fuzzy preference model over the crisp preference model increases substantially beyond first choice prediction, as reflected in the sum and weighted sum measures. The advantage increases from 64 percent for first choice prediction to 140 percent for the sum of choices and 212 percent for the weighted sum measures. The relative improvement of the fuzzy conjoint model over the naive model is even larger, ranging from 390 percent for first choice prediction to 11500 percent for the first 5 choices in order (16 percent fuzzy prediction rate versus .138 percent naive prediction rate).

The overall improvement is obtained by averaging the fuzzy conjoint model improvement over the crisp model for the first choice, sum of choices and weighted sum measures. The fuzzy conjoint model is 138 percent better than the crisp conjoint model overall. To conduct a statistical test of differences in predictive validity, the mean of the prediction measure for each subject is calculated as 2.16, .90 and .28 for the fuzzy conjoint model, the crisp conjoint model and the naive model respectively. On average, the fuzzy conjoint model predicts the first two choices in order from the 6 holdout profiles, an occurrence that would be expected by chance only 1 in 33 times. Comparing the mean predictions, the fuzzy conjoint model is significantly better than the crisp conjoint model with probability of error less than .0001.

To examine the second hypothesis, the effects of both the number of set elements used in defining sets and the number of vague linguistic attributes are considered. The number of non-zero elements in a fuzzy set can be varied in stages from a single element "crisp" set (one prototype element with membership 1.0) to the 15 element sets (11-14 non-zero elements) used in the prediction results. The predictive validity of the fuzzy conjoint model is tested using four variations of the same basic fuzzy set definition using 1,3,7 and 15 element sets for each of the 7 linguistic terms. The first three sizes are defined over 7 domain elements (1 element per rating on the scale), while the larger size is defined over 15 domain elements, with intermediate elements between ratings and two additional endpoint elements. The prediction results clearly demonstrate a large improvement in adding just 2 set elements to a single element set, and smaller improvements as set size increases. First choice prediction increases from 62 to 74 percent over 1 to 3 element sets, and to 82 percent for 7 and 15 element sets. Prediction means improve from 1.04 to 1.50 for 1 to 3 element sets, a significant increase ( $p < .03$ ). Further improvements with 7 and 15 set elements are not significant, although such improvements would be very valuable in a conjoint study.

Vagueness in product information is also expected to influence model performance. A linguistic term ("medium") is certainly more vague than a numeric attribute value (price = \$12.70). The number of linguistic attributes (1-3) selected by subjects provides a simple measure of vagueness which is graphed in Figure 2 according to the number of linguistic attributes selected. The results confirm that the fuzzy conjoint model performs well at all three levels of vagueness, while the predictive validity of the crisp conjoint model declines steadily as vagueness in attribute information increases. For subjects selecting one linguistic attribute, the fuzzy conjoint model has a higher mean prediction than the crisp model, although the t-value of the difference is not significant ( $t=0.59$ ). For two linguistic attributes, the fuzzy set mean prediction is significantly higher than the crisp mean, with a t-value of 3.31 ( $p < .003$ ), improving further for three linguistic attributes to a t-value of 3.57 ( $p < .002$ ). This relative improvement in fuzzy conjoint predictive validity is also reflected in the correlation coefficient between vagueness and prediction, which is not significant ( $-.110$ ) for the crisp conjoint model and is significant ( $.322$ ,  $p < .02$ ) for the fuzzy conjoint model. Thus the relative predictive validity of the fuzzy conjoint model improves with vagueness in attribute information, an important quality since subjects selected an average of 2.14 linguistic attributes in their top 4 attributes. The prediction improvements due to set size and linguistic attributes support the second hypothesis.

#### Market Share Prediction

Market share prediction is a very important component of conjoint analysis. Most conjoint studies use computer software to simulate choice and to compare estimated and actual market shares based on overall preference ratings of a cross-validation group of products (Green and Srinivasan 1990). The logit choice axiom is widely used to convert preference scores to choice probabilities. The probability of choosing a given alternative  $m$  from a choice set is given by

$$(4) \quad P(m) = \left( e^{y(m)} / \sum_{p=1}^6 e^{y(p)} \right)$$

where  $P(m)$  is the probability of selecting product  $m$ , given a crisp preference score  $y(m)$  and a set of six hypothetical products (Batsell and Lodish 1981). The choice probabilities are averaged across subjects to estimate overall market share. Ultimately managers need to know the market share that would result from a particular attribute level, all else being equal. Preferences can be linked to the main effects of attribute levels (e.g. price) according to Equation 4 and the stimulus design. Analysis software tracks the different choices of attributes and attribute orders relative to the fixed stimulus design among subjects to calculate overall shares for attribute levels, linking these to prototype values.

For example, the average market share for holdout products with a medium level of price can be calculated for each model and compared to \$13.36, the mean prototype elicited from subjects that selected the price attribute.

To estimate market shares for the fuzzy conjoint model, the overall preference must be converted to a crisp number to use in Equation 4. A weighted centroid de-fuzzification method was developed for this research. The membership value of each set element is weighted by a preference value equal to the domain variable index (1-15 for 15 element sets) and then summed for all elements and divided by 15. A crisp preference score is calculated as

$$(5) \quad y(m) = \sum_{n=1}^{15} (\mu_{B'}(y_n, m) \times n) / 15$$

where  $\mu_{B'}(y_n, m)$  is the fuzzy conjoint model output for product  $m$  from Equation 2. Market shares are given in Table 2 for the pizza ( $n=26$ ) and car ( $n=24$ ) product categories for the three attribute levels (labelled in terms of their evaluation as good, average or poor). All comparisons are done by subject, with the actual share computed from the subject's overall evaluations of the six holdout profiles. For all three attribute levels and both products, the fuzzy conjoint model market share correlations with the actual share are higher than the crisp model's. Four of the six fuzzy model correlations are significant at the .01 level, while none of the crisp model correlations are significant at this level and only two are significant at the .05 level.

The mean share error of the absolute difference between estimated and actual shares is also lower for the fuzzy conjoint model in every case. The average market share error is 5.09 percent for the fuzzy model and 7.64 percent for the crisp model. The crisp model estimated market shares have 50 percent more deviation from the actual share. Accurate share estimates are critical to managing existing products. For pizza, the actual share for a medium price (mean=\$13.36) is 48, compared to 24 for a low and 28 for a high price level. This suggests that a price slightly above medium would be optimal. The fuzzy conjoint model would recommend a similar optimal price based on estimated shares of 32, 42 and 26 for low, medium and high price levels respectively. The crisp conjoint model estimated shares of 40, 42 and 18 (for L/M/H levels) differ substantially from the actual shares, resulting in a much lower optimal pizza price between low and medium. The more accurate fuzzy conjoint share estimates would allow a manager to charge a higher price, increasing profits without a loss in market share. Using existing methods, the fuzzy conjoint model substantially improves market share estimation and predictive validity.

## CONCLUSIONS AND DISCUSSION

The results demonstrate the substantial benefits from using fuzzy sets to represent consumer ratings. The fuzzy conjoint model significantly improves predictive validity compared to existing conjoint models using identical data in a typical conjoint experiment, predicting the first choice of 82 percent of subjects. The largest improvements are for the more difficult task of predicting the ranking of preferences beyond first choice, which is reflected in the overall 138 percent improvement. The results are consistent across attributes types, product categories, administration methods, stimulus designs and 192 subjects. The underlying measurement properties of the fuzzy conjoint model require only ordinal information. Results show that both crisp and fuzzy conjoint models perform well using computer software that fully adjusts to the subject preferences and attributes. Linguistic attributes are clearly important to consumers, since subjects selected more than two, on average, among their top four attributes. The predictive validity of the fuzzy model does not decline when linguistic attributes are

present, as the crisp model does. The fuzzy conjoint model gives practitioners the flexibility to deal with linguistic attributes in an individual-level model with increasingly good predictive validity in situations in which current models are not suitable.

The results also demonstrate the value of identifying situations and subjects for which a fuzzy set preference model is most appropriate and of adapting conjoint analysis techniques to fuzzy set models (e.g. statistical estimation, hybrid models, product optimization). Vagueness measures may form an important part of a more general model relating predictive validity to subject, situation and preference model characteristics. Alternative preference models based on fuzzy production rule combinations of attribute values and approximate reasoning also show considerable promise (Willson 1991).

The results also show that the fuzzy conjoint model can be readily applied to marketing problems using automated software for data collection. The experimental software is easily used by subjects, providing all of the information needed to implement both crisp and fuzzy models in about 25 minutes of interaction per product category. Once data is collected, the fuzzy conjoint model is actually easier to implement and estimate than existing crisp models. Automated analysis software created for this research can read and verify returned data, generating preference predictions and market shares. In addition, the fuzzy conjoint model is an important module in a broader intelligent business system which combines the best fuzzy logic and management science models to provide enterprise-wide management systems. Improved market share and demand estimates from the fuzzy conjoint model would be an important input to manufacturing and distribution systems.

## REFERENCES

- Batsell, Richard R. and Leonard M. Lodish (1981), "A Model and Measurement Methodology for Predicting Individual Consumer Choice," Journal of Marketing Research, 18 (February), 1-12.
- Currim, Imran S. and R.K. Sarin (1984), "A Comparative Evaluation of Multiattribute Consumer Preference Models," Management Science, 30 (May), 543-61.
- Green, Paul E. (1984), "Hybrid Models for Conjoint Analysis: An Expository Review," Journal of Marketing Research, 21 (May), 155-69.
- Green, Paul E. and V. Srinivasan (1990), "Conjoint Analysis in Marketing Research: New Developments and Directions," Journal of Marketing, 54 (October), 3-19.
- Turksen, I.B. (1986), "Interval Valued Fuzzy Sets Based On Normal Forms," Fuzzy Sets and Systems, 20, 191-210.
- Turksen, I.B. (1991), "Measurement of Membership Functions and their Acquisition," Fuzzy Sets and Systems, 40 (1), 5-38.
- Willson, Ian A. (1991), "Fuzzy Sets and Uncertain Choice Processes: An Empirical Test of Models," Doctoral dissertation, Faculty of Management, University of Toronto.
- Wittink, Dick R. and Philippe Cattin (1989), "Commercial Use of Conjoint Analysis: An Update," Journal of Marketing, 53 (July), 91-96.
- Zadeh, Lofti A. (1975), "The Concept of a Linguistic Variable and its Application to Approximate Reasoning," Parts 1 and 2, Information Sciences, 9, 199-249, 301-357.

**TABLE 1: PREDICTION RESULTS**

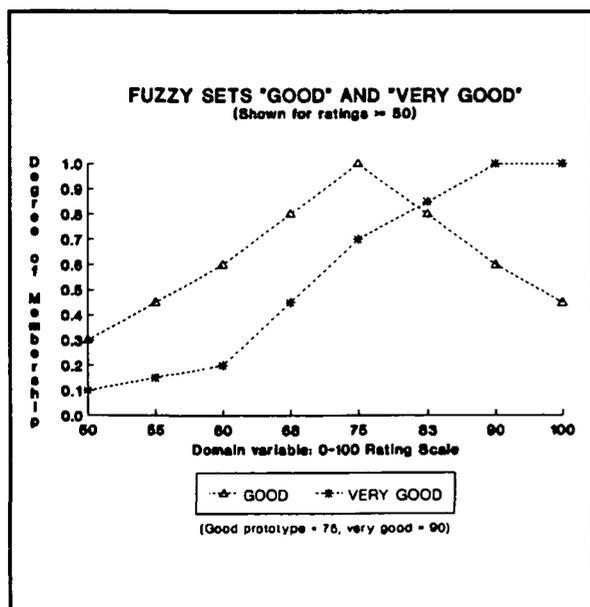
<u>Prediction Measure:</u>	<u>Naive Model</u>	<u>Crisp Model</u>	<u>Fuzzy Model</u>
1st choice rate:	16.67%	50%	82%
Sum of choices measure:	10.69	45	108
Weighted sum measure:	19.66	80	250
Mean prediction:	0.28	0.90	2.16
t-value of mean (fuzzy - crisp):			4.53 (p < .0001)

**TABLE 2: MARKET SHARE ESTIMATES BY ATTRIBUTE LEVEL**

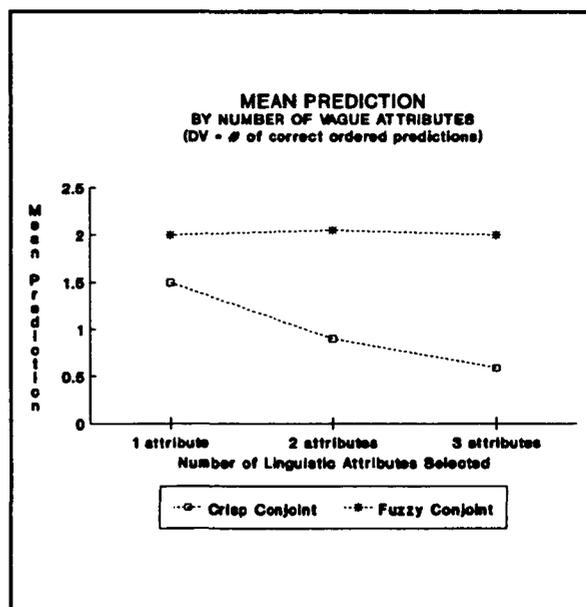
<u>Attribute Level</u>	<u>Crisp Conjoint</u>		<u>Fuzzy Conjoint</u>	
	<u>Mean Abs. Share Error</u>	<u>Correlation W/. Actual</u>	<u>Mean Abs. Share Error</u>	<u>Correlation W/. Actual</u>
<b>Pizza Product:</b>				
Good Level:	6.26	0.627	5.31	0.917 <sup>a</sup>
Average Level:	6.02	0.828 <sup>b</sup>	4.74	0.901 <sup>a</sup>
Poor Level:	4.00	0.866 <sup>b</sup>	3.32	0.906 <sup>a</sup>
<b>Car Product:</b>				
Good Level:	10.34	-0.191	5.95	0.412
Average Level:	11.73	0.044	5.98	0.298
Poor Level:	7.51	0.538	5.28	0.837 <sup>a</sup>

<sup>a</sup> p < .01. <sup>b</sup> p < .05. Abs. = Absolute Value

**FIGURE 1: FUZZY SETS GOOD/VG**



**FIGURE 2: PREDICTION BY VAGUENESS**



533-82

161833

p-8

N93-29549

## Information Compression in the Context Model

Jörg Gebhardt    Rudolf Kruse    Detlef Nauck

Department of Computer Science  
Technical University of Braunschweig  
W-3300 Braunschweig, Germany

### Abstract

The Context Model provides a formal framework for the representation, interpretation, and analysis of vague and uncertain data. The clear semantics of the underlying concepts makes it feasible to compare well-known approaches to the modeling of imperfect knowledge like given in Bayes Theory, Shafer's Evidence Theory, the Transferable Belief Model, and Possibility Theory.

In this paper we present the basic ideas of the Context Model and show its applicability as an alternative foundation of Possibility Theory and the epistemic view of fuzzy sets.

## 1 Introduction

One origin of imperfect data is due to situations, where the incompleteness of the available information does not support state-dependent specifications of objects by their characterizing tuples of elementary or set-valued attributes.

The most important kinds of imperfect knowledge to be investigated are vagueness and uncertainty. Within the Context Model [Gebhardt, Kruse 1992a, Gebhardt 1992, Kruse et. al. 1992] vagueness is referred to the specification of so-called vague characteristics, which formalize imprecise, possibly contradicting and partial incorrect observations of attribute values with respect to a finite number of conflicting consideration contexts.

The integration of conflicting contexts is related to the phenomenon of competition, whereas imprecision shows that a specialization of the context-dependent non-elementary characteristics attached to a vague characteristic is unjustified without having further information about the corresponding vaguely specified object. Hence, vagueness is the combination of two types of partial ignorance, which are the existence of conflicting contexts (to be called competition) and imprecision.

Uncertainty, on the other hand, is connected with the valuation of vague characteristics: When we have defined a vague characteristic to specify a vague observation of an inaccessible characteristic of an object's attribute in a given state, a decision maker should be enabled to quantify his or her degree of belief in this vague observation — either by objective measurement or by subjective valuation. Since we restrict ourselves to numerical, non-logical approaches to partial ignorance, the theory of measurement seems to be the adequate formal environment for the representation of uncertainty aspects.

The mentioned approach to vagueness and uncertainty modelling leads canonically to the concept of a valuated vague characteristic which is introduced in section 2 and serves as one of the foundations of the Context Model.

---

### Acknowledgements

We thank Didier Dubois for his helpful comments regarding an improvement of this paper.

Since we intend to focus our attention to *information compression aspects*, we show in which way valuated vague characteristics, and the important notions of correctness-, contradiction-, and sufficiency-preservation turn out to be helpful for establishing richer underlying semantics of Possibility Theory and the epistemic view of fuzzy sets. For this reason section 3 deals with an appropriate definition of possibility functions, while section 4 clarifies how to operate on possibility functions with the requirement of coming to most specific correct results, when correctness assumptions on the composed possibility functions are fulfilled. As an example we refer to some foundations of Fuzzy Control. Finally section 5 shows an interpretation of fuzzy sets and a justification of Zadeh's extension principle by the Context Model.

## 2 The Context Model: Basic Concepts

In this section we outline basic concepts of the Context Model as far as they are important for the other sections. The following definitions have already been motivated by the general idea of a valuated vague characteristic mentioned in the introduction.

**Definition 2.1** Let  $D$  be a nonempty universe of discourse (*frame of discernment, domain of a data type*) and  $C$  a nonempty finite set of contexts.

$\Gamma_C(D) \stackrel{Df}{=} \{\gamma \mid \gamma : C \rightarrow 2^D\}$  is defined to be the set of all vague characteristics of  $D$  w.r.t.  $C$ .

Ignoring the contexts,  $\Gamma(D) \stackrel{Df}{=} 2^D = \{A \mid A \subseteq D\}$  designates the set of all (imprecise) characteristics of  $D$ .

Let  $\gamma, \nu \in \Gamma_C(D)$  and  $A \in \Gamma(D)$ .

- (a)  $\gamma$  empty, iff  $\gamma(C) = \{\gamma(c) \mid c \in C\} = \{\emptyset\}$ ;
- (b)  $\gamma$  elementary, iff  $(\forall c \in C) (|\gamma(c)| = 1)$ ;
- (c)  $\gamma$  precise, iff  $(\forall c \in C) (|\gamma(c)| \leq 1)$ ;
- (d)  $\gamma$  contradictory, iff  $(\exists c \in C) (\gamma(c) = \emptyset)$ ;
- (h)  $\gamma$  specialization of  $\nu$  ( $\nu$  generalization of  $\gamma$ ,  $\gamma$  more specific than  $\nu$ ,  $\nu$  correct w.r.t.  $\gamma$ ), iff  $(\forall c \in C) (\gamma(c) \subseteq \nu(c))$ ;

**Definition 2.2** Let  $(C, 2^C, P_C)$  be a finite measure space that is referred to a given context set  $C$ . Each vague characteristic  $\gamma \in \Gamma_C(D)$  is called valuated w.r.t.  $(C, 2^C, P_C)$ .

**Remark** Obviously there are formal analogies, but even semantical differences to the concept of a random set recommended by Matheron [Matheron 1975] and Nguyen [Nguyen 1978]. Considering the original idea of a random set, if  $\gamma \in \Gamma_C(D)$ , then for all  $c \in C$ ,  $\gamma(c)$  should be interpreted as an indivisible *set-valued datum* attached to an outcome  $c$  of an underlying random experiment which is formalized by a probability space  $(C, 2^C, P_C)$ .

Following a reasonable interpretation of Nguyen's approach,  $\gamma(c)$  specifies the *set of single-valued data* which are possible in a context  $c$ , where  $P_C(\{c\})$  quantifies the (objective or subjective) probability that  $c$  is the "true" context.

On the other hand, using  $\gamma$  as a valuated vague characteristic,  $P_C(\{c\})$  reflects the degree of *reliability* that the context  $c$  delivers a *correct* specification of an original characteristic  $Orig_\gamma \subseteq D$  (i.e.:  $Orig_\gamma \subseteq \gamma(c)$ ), where  $Orig_\gamma$  is an (inaccessible) state-dependent characterization of an object of interest.

Whenever  $P_C(\{c\})$  stands for a reliability degree, then  $P_C$  in general will neither be defined as a probability measure nor be normalized to a probability measure. Furthermore the interpretation of a valuated vague characteristic does not require that one of the available contexts is the "true" one which has to be selected.

### 3 Possibility Functions

The main application of (valuated) vague characteristics  $\gamma \in \Gamma_C(D)$  refers to the specification of a vague observation of an (inaccessible) characteristic  $Orig_\gamma \subseteq D$ , the so-called *original* of  $\gamma$  which — generally speaking — characterizes an object in its actual state. As an example consider a control system with a single input variable and a single output variable taking their values on the domains  $X$  and  $Y$ , respectively. The state of this control system may be defined by the actual input value  $x_0 \in X$  and the control function  $g : X \rightarrow Y$  that relates the possible input values  $x \in X$  to their corresponding output values  $y \in Y$ .

The behaviour of the system can be specified by the inference mechanism that transfers  $x_0$  to the actual output value  $y_0 = g(x_0)$ , which is

$$\begin{aligned} \text{infer} &: \Gamma(X) \times \Gamma(X \times Y) \rightarrow \Gamma(Y), \\ \text{infer}(X_0, R) &\stackrel{Df}{=} \{y \mid (x, y) \in R \cap X_0 \times Y\}. \end{aligned}$$

In the special case  $X_0 = \{x_0\}$  and  $R = g \subseteq X \times Y$  we in fact obtain

$$\text{infer}(X_0, R) = \text{infer}(\{x_0\}, g) = \{g(x_0)\}.$$

In the situation (well-known from fuzzy control) when  $g$  and sometimes even  $x_0$  are not available, but only vaguely observed, the context model suggests the specification of vague characteristics  $\gamma_1 \in \Gamma_{C_1}(X)$  and  $\gamma_2 \in \Gamma_{C_2}(X \times Y)$  based on appropriate context measure spaces  $\mathcal{M}_1 = (C_1, 2^{C_1}, P_{C_1})$  and  $\mathcal{M}_2 = (C_2, 2^{C_2}, P_{C_2})$ .

The adequate choice of context measure spaces is an application-dependent problem, but for our example it seems to be convincing that the contexts have to be defined by their maximum measurement tolerance, namely the maximum distance between the measured input value and the original input value that should have been taken.

In practical applications incomplete information and the complexity of required operations will often advise us to avoid the detailed consideration of the underlying context measure spaces, but to use an *information compressed* specification of valuated vague characteristics, as done — from the context model's point of view — in Possibility Theory [Dubois, Prade 1988] and Fuzzy Set Theory [Klir, Folger 1988].

Viewing a valuated vague characteristic  $\gamma \in \Gamma_C(D)$  in a pure formal sense as a *generalized random set*, one promising way of coming to an information compressed representation of  $\gamma$  is the choice of the *contour function* of  $\gamma$ , which we prefer — for semantical reasons — to be denoted as the *possibility function* of  $\gamma$ .

**Definition 3.1** Let  $\gamma \in \Gamma_C(D)$  be valuated w.r.t.  $\mathcal{M} = (C, 2^C, P_C)$ . Then,

$$\pi_{\mathcal{M}}[\gamma] : D \rightarrow \mathbb{R}_0^+, \pi_{\mathcal{M}}[\gamma](d) \stackrel{Df}{=} P_C(\{c \in C \mid d \in \gamma(c)\})$$

is called the possibility function of  $\gamma$ , where  $(\mathbb{R}_0^+ \stackrel{Df}{=} \{r \in \mathbb{R} \mid r \geq 0\})$ .

$POSS(D) \stackrel{Df}{=} \{\pi \mid \pi : D \rightarrow \mathbb{R}_0^+ \wedge |\pi(D)| \in \mathbb{N}\}$  is defined to be the set of all possibility functions w.r.t.  $D$ .

For  $\pi \in POSS(D)$ ,  $Repr(\pi) \stackrel{Df}{=} \{(\alpha, \pi_\alpha) \mid \alpha \in \mathbb{R}_0^+\}$  with the  $\alpha$ -cuts  $\pi_\alpha \stackrel{Df}{=} \{d \in D \mid \pi(d) \geq \alpha\}$  denotes the identifying set representation of  $\pi$ .

Let  $\gamma \in \Gamma_C(D)$  be the vague characterization of an elementary original  $Orig_\gamma \in \Gamma(D)$ . Obviously, for all  $d \in D$ ,  $\pi_{\mathcal{M}}[\gamma](d)$  quantifies the measure of all contexts  $c \in C$ , for which a specialization of  $\gamma(c)$  into the elementary characteristic  $\{d\}$  is feasible. In other words:  $\pi_{\mathcal{M}}[\gamma](d)$  is the measure of all context that do not contradict  $\{d\}$  to be the original of  $\gamma$  and therefore expresses the *possibility* that  $Orig_\gamma = \{d\}$  is valid. That is one reason why we call  $\pi_{\mathcal{M}}[\gamma]$  a *possibility function*.

But there is even more behind  $\pi_{\mathcal{M}}[\gamma]$  than only measuring possibility degrees. Whenever each context valuation  $P_C(\{c\})$  is expected to be the presupposed *reliability degree* with which  $c$  delivers a *correct* imprecise characterization  $\gamma(c)$  w.r.t.  $Orig_\gamma$  (which means that  $Orig_\gamma \subseteq \gamma(c)$ ), then, for all  $\alpha \geq 0$ , the  $\alpha$ -cut  $\pi_{\mathcal{M}}[\gamma]_\alpha$

is the most specific characteristic that is for sure correct w.r.t.  $Orig_\gamma$ , if the  $\alpha$ -correctness of  $\gamma$  w.r.t.  $Orig_\gamma$  is given (which means that the measure of all contexts  $c \in C$  that are correct w.r.t.  $Orig_\gamma$  equals  $\alpha$  or is greater than  $\alpha$ ).

**Definition 3.2** Let  $\gamma \in \Gamma_C(D)$  be valuated w.r.t.  $(C, 2^C, P_C)$  and  $A, B \subseteq D$  two characteristics. Furthermore let  $\alpha \geq 0$ .

- (a)  $B$  is correct w.r.t.  $A$ , iff  $A \subseteq B$ .
- (b)  $\gamma$  is  $\alpha$ -correct w.r.t.  $A$ , iff  $P_C(\{c \in C \mid A \subseteq \gamma(c)\}) \geq \alpha$ .

The choice of an appropriate correctness level  $\alpha^*$  depends on the semantical environment in which  $\gamma \in \Gamma_C(D)$  is used. If  $C$  is a set of outcomes of an underlying random experiment, then  $P_C(\{c\})$  quantifies the probability of the outcome  $c$ .

In this case exactly one of the contexts contained in  $C$  is selected to be the "true" context, and  $P_C$  should be seen as a probability measure (i.e.  $P_C(C) = 1$ ).

In a more general sense  $C$  is a set of contexts that represent distinguishable consideration points of view (e.g. experts, sensors). Then it is of course not always reasonable to talk about the existence of a single true context, but rather to interpret  $P_C(\{c\})$  as the degree of success with which the context  $c \in C$  has delivered correct imprecise characterizations  $\gamma_i(c)$  w.r.t. a number of checkable representative vague observations  $\gamma_i \in \Gamma_C(D)$  of original characteristics  $Orig_{\gamma_i} \subseteq D, i = 1, \dots, n$ .

If we define

$$\begin{aligned} \alpha^{(i)} &\stackrel{Df}{=} \max\{\alpha \mid Orig_{\gamma_i} \subseteq \pi_{\mathcal{M}}[\gamma_i]_\alpha\}, \quad i = 1, \dots, n, \text{ and} \\ \alpha_{\min} &\stackrel{Df}{=} \min\{\alpha^{(i)} \mid i \in \{1, \dots, n\}\}, \\ \alpha_{\max} &\stackrel{Df}{=} \max\{\alpha^{(i)} \mid i \in \{1, \dots, n\}\}, \end{aligned}$$

then  $\alpha^* \in [\alpha_{\min}, \alpha_{\max}]$  seems to be an acceptable choice for the postulation of the correctness degree of future vague characterizations  $\gamma \in \Gamma_C(D)$  w.r.t. their (inaccessible) original  $Orig_\gamma \subseteq D$ .

## 4 Operating on Possibility Functions

In the previous section we introduced the concepts of a possibility function and the correctness of (vague) characteristics  $\gamma \in \Gamma_C(D)$  with respect to their underlying original characteristics  $Orig_\gamma \subseteq D$ .

Now we change over to the important question how to operate on possibility functions. For this reason let us again come back to our control system example. We assumed to have the vague characterization  $\gamma_1 \in \Gamma_{C_1}(X)$  of the actual input value  $x_0 \in X$  and the vague characterization  $\gamma_2 \in \Gamma_{C_2}(X \times Y)$  of the control function  $g \subseteq X \times Y$ , referred to the context measure spaces  $\mathcal{M}_1 = (C_1, 2^{C_1}, P_{C_1})$  and  $\mathcal{M}_2 = (C_2, 2^{C_2}, P_{C_2})$ , respectively.

Following the notion of the context model, the starting point in fuzzy control is to neglect  $\gamma_1$  and  $\gamma_2$ , and to restrict the attention to the induced possibility functions  $\pi_{\mathcal{M}_1}[\gamma_1]$  and  $\pi_{\mathcal{M}_2}[\gamma_2]$ . Postulating  $\alpha_1$ -correctness of  $\gamma_1$  w.r.t.  $\{x_0\}$  and  $\alpha_2$ -correctness of  $\gamma_2$  w.r.t.  $g$ , we intend to calculate the most specific set  $Y_0 \subseteq Y$  of output values which is correct w.r.t.  $\{g(x_0)\}$ .

In the final decision making process one of the elements contained in  $Y_0$  has to be selected as the adequate output value of the system. Note that — as we handle imprecision as well as conflicting contexts — in the normal case we have no chance to obtain a single output value from the inference mechanism. The choice of an element of  $Y_0$  as the actual output value corresponds to the defuzzification step in applied Fuzzy Control.

For the calculation of  $Y_0$  we consider the more general environment, where  $\gamma_i \in \Gamma_{C_i}(D_i)$  are valuated w.r.t.  $\mathcal{M}_i = (C_i, 2^{C_i}, P_{C_i}), i = 1, \dots, n$ . Each  $\gamma_i$  is interpreted as a valuated  $\alpha_i$ -correct specification of a vague

observation of an inaccessible non-empty characteristic  $A_i \subseteq D_i$ . Furthermore let  $f : \prod_{i=1}^n \Gamma(D_i) \rightarrow \Gamma(D)$  be a function of imprecise characteristics. Suppose to have the task to determine the most specific characteristic in  $\Gamma(D)$  which is correct w.r.t.  $f(A_1, \dots, A_n)$ . This characteristic is called *sufficient* for  $f$  w.r.t.  $(\gamma_1, \dots, \gamma_n)$  and  $(\alpha_1, \dots, \alpha_n)$ . We now formalize the notion of sufficiency and show how to evaluate sufficient characteristics.

**Definition 4.1** Let  $\gamma_i \in \Gamma_{C_i}(D_i)$ ,  $i = 1, 2, \dots, n$  be valuated w.r.t.  $(C_i, 2^{C_i}, P_{C_i})$ . Consider correctness-levels  $\alpha_i > 0$ ,  $i = 1, 2, \dots, n$ , a function  $f : \prod_{i=1}^n \Gamma(D_i) \rightarrow \Gamma(D)$ , and a characteristic  $F \in \Gamma(D)$ .

(a)  $F$  is correct for  $f$  w.r.t.  $(\gamma_1, \dots, \gamma_n)$  and  $(\alpha_1, \dots, \alpha_n)$ , iff

$$\left( \bigvee (A_1, \dots, A_n) \in \prod_{i=1}^n \Gamma(D_i) \right) \left( \bigvee i \in \{1, \dots, n\} (\gamma_i \text{ is } \alpha_i\text{-correct w.r.t. } A_i) \implies F \text{ correct w.r.t. } f(A_1, \dots, A_n) \right);$$

(b)  $F$  is sufficient for  $f$  w.r.t.  $(\gamma_1, \dots, \gamma_n)$  and  $(\alpha_1, \dots, \alpha_n)$ , iff  $F$  fulfils (a) and

$$\left( \bigvee F^* \stackrel{c}{\neq} F \right) (F^* \text{ is not correct for } f \text{ w.r.t. } (\gamma_1, \dots, \gamma_n) \text{ and } (\alpha_1, \dots, \alpha_n)).$$

It turns out that under weak conditions there is an efficient computation of sufficient characteristics by application of the induced possibility functions  $\pi_{\mathcal{M}_i}[\gamma_i]$ , without explicitly referring to the underlying valuated vague characteristics and the context measure spaces  $\mathcal{M}_i$ .

Before coming to that result we state the following four (technical) definitions.

**Definition 4.2** Let  $D_1, D_2, \dots, D_n, D$  be universes of discourse and  $f : \prod_{i=1}^n \Gamma(D_i) \rightarrow \Gamma(D)$  a function.

(a)  $f$  is called correctness-preserving, iff

$$f(A_1, \dots, A_n) \subseteq f(B_1, \dots, B_n) \text{ for all } A_i, B_i \text{ with } A_i \subseteq B_i \subseteq D_i, i = 1, 2, \dots, n.$$

(b)  $f$  is called contradiction-preserving, iff

$$(\forall A_1, \dots, A_n) ((\exists i \in \{1, \dots, n\}) (A_i = \emptyset) \implies f(A_1, \dots, A_n) = \emptyset)$$

**Definition 4.3** Let  $D_1, \dots, D_n, D$  be universes of discourse and  $f : \prod_{i=1}^n \Gamma(D_i) \rightarrow \Gamma(D)$  a contradiction-preserving mapping.  $f$  is sufficiency-preserving, iff

$$f(A_1 \cup B_1, \dots, A_n \cup B_n) = \bigcup \{F \mid (\exists C_1, \dots, C_n) (F = f(C_1, \dots, C_n) \wedge (\forall j \in \{1, \dots, n\}) (C_j = A_j \vee C_j = B_j))\}$$

for all  $A_i, B_i \in \Gamma(D_i)$ ,  $i = 1, 2, \dots, n$ .

**Definition 4.4** Let  $\pi \in POSS(D)$ .  $\pi$  is correct (sufficient) for  $f$  w.r.t.  $(\gamma_1, \dots, \gamma_n)$ , iff  $(\forall \alpha > 0)$  ( $\pi_\alpha$  correct(sufficient) for  $f$  w.r.t.  $(\gamma_1, \dots, \gamma_n)$  and  $(\alpha_1, \dots, \alpha_n)$ ).

**Definition 4.5** Let  $\pi_i \in POSS(D_i)$ ,  $i = 1, \dots, n$ , and  $f : \prod_{i=1}^n \Gamma(D_i) \rightarrow \Gamma(D)$ . The possibility function  $f[\pi_1, \dots, \pi_n] : D \rightarrow \mathbb{R}_0^+$  which is determined by its identifying set representation  $Repr(f[\pi_1, \dots, \pi_n]) \stackrel{Df}{=} \{(\alpha, f[\pi_1, \dots, \pi_n]_\alpha) \mid \alpha \in \mathbb{R}_0^+\}$  with  $f[\pi_1, \dots, \pi_n]_0 \stackrel{Df}{=} D$  and  $(\forall \alpha > 0)$   $(f[\pi_1, \dots, \pi_n]_\alpha = f((\pi_1)_\alpha, \dots, (\pi_n)_\alpha))$  is called the image of  $(\pi_1, \dots, \pi_n)$  under  $f$ .

**Theorem 4.6** Let  $\mathcal{M}_i = (C_i, 2^{C_i}, P_{C_i})$ ,  $|C_i| \geq 2$ , be context measure spaces.

Additionally let  $f : \prod_{i=1}^n \Gamma(D_i) \rightarrow \Gamma(D)$  be a correctness- and contradiction-preserving mapping.  
*f* is sufficiency-preserving, iff

$$\left( \forall (\gamma_1, \dots, \gamma_n) \in \prod_{i=1}^n \Gamma_{C_i}(D_i) \right) (f[\pi_{\mathcal{M}_1}[\gamma_1], \dots, \pi_{\mathcal{M}_n}[\gamma_n]] \text{ sufficient for } f \text{ w.r.t. } (\gamma_1, \dots, \gamma_n))$$

The result is especially related to possibility functions, where  $\alpha = \alpha_1 = \alpha_2 = \dots = \alpha_n$ , but an analogous theorem holds in the case when the levels  $\alpha_i$  are chosen arbitrarily.

Since the function *infer* is sufficiency-preserving, applying the theorem to our example, the characteristic  $Y_0 \stackrel{Df}{=} \text{infer}(\pi_{\mathcal{M}_1}[\gamma_1]_\alpha, \pi_{\mathcal{M}_2}[\gamma_2]_\alpha)$  is sufficient w.r.t.  $\{g(x_0)\}$ , if  $\alpha$ -correctness of  $\gamma_1$  w.r.t.  $\{x_0\}$  and  $\alpha$ -correctness of  $\gamma_2$  w.r.t.  $g$  is given. Hence the output value of the control system has to be selected from  $Y_0$ .

## 5 Fuzzy Sets

Within the Context Model the interpretation of fuzzy sets [Dubois, Prade 1989, Dubois, Prade 1991] and the most important operations on fuzzy sets are based on the concept of valuated vague characteristics in the following way:

Let  $F(D) \stackrel{Df}{=} \{\mu \mid \mu : D \rightarrow [0, 1] \wedge |\mu(D)| \in N\}$ ,  $D \neq \emptyset$ , be the set of all fuzzy sets with finite codomain. Then  $\mu \in F(D)$  is considered to be the information compression  $\pi_{\mathcal{M}}[\gamma]$  of an underlying vague characteristic  $\gamma$  valuated w.r.t. an appropriate context measure space  $\mathcal{M} = (C, 2^C, P_C)$ , where  $P_C$  is a probability measure. Since the aim of fuzzy sets is the modelling of vague concepts like "young" and "tall", we now abstract from the existence of a vaguely observed original characteristic  $\text{Orig}_\gamma \in \Gamma(D)$  by interpreting  $\gamma$  as the specification of a vague property [Kruse, Meyer 1987, Kruse et. al. 1991a]. Nevertheless  $F(D)$  equals – at least at the formal level – a set of possibility functions, and therefore all results obtained in section 5 are applicable to fuzzy sets without affecting their special interpretation.

As examples we will investigate the union and intersection of fuzzy sets and Zadeh's extension principle [Zadeh 1975] by application of the following theorem.

**Theorem 5.1** Let  $\gamma_i \in \Gamma_{C_i}(D_i)$ ,  $i = 1, \dots, n$  be non-empty and valuated w.r.t.  $\mathcal{M}_i = (C_i, 2^{C_i}, P_{C_i})$ .

Furthermore let  $f : \prod_{i=1}^n \Gamma(D_i) \rightarrow \Gamma(D)$  be a mapping.

*f* sufficiency-preserving  $\implies$

$$\left( \forall d \in D \right) \left( f[\pi_{\mathcal{M}_1}[\gamma_1], \dots, \pi_{\mathcal{M}_n}[\gamma_n]](d) = \sup \{ \min\{\pi_{\mathcal{M}_1}[\gamma_1](d_1), \dots, \pi_{\mathcal{M}_n}[\gamma_n](d_n)\} \mid (d_1, \dots, d_n) \in \prod_{i=1}^n D_i \wedge d \in f(\{d_1\}, \dots, \{d_n\}) \} \right)$$

### Union and Intersection of Fuzzy Sets

Let  $\mu_1, \mu_2 \in F(D)$  be fuzzy sets and  $\gamma_1 \in \Gamma_{C_1}(D)$ ,  $\gamma_2 \in \Gamma_{C_2}(D)$  their underlying vague characteristics;  $\gamma_i$  is assumed to be valuated w.r.t.  $\mathcal{M}_i = (C_i, 2^{C_i}, P_{C_i})$ , where  $P_{C_i}(C_i) = 1$ ,  $i = 1, 2$ . Furthermore suppose that  $\mu_1 = \pi_{\mathcal{M}_1}[\gamma_1]$  and  $\mu_2 = \pi_{\mathcal{M}_2}[\gamma_2]$ . Consider the contradiction-preserving union of characteristics, defined by

$$f_{\cup} : \Gamma(D) \times \Gamma(D) \rightarrow \Gamma(D),$$

$$f_{\cup}(A, B) \stackrel{Df}{=} \begin{cases} A \cup B & , \text{ iff } A \neq \emptyset \wedge B \neq \emptyset \\ \emptyset & , \text{ otherwise} \end{cases}$$

Since  $f_{\cup}$  is sufficiency-preserving, we know by application of Theorem 4.6 that  $f_{\cup}[\mu_1, \mu_2]$  is sufficient for  $f_{\cup}$  w.r.t.  $(\gamma_1, \gamma_2)$ . Applying Theorem 5.1 it is easy to calculate  $f_{\cup}[\mu_1, \mu_2](d) = \max\{\mu_1(d), \mu_2(d)\}$ ,  $d \in D$ . In an analogous way we obtain  $f_{\cap}[\mu_1, \mu_2](d) = \min\{\mu_1(d), \mu_2(d)\}$ ,  $d \in D$ , with respect to the intersection

$f_{\cap}$  of characteristics;  $(\min, \max)$  appears as the well-known pair of  $t$ -norm and  $t$ -conorm often applied to define intersection and union of fuzzy sets [Klir, Folger 1988]. Using alternative assumptions regarding the underlying context measure spaces, additional  $t$ -norms and  $t$ -conorms are motivated by the Context Model.

## Extension Principle

Zadeh's extension principle [Zadeh 1975] arises as a special case of Theorem 5.1. This principle is defined as follows:

Let  $n \in \mathbb{N}$  and  $(\mu_1, \dots, \mu_n) \in [F(\mathbb{R})]^n$ . Furthermore let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ .

The fuzzy set  $f^*[\mu_1, \dots, \mu_n] \in F(\mathbb{R})$ , defined by

$f^*[\mu_1, \dots, \mu_n](t) \stackrel{\text{Df}}{=} \sup \{ \min \{ \mu_1(t_1), \dots, \mu_n(t_n) \} \mid (t_1, \dots, t_n) \in \mathbb{R}^n \wedge f(t_1, \dots, t_n) = t \}$ ,  $t \in \mathbb{R}$  is called the image of  $(\mu_1, \dots, \mu_n)$  under  $f$ , where  $\sup \emptyset \stackrel{\text{Df}}{=} 0$ .

If we interpret  $\mu_1, \dots, \mu_n$  as possibility functions of valuated vague characteristics, then there exist  $\gamma_i \in \Gamma_{C_i}(\mathbb{R})$  and context measure spaces  $\mathcal{M}_i = (C_i, 2^{C_i}, P_{C_i})$  fulfilling  $P_{C_i}(C_i) = 1$  and  $\mu_i = \pi_{\mathcal{M}_i}[\gamma_i]$ ,  $i = 1, 2, \dots, n$ . We define the sufficiency-preserving mapping  $g : \Gamma(\mathbb{R})^n \rightarrow \Gamma(\mathbb{R})$ ,  $g(A_1, \dots, A_n) \stackrel{\text{Df}}{=} f(A_1 \times \dots \times A_n)$  and obtain by application of Theorems 4.6 and 5.1 that  $f^*[\mu_1, \dots, \mu_n] \equiv f[\mu_1, \dots, \mu_n]$ , i.e.  $f^*[\mu_1, \dots, \mu_n]$  is sufficient for  $g$  w.r.t.  $(\gamma_1, \dots, \gamma_n)$ . We infer that within the Context Model the extension principle is nothing else than the description of how to get sufficiency-preserving mappings of a restricted class of sufficient possibility functions.

## 6 Concluding Remarks

In this paper we have outlined the application of the Context Model for a new interpretation of Possibility Theory and fuzzy sets. Based on context measure spaces, valuated vague characteristics, induced possibility functions, and the very important concepts of *correctness* and *sufficiency* we demonstrated how to operate on possibilistic data and how to get a new justification of the extension principle.

A short example of fuzzy control was taken to show the practical use of the mentioned ideas. The in-depth look at the whole theory will be distributed on different papers. A comprehensive presentation of the basic semantical aspects of the Context Model, and its relationships to random sets [Nguyen 1978], Dempster-Shafer-Theory [Shafer 1976, Shafer, Pearl 1990], the Transferable Belief Model [Smets, Kennes 1991], and Bayes-Theory [Pearl 1988] is already given in [Gebhardt, Kruse 1992a], whereas [Gebhardt 1992] and [Gebhardt, Kruse 1992b] contain the more detailed approach to a modified view of Possibility Theory. Concerning the semantical foundation of the heuristic methods of Fuzzy Control it turns out that under weak restrictions the well-known if-then-rules should be interpreted by their induced Gödel relations and composed by intersection. Except from the composition mechanism for the rules (which from the Context Model's point of view is rather conjunctive than disjunctive, and therefore coincides with similar composition techniques known from the field of knowledge based systems), the resulting fuzzy controller partly behaves like Mamdani's controller, but — as a consequence of the strict formal and semantical environment — it does not suffer from the inconsistencies of max-min-inference and the problem of justifying the combination of different mathematical formalisms as they are used for fuzzification, fuzzy-inference, and defuzzification (e.g. center of gravity method).

## References

- [Dubois, Prade 1988] D.Dubois, H.Prade: Possibility Theory, Plenum Press, New York (1988)
- [Dubois, Prade 1989] D.Dubois, H.Prade: Fuzzy Sets, Probability and Measurement, European Journal of Operational Research 40 (1989), 135-154

- [Dubois, Prade 1991] D.Dubois, H.Prade: Fuzzy Sets in Approximate Reasoning, Part 1: Inference with Possibility Distributions, Fuzzy Sets and Systems 40, No.1 (1991), 143-202
- [Gebhardt 1992] J.Gebhardt: Ein integrierender Ansatz zur possibilistischen Inferenz in wissensbasierten Systemen. Doctoral thesis, University of Braunschweig, Germany (1992)
- [Gebhardt, Kruse 1992a] J.Gebhardt, R.Kruse: The Context Model - An Integrating View of Vagueness and Uncertainty, Int. Journal of Approximate Reasoning , Special Issue on Fuzzy Expert Systems (to appear)
- [Gebhardt, Kruse 1992b] J.Gebhardt, R.Kruse: A New Approach to Possibility Theory, long paper (1992)
- [Klir, Folger 1988] G.J.Klir, T.A.Folger: Fuzzy Sets, Uncertainty and Information, Prentice Hall, New York (1988)
- [Kruse, Meyer 1987] R.Kruse, K.D.Meyer: Statistics with Vague Data. Reidel, Dordrecht (1987)
- [Kruse et. al. 1991a] R. Kruse, E. Schwecke, J. Heinsohn: Uncertainty and Vagueness in Knowledge Based Systems: Numerical Methods, Series Artificial Intelligence, Springer, Heidelberg (1991)
- [Kruse et. al. 1991c] R. Kruse, D. Nauck, F. Klawonn: Reasoning with Mass Distributions. In: B.D.D'Ambrosio, P.Smets, P.P.Bonissone (eds.): Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, California, 182-187 (1991)
- [Kruse et. al. 1992] R. Kruse, J. Gebhardt, F. Klawonn: Numerical and Logical Approaches to Fuzzy Set Theory by the Context Model. In: R.Lowen: Fuzzy Logic - State of the Art, Series: System Theory, Knowledge Engineering and Problem Solving, Kluwer Academic Publishers (to appear)
- [Matheron 1975] G.Matheron : Random Sets and Integral Geometry, Wiley, New York (1975)
- [Nguyen 1978] H.T.Nguyen: On Random Sets and Belief Functions. Journal of Mathematical Analysis and Appl. 65, 532-542 (1978)
- [Pearl 1988] J.Pearl: Probabilistic Reasoning in Intelligent Systems, Networks of Plausible Inference, Morgan Kaufmann, New York (1988)
- [Shafer 1976] G.Shafer: A Mathematical Theory of Evidence, Princeton University Press, Princeton (1976)
- [Shafer, Pearl 1990] G.Shafer, J.Pearl: Readings in Uncertain Reasoning. Morgan Kaufmann, San Mateo, CA (1990)
- [Smets, Kennes 1989] P.Smets, R.Kennes: The Transferable Belief Model: Comparison with Bayesian Models. Technical Report. IRIDIA, Université Libre de Bruxelles, Belgium (1989)
- [Smets, Kennes 1991] P.Smets, R.Kennes: The Transferable Belief Model, Technical Report : TR - IRIDIA - 90 - 14 (1990)
- [Zadeh 1975] L.A.Zadeh: The Concept of a Linguistic Variable and its Application to Approximate Reasoning, Inform. Sci. 8, 199 - 249, 301 - 37, Inform. Sci. 9, 43 - 80 (1975)

534-67

161834

l - 8

N93-29550

**FUZZY KNOWLEDGE BASE CONSTRUCTION THROUGH BELIEF**  
**NETWORKS BASED ON LUKASIEWICZ LOGIC**

**Felipe Lara-Rosano**  
**Instituto de Ingenieria**  
**Universidad Nacional Autonoma de Mexico**  
**Apartado 70-418 Coyoacan**  
**04510 Mexico, DF MEXICO**

**KEYWORDS:** Knowledge Representation, Fuzzy Knowledge Bases, Logical Networks, Lukasiewicz Logic.

**ABSTRACT.**

In this paper a procedure is proposed to build a fuzzy knowledge base founded on fuzzy belief networks and Lukasiewicz logic. Fuzzy procedures are developed to assess the belief values of a consequent in terms of the belief values of its logical antecedents and the belief value of the corresponding logical function and to update belief values when new evidence is available.

**INTRODUCTION.**

Expert Systems also called Knowledge-based Systems are one of the most fruitful areas of Artificial Intelligence (Graham 1991). A knowledge base is a collection of logical propositions whose relationships model the knowledge about a certain topic.

One of the principal issues in building expert systems is related to the design and construction of knowledge bases capable of modeling real knowledge situations characterized by uncertainty (Yager 1992). This uncertainty may be produced by following factors: (Lara-Rosano 1989)

a) It is impossible to assign the whole truth or the whole falsity to propositions, even to those taken as premises or starting points of a logical discourse.

- b) The logical support of a set of premises or conditions for determining a given conclusion or result is uncertain.
- c) The premises contain fuzzy terms.

In this paper a procedure to build fuzzy knowledge bases is introduced based on fuzzy belief networks and Lukasiewicz logic (Lukasiewicz & Tarski 1930).

## BELIEF NETWORKS AND FUZZY KNOWLEDGE BASES

Uncertain knowledge may be represented by a fuzzy knowledge base structured as a fuzzy belief network (Lara-Rosano 1989). Fuzzy belief networks are weighted directed acyclic graphs in which the nodes represent propositions, and the arcs express and quantify in a fuzzy manner the logical dependencies of the consequents in terms of its immediate antecedents, according to present knowledge. The logical belief functions should be drawn from a specific fuzzy logic.

Thus, if the nodes represent the propositions  $q_1, q_2, \dots, q_n$ , then each proposition  $q_i$  draws arrows from a subset  $S_i$  of propositions which are the direct logical antecedents of  $q_i$ . Each arrow has a weight that expresses the conditional belief on  $q_i$  given the belief of the corresponding logical predecessor.

For instance, consider following fuzzy knowledge, where  $q$  are propositions and the terms under brackets represent their corresponding belief values:

$q_3 =$  *If John takes a glass water and the water is contaminated with harmful bacteria then John could get sick* [ $v(q_3)=0.8$ ]

and the following uncertain (fuzzy) facts:

$q_1 =$  *John is thirsty and probably takes a glass water* [ $v(q_1)=0.7$ ]

$q_2 =$  *The water could be contaminated with harmful bacteria* [ $v(q_2)=0.6$ ]

The question is how to assess the belief value of the hypothesis:

$q_4 =$  *John could get sick.*

In this case, it is obvious that the belief value of  $q_4$  is not independent of the belief values of its antecedents, but a fuzzy function of them. The problem now is to find the most suitable multivalued logic to assess the belief value of an uncertain logical consequence in terms of the belief values of its immediate antecedents and the belief value of the implication.

From the possible multivalued logics it is argued that the most appropriate for use in fuzzy logical networks is the Lukasiewicz logic. (Lukasiewicz & Tarski 1930). In fact:

a) Lukasiewicz logic is the multivalent logic underlying Zadeh's ordinary fuzzy set theory (Giles 1976), having equivalent definitions for union (disjunction), intersection (conjunction) negation and set inclusion. (Zadeh 1965).

b) The fundamental operators & and U are commutative, associative and distributive over one another and idempotent. (Dubois & Prade 1980)

c) Lukasiewicz logic satisfies the De Morgan Laws and is compatible with the Piaget Group of logical transformations (Sinclair 1972), but does not satisfy the Middle-Excluded Law. That is, in this logic a certain proposition could be at the same time more or less true and more or less false, such as is the actual case in uncertain propositions. (Dubois & Prade 1980).

Given propositions a,b and their respective belief values  $v(a)$ ,  $v(b)$ , Lukasiewicz logic defines the following operators:

Conjunction:  $v(a \& b) = \min [v(a), v(b)]$

Disjunction:  $v(a \cup b) = \max [v(a), v(b)]$

Negation:  $v(\neg a) = 1 - v(a)$

Implication:  $v(a \Rightarrow b) = \min [1, 1 - v(a) + v(b)]$

Modus ponens:  $v(a \& [a \Rightarrow b]) = \max [0, v(a) + v(a \Rightarrow b) - 1]$

Therefore, in the former example:

$$v(q_1 \& q_2) = \min[v(q_1), v(q_2)] = \min(0.7, 0.6) = 0.6$$

$$v(q_3) = 0.8$$

and the belief value for  $q_4$ : *John could get sick* is:

$$v(q_4) = \max[0, v(q_1 \& q_2) + v(q_3) - 1] = 0.4$$

a low belief value, indicating a low possibility for John to get sick.

Due to its mathematical logical foundations, this method is as theoretically sound as the probabilistic methods (Schafer 1976) because it gives belief values for derived propositions with fully logical consistence with respect to the rest of the network.

Adopting the conceptual frame of fuzzy set theory (Zadeh 1965), we may define the uncertain implication as a fuzzy logical function such that:

a) if the premise x is true, then the conclusion y has a partial belief  $v(y) = s(x/y)$

b) if the premise  $x$  is false, then the conclusion  $y$  may have any belief value.

The value  $s(x/y)$  is on the interval  $[0,1]$  and will be called the degree of sufficiency or sufficiency value of  $x$  over  $y$ , that is the degree of support given by the true proposition  $x$  to the uncertain proposition  $y$ . It may be further interpreted as the degree of membership of  $x$  to the fuzzy set  $S$  of sufficient conditions for  $y$  to be true.

### SEMANTIC CLUSTERS OF PREMISES

Let us suppose a set of  $n$  premises  $\{x_1, x_2, \dots, x_n\}$ , each one having its own belief value  $v(x_i)$   $i=1,2,\dots,n$  and associated in a conjunctive way to support a conclusion  $y$ . Let us call  $s(x_i/y)$  the sufficiency value of premise  $x_i$  over  $y$ . In general, the conjunction  $(x_i \& x_j \& \dots)$  of two or more premises will have a specific sufficiency value  $s(x_i \& x_j \dots/y)$  over a conclusion  $y$  according to the synergistic sufficiency of the set, that is, its degree of membership to the fuzzy set of sufficient conditions for  $y$ .

In general,  $s(x_i \& x_j \dots/y)$  will be non-separable in terms of the single values  $s(x_i/y)$ ,  $s(x_j/y)$ , ... due to the synergistic effect of the conjunction on  $y$ . Moreover, the synergy will be more pronounced in certain specific conjunctive sets of premises than in others. These privileged conjunctive sets of premises with higher overall sufficiency are called *semantic clusters* and their identification among all possible conjunctive sets of premises is a matter of expertise and field knowledge.

For instance, a doctor may assign a high belief value to the hypothesis *apendicitis*, based on a semantic cluster defined by a couple of symptoms, none of which taken alone would bring high credibility to the hypothesis.

Every semantic cluster of premises defines a specific implication with its own belief value.

### EXAMPLE

For instance, let us have the following reasoning scheme:

$x_1 =$  *It is cloudy*

$x_2 =$  *The barometric pressure is low*

$y =$  *It will rain*

*If it is cloudy and the barometric pressure is low, then it is absolutely probable that it will rain.*

In this case, the conclusion whose belief value is going to be estimated is  $y$ . The supporting

premises are  $x_1$  and  $x_2$ ;  $v(x_1)$  and  $v(x_2)$  are their belief values and  $s(x_1/y)$  and  $s(x_2/y)$  are their single sufficiency values. The logical combining function of the premises is the conjunction: *it is cloudy and the barometric pressure is low*, expressed as  $(x_1 \& x_2)$ . The conjunction has an overall sufficiency value on  $y$  represented as  $s([x_1 \& x_2]/y)$ .

Given the belief values  $v(x_1) = 0.8$   $v(x_2) = 0.85$  and the overall sufficiency value  $s([x_1 \& x_2]/y) = 0.9$  to estimate  $v(y)$ , the belief value of  $y$  we need to apply the fuzzy expression:

$$v(y) = \max[0, v(x_1 \& x_2) + s([x_1 \& x_2]/y) - 1]$$

but  $v(x_1 \& x_2) = \min(0.8, 0.85) = 0.8$  and  $s([x_1 \& x_2]/y) = 0.9$ . Therefore:

$$v(y) = \max[0, 0.8 + 0.9 - 1] = 0.7$$

### EVIDENTIAL BELIEF UPDATING OF FUZZY KNOWLEDGE BASES

In real life situations the initial knowledge base normally is composed of a small set of premises with low belief values, because of lack of evidence. Later on, when evidence arrives, new premises are introduced in the knowledge base, bringing new synergistic support to other premises and modifying the belief value of the uncertain implications.

Thus, we have two different kinds of belief updating of fuzzy knowledge bases:

- a) Belief updating of fuzzy implications.
- b) Belief updating of premises.

For belief updating of fuzzy implications, the new evidence is joined to the old one, trying to identify new semantic clusters or to reinforce the existing ones. Then the new combined sufficiency values are estimated giving the new belief values for the implications.

In the case of belief updating of premises, because the new evidence,  $e$  may confirm or disconfirm the related propositions, it is useful to apply the following Bayesian formula proposed by Pearl (1986) to update the belief values:

$$v'(x) = \alpha L v(x)$$

where  $v'(x)$  is the new belief value of the proposition  $x$  under new evidence  $e$ ,  $v(x)$  is the old belief value,  $L$  is a likelihood ratio expressed by:

$$L = P(e|x) / P(e|-x)$$

where  $P(e|x)$  is the probability of occurring evidence  $e$  giving  $x$ . Thus the meaning of  $L$  is: how many times more likely would it be for evidence  $e$  to occur under  $x$  as opposed to under **not- $x$**

and  $\alpha$  is a normalizing factor:

$$\alpha = 1 / [L v(x) + 1 - v(x)]$$

The role of  $\alpha$  is to maintain the belief value  $v'(x)$  less than or equal to one. In order to  $v'(x)$  to be one the old belief value  $v(x)$  should be also equal to one and the evidence should support  $x$ , that is,  $L$  should be greater than one. If  $v(x)$  is less than one, then  $\alpha$  would be smaller than  $L v(x)$  and the new belief value  $v'(x)$  would be also less than one.

The likelihood parameter  $L$  should be assessed by an expert, taking into account the possible synergy of the new evidence with the old one.

Once the new belief values of the premises and implications are estimated according to the new evidence, the whole belief network is recalculated applying the rules of fuzzy Lukasiewicz logic, to have logical coherent belief values for every proposition in the network.

Suppose in our last example that new evidence comes with certainty:

$$x_3 = \textit{Tropical wet wind is blowing from the south}; v(x_3) = 1$$

Then we identify a new semantic cluster as  $(x_1 \ \& \ x_2 \ \& \ x_3)$  whose sufficiency value we estimate say as 0.95

Further let us suppose that this evidence will bring new confirming support to our premise  $x_1$ : *It is cloudy*. Let us estimate the likelihood ratio  $L(x_1, x_3) = P(x_1 | x_3) / P(x_1 | \neg x_3) = 3$

Then the normalizing factor for  $x_1$  is:

$$\alpha = 1 / [L v(x_1) + 1 - v(x_1)] = 1 / [(3 \times 0.8) + 1 - 0.8]$$

$$\alpha = 1 / 2.6 = 0.3846$$

The new belief value  $v'(x_1)$  will be then:

$$v'(x_1) = \alpha L v(x_1) = 0.3846 \times 3 \times 0.8 = 0.923$$

Therefore, it is more likely that it is cloudy. The belief value of the new conjunction  $(x_1 \ \& \ x_2 \ \& \ x_3)$  is:

$\min ( 0.923, 0.85, 1 ) = 0.85$

and the belief value of  $y = It\ will\ rain$  will be:

$v(y) = \max [0, v(x_1 \& x_2 \& x_3) + s([x_1 \& x_2 \& x_3]/y) - 1] = \max [0, 0.85 + 0.95 - 1] = 0.8$

The new incoming evidence has increased the belief value of *It will rain* from 0.7 to 0.8 due to an impact over one of the premises and the definition of a new semantic cluster with a higher sufficiency value.

## CONCLUSIONS

In this paper a procedure to build a fuzzy knowledge base founded on fuzzy belief networks and Lukasiewicz logic was proposed. It is based on a knowledge network structure composed by uncertain propositions interconnected by fuzzy logical functions according to their logical dependencies. Under this basis, the belief value of a logical consequent in the knowledge network is defined and fuzzy procedures are developed to assess it in terms of the belief values of its logical antecedents and the belief value of the corresponding logical function.

The procedure permits also updating of fuzzy knowledge bases when new evidence arrives. This updating is then propagated in a logical antecedent-consequent order through the network until the last conclusions are updated. For this updating a Bayesian formula developed by Pearl (1986) is applied, requiring the estimation of only one parameter. Due to the analytical support of a logical mathematical theory, the results have complete logical coherence.

## REFERENCES

- Dubois D & Prade H, (1980) "Fuzzy Sets and Systems", New York: Academic Press.
- Giles R, (1976) "Lukasiewicz Logic and Fuzzy Set Theory", *Int.J. Man-Machine Studies*, 8, 313-327.
- Graham, I. (1991) "Fuzzy Logic in Commercial Expert Systems - Results and Prospects", *Fuzzy Sets and Systems*, 40, 451-472.
- Lara-Rosano, F. (1989). "Uncertain Knowledge Representation through Fuzzy Knowledge Networks based on Lukasiewicz Logic". in "*Advances in Computer Science*", Lasker, G.E. (ed). Windsor, Canada: International Institute for Advanced Studies in Systems Research and Cybernetics.
- Lukasiewicz J & Tarski A, (1930) "Investigations in the Sentential Calculus". Translation in J. Lukasiewicz "*Selected Works*", Amsterdam: Reidel (1970).
- Pearl, J. (1986). "On Evidential Reasoning in a Hierarchy of Hypotheses". *Artificial Intelligence*, 28, 9-25.
- Schafer, G. (1976). "Mathematical Theory of Evidence". Princeton Uni. Press. Princeton,

N.J., U.S.A.

**Shorliffe E H & Buchanan B G**, (1975) "A Model of Inexact Reasoning in Medicine", *Mathematical Biosciences*, 23.

**Sinclair H**, (1972) "Piaget's Theory of Development: The Main Stages" in Murray F B (ed) *"Critical Features of Piaget's Theory of Development of Thought"* Newark: Univ. of Delaware Press.

**Yager R R**, (1992) "Expert Systems using Fuzzy Logic" in Yager R R & Zadeh L A (eds) *"An Introduction to Fuzzy Logic Applications in Intelligent Systems"*. Boston: Kluwer Academic Publishers.

**Zadeh L A**, (1965) "Fuzzy Sets", *Information and Control*, 8, 338-353

# INTELLIGENT FUZZY CONTROLLER FOR EVENT-DRIVEN REAL TIME SYSTEMS

Janos Grantner<sup>1</sup>

Department of Electrical Engineering  
University of Minnesota, Minneapolis, MN 55455

Marek Patyra and Marian S. Stachowicz

Department of Computer Engineering  
University of Minnesota, Duluth, MN 55812

## ABSTRACT

Most linguistic models known are essentially static, that is, time is not a parameter in describing the behavior of the object's model. In this paper we show a model for synchronous finite state machines based on fuzzy logic. Such finite state machines can be used to build both event-driven time-varying rule-based systems and also the control unit section of a fuzzy logic computer. The architecture of a pipelined intelligent fuzzy controller is presented, and the linguistic model is represented by an overall fuzzy relation stored in a single rule memory. A VLSI integrated circuit implementation of the fuzzy controller is suggested. At a clock rate of 30 MHz, the controller can perform 3 MFLIPS on multi-dimensional fuzzy data.

**KEYWORDS:** Fuzzy Modeling, Intelligent Fuzzy Controller, Fuzzy Logic Hardware Accelerator, VLSI Implementation

## 1. FUZZY LOGIC FINITE STATE MACHINES

The general model of a finite state machine (FSM) is illustrated in Figure 1.1. Formally, a sequential circuit is specified by two sets of Boolean logic functions:

$$f_z(X, y) \rightarrow Z, \text{ and} \\ f_y(X, y) \rightarrow Y,$$

where  $X$ ,  $Z$ ,  $y$ , and  $Y$  stand for a finite set of inputs, outputs, present and next state of the state variables, respectively. Functions  $f_z$  and  $f_y$  map the inputs and the present states of the state variables to the outputs and the next states of the state variables, respectively.

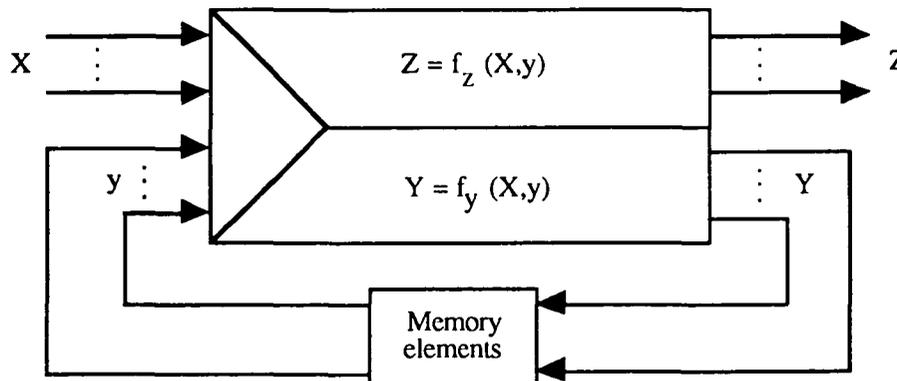


Figure 1.1 General model of a finite state machine (FSM).

The current states of the memory elements hold information on the past history of the circuit. The behavior of a synchronous sequential circuit can be defined from the knowledge of its signals at discrete instants of time. Those time instants are determined by a periodic train of clock pulses. The memory elements hold their outputs until the next clock pulse arrives.

<sup>1</sup> Visiting Professor from Technical University of Budapest, Hungary.

We extend this model by introducing membership functions and fuzzy relations to map the changes which take place in fuzzy input data to fuzzy outputs and next states of the state variables.

With the model presented in this paper, the definition of states will remain crisp, that is, the state of the system can be represented in one of the usual ways (i.e. by isolated flip-flops, registers or a microprogrammed control unit). The fuzzy outputs will be devised from a dynamically changing linguistic model since the response to a specific change at the fuzzy inputs will vary with different states of the FSM. We will refer to this model as Crisp-State-Fuzzy-Output FSM or CSFO FSM. A block diagram of the CSFO FSM is shown in Figure 1.2. X and Z stand for a finite set of fuzzy inputs and outputs, respectively.

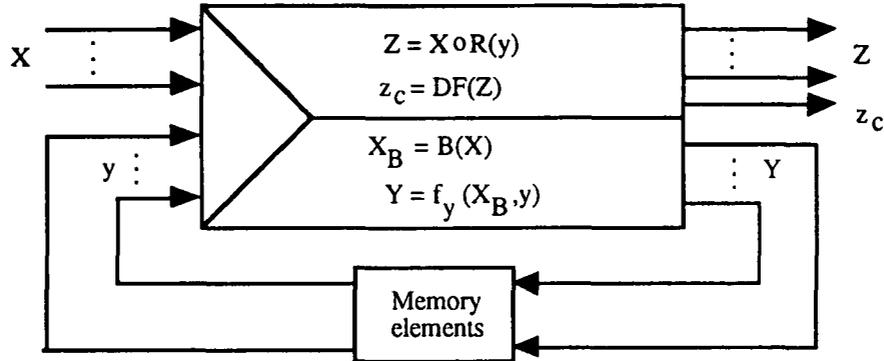


Figure 1.2 General model of the Crisp-State-Fuzzy-Output FSM (CSFO FSM).

R stands for the object's model which is now function of the y present states of the state variables, and o is the operator of composition. The  $z_c$  crisp values of the fuzzy outputs are obtained by computing the DF defuzzification strategy. B stands for the transformation which maps the linguistic values of the X linguistic (fuzzy) variables to the  $X_B$  Boolean (two-valued) logic variables. Function  $f_y$  maps both the  $X_B$  Boolean logic variables and also the y present states to the Y next states of the state variables.

To accelerate the mapping of the fuzzy inputs X to a new set of fuzzy and crisp outputs Z and  $z_c$ , respectively, (i.e. to compute fuzzy inference and the DF defuzzification strategy) our pipelined fuzzy logic hardware accelerator model [5] will also be employed with the CSFO FSM. The next states of the state variables will be devised from the present states and the  $X_B$  Boolean logic variables. For instance, a Boolean variable X1LOW is true if the position of the maximum in the membership function for linguistic variable X1 falls in the range 1 to 5. X1LOW is otherwise false.

The state transients will be completed simultaneously with the fuzzy pre-processing pipeline step (Figure 3.3). A new  $S_K$  state of the CSFO FSM will then select an overall fuzzy relation  $R_K$  which will in turn be used as the linguistic model in the fuzzy inference pipeline step while the system is in state  $S_K$ . With this model, the state variables will take their new values at the rate at which the pipeline steps proceed. The fuzzy outputs will be defuzzified in the last pipeline step.

In the course of the learning process (eq. 2.3), an overall relation  $R_I$  is created for each state  $S_I$  ( $I = 1, \dots, N$ ) of the CSFO FSM.

## 2. ALGORITHM OF CREATING A MULTIPLE-INPUT FUZZY MODEL

A linguistic model of a process can be built by software; fuzzy inference and defuzzification strategies can also be computed without using any dedicated hardware. However, in case of real-time control applications, the pure software approach may not be sufficient. We suggest a hardware accelerator for a multiple-input fuzzy logic controller. The accelerator is based upon the mathematical model as follows.

The process operation control strategy is created by analysis of input and output values, in which not only measurable quantities are taken into account but also parameters which cannot be measured, only observed [1]. On the basis of the verbal description, which is called a linguistic model, a fuzzy relation R is created:

$$R = \bigast_{I=1}^N (X_I \rightarrow Y_I). \quad (2.1)$$

In formula (2.1)  $\rightarrow$  is a symbol of the operation or operations by which fuzzy implications are defined, and the symbol  $*$  represents an operation which interprets the sentence connective ALSO.

We shall present the algorithm not only intended for creating a fuzzy model with a given verbal description is given, but also for determining the model's answer to a given input [2].

The verbal description of the process performance contains  $N$  relations, and fuzzy sets describe the particular states which occur in the verbal description of inputs  $X^{(1)}$  and  $X^{(2)}$  and output  $Y$  be given in formula (2.2). The graphic interpretations [4] of fuzzy sets  $X^{(1)}$ ,  $X^{(2)}$ , and  $Y$  are illustrated in Figure 2.1.

$$R1: \text{ IF } X^{(1)} \text{ is very small } (X^{(1)}_1) \text{ AND } X^{(2)} \text{ is medium } (X^{(2)}_1) \text{ THEN } Y \text{ is medium } (Y1) \text{ ALSO} \quad (2.2)$$

$$RN: \text{ IF } X^{(1)} \text{ is very big } (X^{(1)}_N) \text{ AND } X^{(2)} \text{ is medium } (X^{(2)}_N) \text{ THEN } Y \text{ is medium } (Y_N)$$

The paragraphs below illustrate in turn:

#### Fuzzy Learning

A method of creating fuzzy relation  $R1$  which represents the first fuzzy implication in the verbal description is interpreted as intersection. The remaining relations  $R2, R3, \dots, RN$  are created analogously by application of the same definition of fuzzy implication.

$$R1 = X1 \times Y1$$

$$\forall (u,w) \in U \times W \quad R1(u,w) = \min(X1(u), Y1(w)) \quad (2.3)$$

$$\forall u \in U \quad X1(u) = \min(X^{(1)}(u), X^{(2)}(u))$$

The final relation  $R$  (being the object's model) is obtained as the union of  $R1, R2, \dots, RN$ , since the sentence connective ALSO is defined as union.

$$R = R1 \cup R2 \cup \dots \cup RN$$

$$\forall (u,w) \in U \times W \quad R(u,w) = \max(R1(u,w), R2(u,w), \dots, RN(u,w)) \quad (2.4)$$

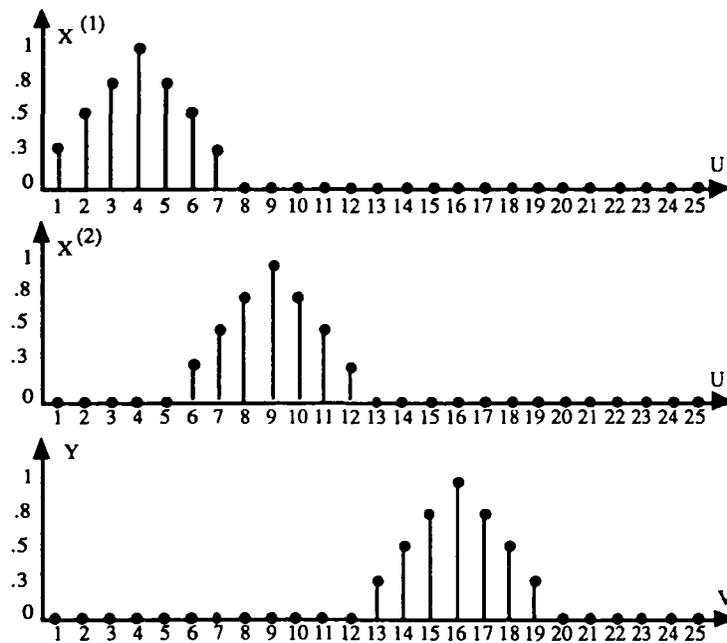


Figure 2.1. Graphic interpretations of fuzzy sets  $X^{(1)}$ ,  $X^{(2)}$ , and  $Y$ .

### Fuzzy Inference

The method of creating fuzzy answer Y to a fuzzy input X is to apply max-min composition.

$$\forall w \in W \ Y(w) = \max_{u \in U} (\min(X(u), R(u, w))) \quad (2.5)$$

### Defuzzification Strategy

The deterministic value of the answer (crisp value) is determined using the formula

$$y_c = \frac{1}{L} \sum_{j=1}^L w_j \quad (2.6)$$

where L is the number of points  $w_j \in W$  in which output set Y reaches a maximum.

## 3. HARDWARE ACCELERATOR

The hardware accelerator which performs the fuzzy learning, fuzzy inference, and defuzzification computation, that is, which maps the fuzzy inputs to fuzzy and/or crisp outputs, is summarized in this section.

Currently, in our research the degree of membership function is a discrete valued function with a 5-element domain set. With two-valued logic, three bits are used to represent each element of the set. The number of levels can be extended up to eight. The universe of discourse of a fuzzy subset is limited to a finite set with 25 elements ( $u_{\max} = w_{\max} = 25$ ). Seventy five bits are used for digitization of the membership function.

The accelerator consists of four basic units: the host interface, the fuzzy pre-processing unit, the combined fuzzy model/fuzzy inference unit, and the defuzzifier unit. The last two are referred to as the fuzzy engine [3]. The functional block diagram of the accelerator is shown in Fig. 3.1. To achieve a high processing rate for real time applications, the units are connected in a four-level pipeline.

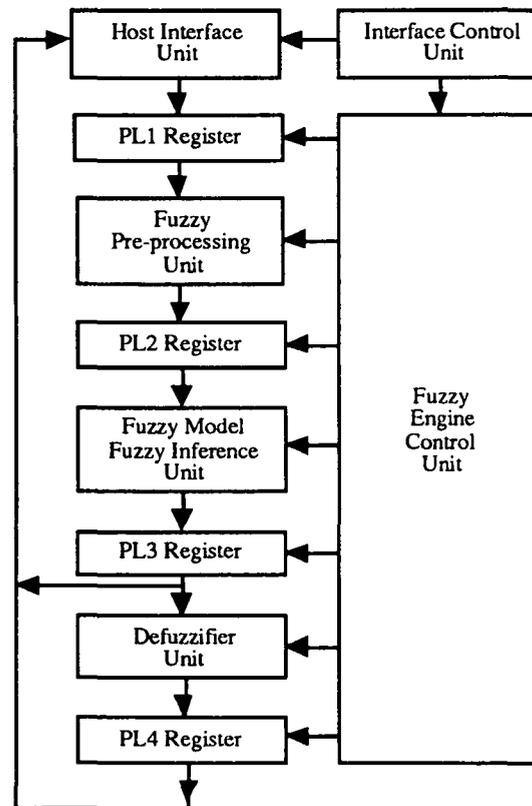


Figure 3.1. Pipeline architecture of the hardware accelerator.

The core of the hardware accelerator is a fuzzy engine which implements the formulæ (eq. 2.4) to (eq. 2.6). It is split into the fuzzy model/fuzzy inference unit and the defuzzifier unit. The functional block diagram of the fuzzy model/fuzzy inference unit (without increased parallelism) is shown in Figure 3.2.

After the XI and YI registers have been loaded, learning a multi-dimensional rule RK takes  $u_{max}$  clock periods. The MUX2 multiplexer at the input of the minimum unit selects the YI register. During the first clock step,  $u_1$  is paired with all  $w$  elements of YI and these pairs are fed to the inputs of the minimum unit. If the current rule is the first in a learning sequence, throughout the learning cycle 0 (non membership) elements will be paired with the outputs of the minimum unit and fed to the inputs of the maximum unit. The whole word of maximum values is stored at the first location of the R rule memory. During the  $j$ th clock step,  $u_j$  is compared to all  $w$  elements of YI simultaneously and the vector of the max elements is stored in the  $j$ th location of R.

If the current rule is not the first one in the learning process, the MUX3 multiplexer at the input of the maximum unit selects the  $i$ th row of R ( $1 \leq i \leq u_{max}$ ) during the  $i$ th clock step and the contents of this row in R will be updated from the outputs of the maximum unit.

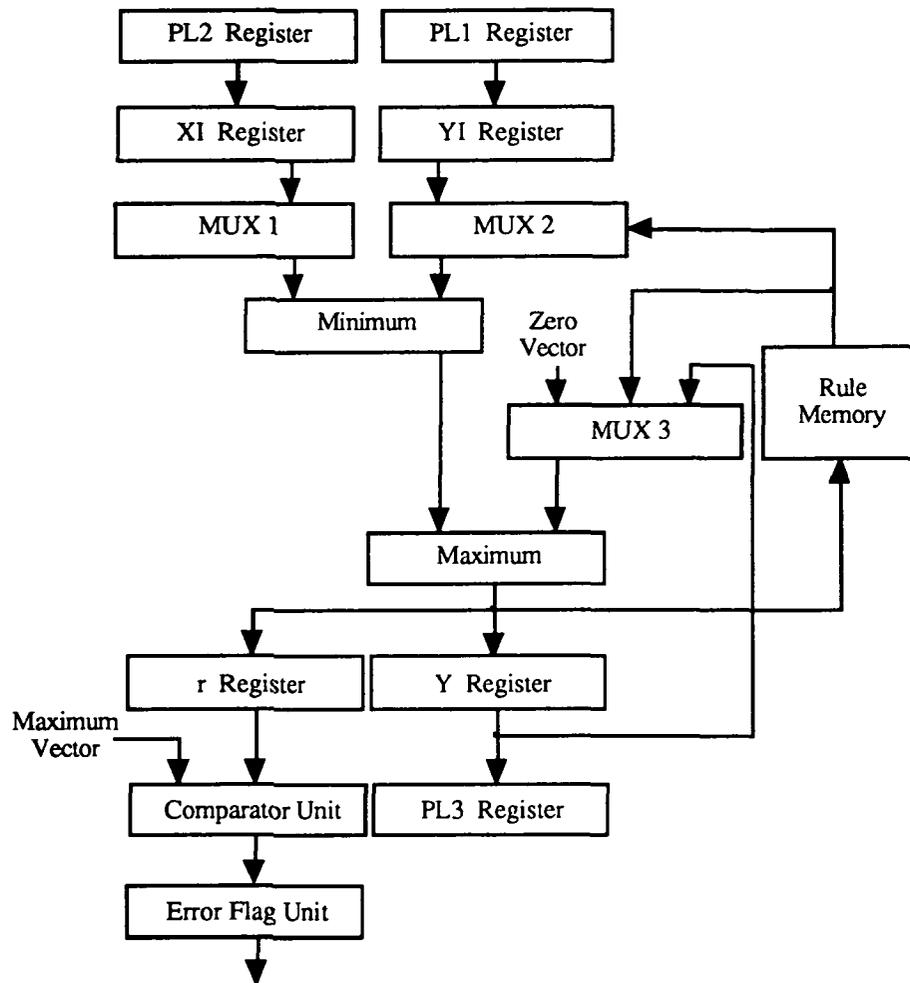


Figure 3.2 Functional block diagram of the fuzzy model/fuzzy inference unit.

Therefore the learning process of  $N$  rules takes  $N \times u_{max}$  clock periods with the architecture shown in Figure 3.2. The clock steps needed to load registers XI and YI are ignored at this point. Computing the fuzzy inference (max-min composition) also takes  $u_{max}$  clock periods. This time the MUX2 multiplexer at the input of the minimum unit pairs the  $u_j$  element of XI with all  $r$  elements of the  $i$ th row in R. If  $i = 1$  (first clock step), then the MUX3 multiplexer at the input of the maximum unit selects 0 as the other operand for each element at the output of the minimum unit. The outputs of the maximum unit are fed to

the inputs of the Y register. From the second to the last clock steps, outputs of the Y register are fed back to the inputs of the maximum unit through the MUX3 multiplexer. Contents of the R rule memory remain unchanged during the fuzzy inference process. After the last clock step, register Y holds the result of the XoR operation in the digitized fuzzy data format.

To detect whether the condition:  $\forall(u,w) \in U \times W, R(u,w) = 1$  is met, an error flag was added to the fuzzy engine. If the error flag is activated at the completion of the learning of a new rule, then all elements in the R rule memory equal 1 (full membership). This flag can be used to generate an interrupt request to the host machine. The system can then recover from this erroneous state by either downloading a "safe" model to the R memory or starting over the learning process with a modified model.

Due to the linear property of the max-min composition, by quadrupling the functional units of the basic architecture, the time required to complete the pipeline steps for either the fuzzy learning or the fuzzy inference process can be reduced to  $[u_{max}+4]+2$  clock periods.

Since the precedence relation of the subtasks (I/O data transfer ( $T_1$ ), the pre-processing of the multiple fuzzy inputs ( $T_2$ ), the learning of a new rule or the performing a fuzzy inference operation ( $T_3$ ), and the defuzzification ( $T_4$ ) are all linear operations, the four basic units of the hardware accelerator form a linear pipeline. The pipeline architecture allows the simultaneous operation of the four units. The space-time diagram in Figure 3.3 illustrates the overlapped operations of the pipeline units. Assuming that the downloading of the fuzzy data from the host system to the accelerator and the reading of the fuzzy and/or crisp output data from there (subtask  $T_1$ ) does not exceed  $[u_{max}+4]+2$  (9) clock periods, the accelerator produces new fuzzy and/or crisp output data every  $[u_{max}+4]+2$  clock periods once the pipeline is filled. Thus, at a clock rate of 30 MHz the fuzzy engine can perform over 3,000,000 fuzzy logical inferences per second with the current fuzzy data format.

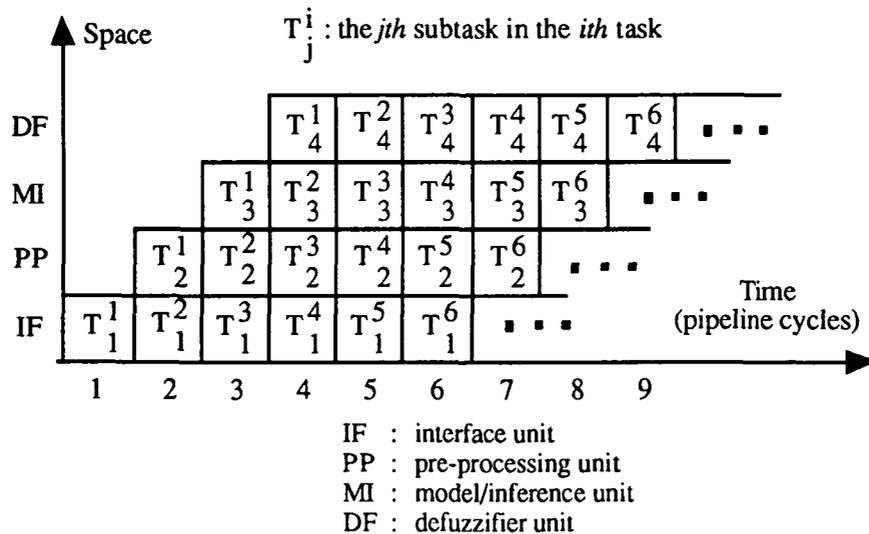


Figure 3.3. Overlapped operations of the pipeline units.

#### 4. VLSI IMPLEMENTATION

One of the most difficult issues coming from the practical realization is associated with the VLSI implementation, therefore the information provided in this section are based on our estimates and previous experience with projects of a similar nature.

Due to our objective constraints, i.e. the MOSIS service is available for chip fabrication at this time, the full design version of the proposed controller will be designed, along with a scaled-down version which will pass the constraints and will finally be fabricated.

There are two different versions of the fuzzy logic controller that could be useful in most practical implementations: a controller working stand alone (SA) or with an appropriate host computer (HC). These options will be taken into consideration.

Let us discuss the VLSI implementation issues in more detail starting with the full scale design. According to our preliminary assumption we come up with the descriptions of design signals which are summarized in Table 1.

Full scale HC version	Full scale SA version
64b scaled address/data bus	64b digital address/data bus @
-55 signal for data bus control *	32b XPROM interface
(Bus Parity), (Command)	4 XPROM control signals
(Status and CP), (Capability)	4 analog inputs %
(Synchronization)	3 signals for fuzzification control #
(Arbitration)	3 signal for inference control #
(Addresses)	2 signals for defuzzification control #
3 signals for fuzzification control #	4 signals for mode control
3 signals for inference control #	2 or 3 analog outputs
3 signals for defuzzification control #	1 CLK global clock
1 CLK global clock	1 STB strobe signal
1 RESET input	1 EN synchronization input
1 CS chip select	1 RESET input
-8 power supply inputs	-8 power supply inputs

\* We are currently working on the data bus control so this number can be changed.

# These options could be programmable.

% This number is a subject of investigation and can be changed.

@ Can be used to substitute for a single analog input/output.

Table 1. Preliminary definitions of signals for full scale versions of HC and SA Fuzzy Logic Controller.

We assume that the proposed fuzzy controller will have three basic cycles of operation: fuzzy learning, fuzzy inference and stand-by. In case of the fuzzy learning and fuzzy inference operations the HC version will be supplied with fuzzy data through the host computer which performs the fuzzification of the analog inputs. It is obvious that HC version will be able to process only digital representation of the fuzzy data prepared by the host computer. In our first approach this version will not be cascadable. The SA version of the chip will input the analog data and perform the fuzzification operation by itself. The stand-by mode will be common for both versions.

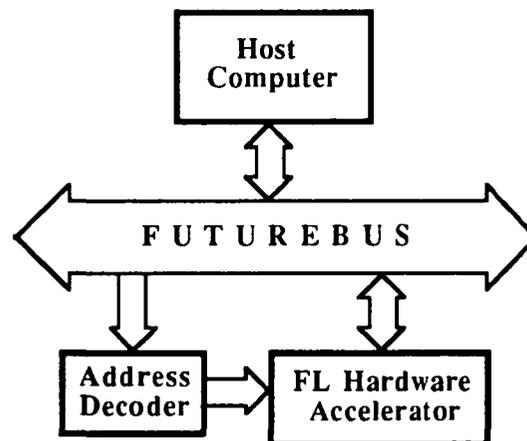


Fig. 4.1. Configuration for the Hardware Accelerator working under the host computer (HC version). One can also see that the HC version will require a very detailed design of the interface to the bus system used by the host computer for data transmission while the SA version will need an A/D converter and a few D/A converter, which will be included in the chip design. It is assumed that such a version will be communicating with the host computer through FUTUREBUS (Fig. 4.1).

Both versions has its own advantages and disadvantages basically due to communications issues, the number of pins and the design effort. It is important to point out that one can expect some instant differences in the performance of the four versions which will further be investigated in detail.

Let us now focus on the scaled down implementations of the SA version of the proposed fuzzy logic controller. There will be designed two basic modes for chip operation: normal and programmed. The normal mode will include cascaded (parallel or serial) and non cascaded operation. The block diagrams illustrating these modes are shown in Figs. 4.2 and 4.3.

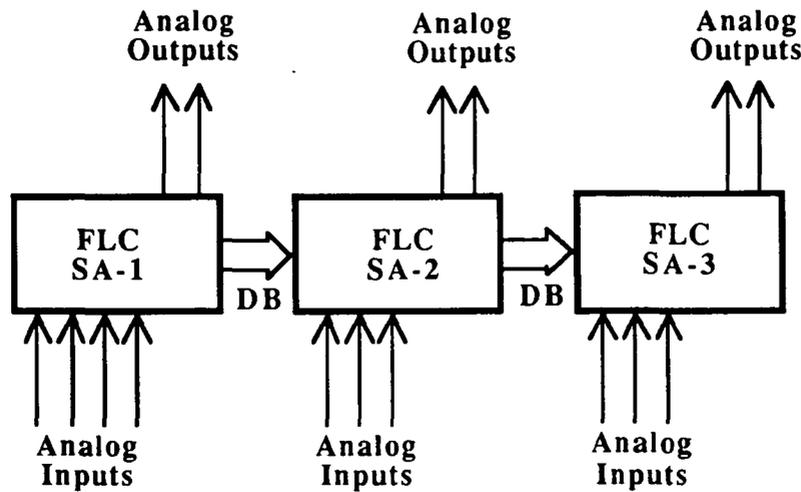


Figure. 4.2. Serial configuration for Fuzzy Logic Controller (SA version).

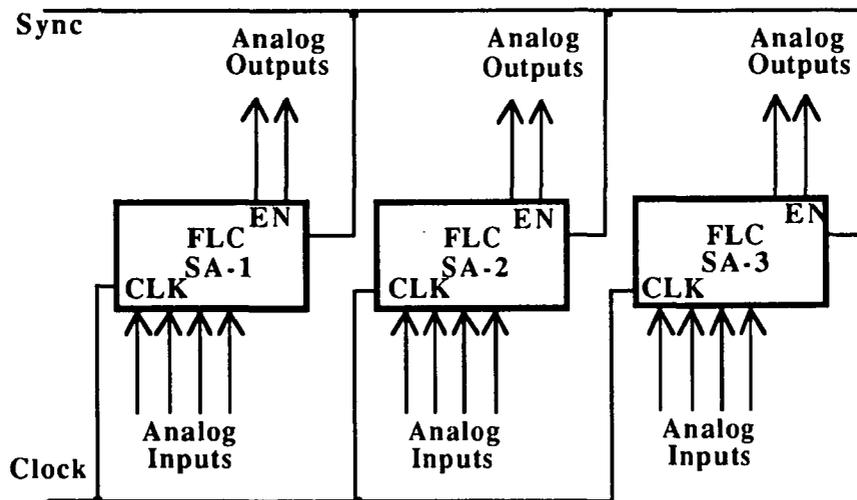


Figure. 4.3. Parallel configuration for Fuzzy Logic Controller (SA version).

In the programming mode, we assume that it will be possible to preprogram the fuzzifier, defuzzifier or inference engine, or any combination of these, in order to preserve a flexible operation tuned to the actual user. In order to achieve programmability, an EPROM/EEPROM type of memory block will be built-into the chip, and will be controlled by the external source through a memory I/O port and control signals. Our decision to fabricate this version is based on both the number of pins and also the number of signals needed to implement this version (no data interface is needed). The scaled down implementation of our design matching the objective constraints is presented as follows.

- Chip size and package

The reasonable MOSIS package has 132 pins and can contain a chip occupying 7.8mm\*9.2mm of silicon area (max). The choice of CMOS technology leads us to the variety of available processes starting from  $\lambda=1\mu\text{m}$  to  $\lambda=0.6\mu\text{m}$ . Keeping in mind the maximum signal frequency for chip operation, which was originally set between 25MHz and 30MHz, as well as the maximum chip size, the n-well, double-metal CMOS technology with  $\lambda=0.6\mu\text{m}$  will be adequate to achieve the design goals.

- Chip area and number of transistors

The maximum chip area of  $72.68\text{mm}^2$  (132 pins package) can contain about 600000 transistors for highly regular structure<sup>2</sup> with the standard CMOS technology ( $\lambda=0.6\mu\text{m}$ ). According to our estimations we will be able to put at most as four parallel fuzzy data processing paths into the chip. The single data processing path including the programming options (memoryless) is estimated to have about 50,000 transistors..

- Clock strategy and clock distribution

We decided to use single external clock signal (CLK) to generate an on-chip, two-phase non overlapping internal clock signals ( $\Phi 1$  and  $\Phi 2$ ), the two phase-clock system having the advantage of making hazard problems within the pipeline paths more easily identifiable. These phases will be distributed over the whole chip using a second metallization layer. Because the longest possible metal line is about 10mm, we chose a tree-like structure for phase distributions driven by high gain clock drivers. These drivers will be designated to drive an appropriate capacitive load of the whole clock line tree. According to the results of our previous research, the single processing path will have the ability to operate at 8 clock cycles/pipeline step. Setting the external clock rate at around 30MHz will enable us to operate the processor at a high processing rate. A future detailed investigation will help us to determine the highest possible clock rate.

- Rule memory

The major problem with the limited capacity of the internal SRAM (for HC version) of EPROM/EEPROM (for SA version) memory for the storage of rules of inferences in previous works [6-8] does not exist in our approach due to the strategy of building the global rule for the whole linguistic model described in Chapter 2. In our case only 1/4 Kbyte SRAM or EPROM/EEPROM is needed to store the global rule. Such an approach creates a luxury of increasing the parallelism of the internal structure by a factor of four, which is discussed in the next section.

It also should be noted that the idea of CSFO FSM is intended to be implemented in the SA version. Furthermore, the required extension of the rule memory (every FSM state will have assigned rule memory) will be evaluated. It is however unlikely that overall number of transistors for a single path will reach 100,000 transistors.

- Pipeline architecture

The estimated number of transistors for a single fuzzy data processing path is around 50,000. This means that the chip under consideration has the capacity containing at least four separate fuzzy data processing paths plus rule memory, which gives total estimation of about 300,000 transistors (look-up table used for defuzzification is included). The estimated area occupied by transistors is about  $45\text{mm}^2$ . The rest of the chip area will be used to provide high speed communication between processing units and the built-in memory (EPROM/EEPROM). It is expected that four parallel data paths will be designed in the chip increasing the actual speed of operation twice. In the proposed design, 3 MFLIPS performance is expected assuming the clock rate will be up to 30MHz.

## 5. CONCLUSIONS

The paper describes the general model for fuzzy state machine (FSM) which is used to formulate the fuzzy controller for event-driven real-time systems. As a result the improved architecture for fuzzy logic controller has been defined.

The improvement with respect to already published architectures [5-9] comprises in a novel strategy for fuzzy model building, which enables fuzzy inferences to be performed in a single stage of a hardware accelerator. As it has been estimated the proposed architecture, appropriately pipelined, for the hardware accelerator will profit in reaching at least 3M fuzzy logical operations per second.

The presented approach can be utilized for fuzzy controller hardware accelerators intended to work in the real time environment.

---

<sup>2</sup> Excluding the area occupied by the chip frame.

## REFERENCES

1. L. A. Zadeh, "A Fuzzy Algorithmic Approach to the Definition of Complex or Imprecise Concepts", *International Journal Man Machine Studies*, 8, pp. 249-291, 1976.
2. M. S. Stachowicz, "The Application of Fuzzy Modeling in Real-Time Expert Systems for Control", *Proc. 49th Ironmaking Conference*, Detroit, pp. 503-512, March 25-28, 1990.
3. M. S. Stachowicz, J. Grantner, L. L. Kinney, "Two-Valued Logic for Linguistic Data Acquisition", *NAFIPS Workshop '91*, University of Missouri-Columbia, Proc. pp. 168-172, May 14-17, 1991.
4. M. S. Stachowicz, M. E. Kochanska, "Graphic Interpretation of Fuzzy Sets and Fuzzy Relations", *Mathematics at the Service of Man*, (eds. A. Ballester, D. Cardins, E. Trillas), Springer-Verlag, West Berlin, pp. 620-629, 1982.
5. M. S. Stachowicz, J. Grantner, L. L. Kinney, "Pipeline Architecture Boosts Performance of Fuzzy Logic Controller", *IFSICC'92 International Fuzzy Systems and Intelligent Control Conference*, Louisville, Kentucky, Proc. pp. 190-198, March 15-18, 1992.
6. M. Togai, H. Watanabe, "Expert System on a Chip: An Engine for Real-Time Approximate Reasoning", *IEEE Expert*, pp. 55-62, Fall 1986.
7. H. Watanabe, W. D. Dettloff, K. E. Yount, "A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture", *IEEE Journal of Solid-State Circuits*, pp. 376-381, Vol. 25, No. 2, April 1990.
8. FC 110 Digital Fuzzy Processor DFPTM. Togai InfraLogic, Inc. 10/1991.
9. M. J. Patyra, "VLSI Implementation of Fuzzy-Logic Circuits", International Fuzzy Systems Association World Congress, Brussels, Belgium, June, 1991.

536-67

161836

N 93 - 29552

## FUZZY COORDINATOR IN CONTROL PROBLEMS

A. RUEDA<sup>1</sup> and W. PEDRYCZ

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
THE UNIVERSITY OF MANITOBA  
WINNIPEG, MANITOBA  
CANADA R3T 2N2

**Abstract:** In this paper a hierarchical control structure using a fuzzy system for coordination of the control actions is studied. The architecture involves two levels of control: a coordination level and an execution level. Numerical experiments will be utilized to illustrate the behaviour of the controller when it is applied to a nonlinear plant.

**Keywords:** fuzzy controller, fuzzy coordinator, hierarchical control.

### 1 INTRODUCTORY REMARKS: HIERARCHY IN CONTROL SYSTEMS

At its standard conceptual level and almost all the existing real-world applications, fuzzy controllers can be perceived as nonlinear mappings, associating current status of a system under control with an appropriate control action. They are legitimate control structures arising as a result of a certain design methodology. This allows us to emulate control abilities of a human operator. As originally proposed in [8,11,12], the fuzzy controller is a simple-level structure. Despite many algorithmic differences and a vast number of software and hardware implementations available, they are usually homogeneous with respect to handling inference and developing control actions. The design methodology is based on the derivation of control rules from the response of a process. In most of the cases, the process is already being controlled by a general purpose controller supervised by a human operator. This operator can tune the controller based on the knowledge of the status of the systems. We are concerned in this paper on emulating the coordination actions of this operator by a fuzzy system. This coordination action is a natural domain for a fuzzy system, since the decisions are taken according to a set of linguistic rules. However, we are not interested in developing a system that can tune the controller, but in one that can coordinate independent and specialized controllers. The reason for this, is that the undesirable fluctuations in the controlled variables that occur when the controller is retuned for a change in the operating point, can be avoided, by smoothly combining the response of different controllers tuned to operate under different conditions.

In this paper, we consider a control architecture that combines human expertise represented by a fuzzy system, with traditional control algorithms. In this approach the control concepts are organized hierarchically in two levels called the coordination level and the execution level [1,13,14,16]. In the coordination level, the status of the control system is being monitored, in order to decide the best control action that can be applied; while in the execution level, there are different control algorithms, each responsible for a specific control task. The response of all these algorithms is combined by the coordinator, to accomplish the control objective. A good choice for the controllers at the execution level are PID controllers, since they are widely used in practice. In this study,

1. Supported by CONACyT, Mexico, Grant #60558.

we investigate a hierarchical control structure composed of a fuzzy system and different PID controllers applied to the control of a nonlinear system.

The paper is structured as follows: the structure of the control hierarchy is introduced in Section 2; in Section 3, the application of the architecture to the control of a nonlinear system is presented; and, finally conclusions are included in Section 4.

## 2 STRUCTURE OF THE SYSTEM

The fuzzy controller operates at the higher conceptual level while "local" PID controllers are distributed as the basic components of the execution level. The example of a single input-single output system is shown in Fig. 1.

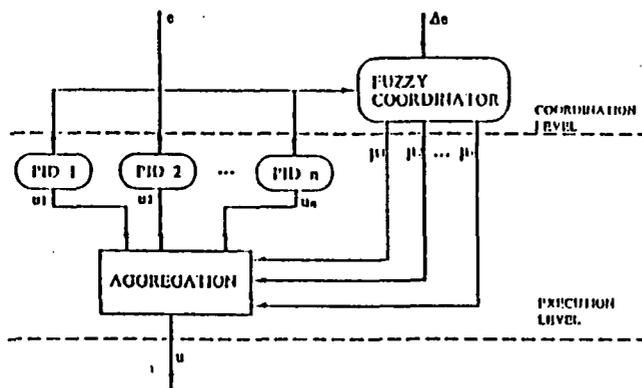


Figure 1. Structure of the system.

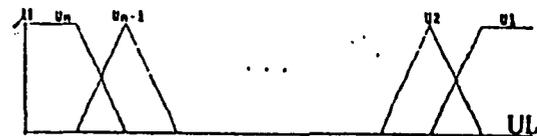


Figure 2a. Memberships for each PID.

The fuzzy controller is driven by the fuzzy sets of error  $E$  and change of error and  $\Delta E$ , defined over the universes of discourse  $UE$  and  $U\Delta E$ , and it infers a fuzzy set for selection of the controllers  $U$ , defined over the universe of discourse  $UL$ . The defuzzified variable over  $UL$  is called  $\lambda$ , and depending on its values a different combination of PID controllers becomes active. Each controller is represented in  $UL$  by a membership function. In this way the outputs of the controllers are combined by a center or area method, as shown in the following equation:

$$u = \frac{\sum_{i=1}^n u_i \mu_i(\lambda)}{\sum_{i=1}^n \mu_i(\lambda)} \quad (1)$$

where  $n$  is the number of PID controllers,  $u_i$  is the outputs of the  $i$ th PID,  $\mu_i(\lambda)$  represents the degree of membership of the  $i$ th PID controller in  $UL$ , and  $u$  is the control output. This final control signal is produced by the aggregation block visualized in Figure 1. The control rules in the fuzzy system are standard rules of the form: IF error is  $E_k$  AND change of error is  $\Delta E_k$  THEN selection is  $U_j$ ,  $k=1,2, \dots, N$ ; where  $N$  stands for the number of rules.  $E_k$  and  $\Delta E_k$  are fuzzy sets defined in the universes of discourse  $UE$  and  $U\Delta E$ .  $U_j$  is a fuzzy sets defined over the universe of discourse  $UL$ . The universe of discourse  $UL$  is partitioned into  $n$  fuzzy sets representing each of the PID controllers, as shown in Figure 2a. The rules are combined into a three-dimensional fuzzy relation  $R=E_1 \times \Delta E_1 \times U_1 + \dots + E_N \times \Delta E_N \times U_N$ . and the inference procedure utilizes the standard max-min compositional rule.

### 2.1 Case of 2 PID controllers

Consider the case of 2 PID controllers and 9 rules. The following is an example of the a set of control rules:

		error			
		N	Z	P	
change of error	N	U <sub>1</sub>	U <sub>1</sub>	U <sub>1</sub>	(2)
	Z	U <sub>1</sub>	U <sub>2</sub>	U <sub>1</sub>	
	P	U <sub>1</sub>	U <sub>1</sub>	U <sub>1</sub>	

The coordination level gives a significant preference to the PID 2 for values of error and its change close to zero, while the PID 1 is used to drive the system close to zero. All the transitions are smooth, guided by the membership functions of the fuzzy sets of error and its change.

In contrast to the coordinator implemented using fuzzy controller, we can also introduce a two-valued relay switch coordinator. It provides a Boolean character of the selection procedure, using rules of the form: IF  $\text{abs}(\text{error}) < \delta_1$  AND  $\text{abs}(\text{change of error}) < \delta_2$  THEN  $u = u_1$  ELSE  $u = u_2$ , where  $\delta_1$  and  $\delta_2$  are used to specify the point of switching.

## 3 APPLICATION TO THE CONTROL OF A WATER TANK

In this section, the hierarchical architecture is applied to the control a water tank. The control objective is to obtain good dynamical properties, such as a fast transient response free of oscillations. This is accomplished by a fuzzy coordinator in conjunction with 2 discrete-time PID controllers. Simulation results of 2 experiments are presented here. Each individual PID is tested fist, then the fuzzy system is introduced to combine both, and its response is compared to that of the relay switch.

### 3.1 Model of the system.

The water tank is shown in Figure 3. The input is the control command  $u$ , that operates the inlet valve in the range from 0 to 100%, and the output is the level  $h$ . It is consider that noise applied to system in the outlet valve, represented by  $a_{out}$ .

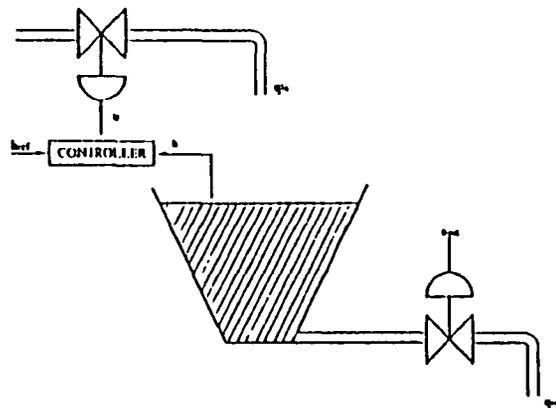


Figure 3. Water tank.

The nonlinear model of the system is given by the following equations

$$\begin{aligned} \frac{d}{dt} h &= (q_{in} - q_{out})/area \\ area &= (h+1)/7 \\ q_{in} &= q_{max} \text{ cval} \\ q_{out} &= a_{out} \sqrt{2g \max(h, 0)} \\ \text{cval} &= \begin{cases} 0 & u < 0 \\ u & 0 \leq u \leq 1 \\ 1 & u > 1 \end{cases} \end{aligned} \quad (3)$$

where  $q_{max}=1$ ,  $g=9.81\text{m/sec}^2$ , and  $a_{out}$  is random noise with a rectangular distribution defined over  $[0,0.125]$ . Notice the nonlinearities introduced by the saturation and the equation of  $area$ . This model is a modification of that one presented in [3]. The valve has a pure time delay that we model as a part of the controller. The error and change of error of the system are defined to be

$$\begin{aligned} e &= h_{ref} - h \\ \Delta e &= h_i - h_{i-1} \end{aligned} \quad (4)$$

### 3.2 The fuzzy system

The membership functions for error and change of error of the fuzzy controller are considered to be the same. Their values have been selected by experimentation. These membership functions and those for selection of the PID controllers are shown in Figure 4.

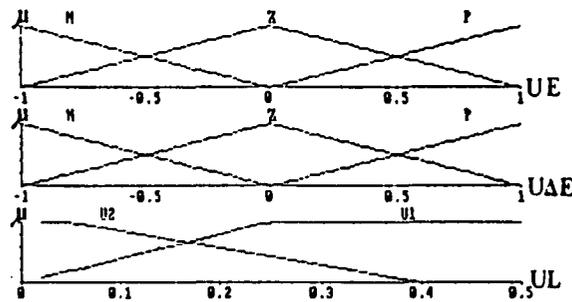


Figure 4. Membership functions for E,  $\Delta E$  and U.

### 3.3 Model of the PID controllers

A discrete-time version of the PID controllers with anti-reset windup [2] is used in the experiments. They have the following structure

$$\begin{aligned} w_i &= K_i \left[ b_i h_{ref} - h \right] + \left[ I_{i-1} + \frac{K_i \Delta t}{T_i} e \right] + \left( \frac{T d_i}{T d_i + N_i \Delta t} \right) \left[ D_{i-1} - \frac{K_i}{N_i} \Delta e \right] \\ I_{i1} &= I_{i-1} + (u_i - w_i) \frac{\Delta t}{T_i} \\ z_i &= \begin{cases} u_{min} & | \quad w_i < u_{min} \\ w_i & | \quad u_{min} \leq w_i \leq u_{max} \\ u_{max} & | \quad w_i > u_{max} \end{cases} \\ u_i &= z_{i-2} \end{aligned} \quad (5)$$

where  $i=1,2$ ,  $K_i$ ,  $T_i$  and  $T_d$  are the proportional gain, integration and derivation time respectively,  $N$  is the maximum derivative gain,  $T_t$  is the tracking constant.  $b_i$  is the set point weight factor,  $u_{max}$  and  $u_{min}$  are the maximum and minimum values of the control output, and  $\Delta t$  is the sampling period. It can observe the control output is delayed by 2 sampling periods in order to model the time delay of the valve of the tank. The PID 1 was tuned so that the response is as fast as possible, while the PID 2 was tuned in such a way that the response has good regulation properties. The values of the parameters of the PID controllers are given in the following table:

PID 1:	PID 2:	Both:	
$K_1=15$	$K_2=1$	$u_{max}=1$	
$b_1=1$	$b_2=0$	$u_{min}=0$	
$T_{i1}=0.1$	$T_{i2}=15$	$\Delta t=0.1$	(6)
$T_{t1}=0.1$	$T_{t2}=1$		
$T_{d1}=10$	$T_{d2}=10$		
$N_1=10$	$N_2=0$		

It can be observed that the nonlinearities of the plant in closed-loop with saturations and time-delays of the controllers yield an overall nonlinear system difficult to control.

### 3.4 Experiment 1

In this experiment it is considered a constant reference level  $h_{ref}=4$ . The results of the experiments are shown in Figures 5a to 5d. The PID 1 produces a fast response but with some undesirable oscillations (Figure 5a), while the PID 2 produces a slow response with better regulation (Figure 5b).

The fuzzy coordinator combines the best features of the controllers, the response is fast with good regulation properties (Figure 5c). Finally, we include the results produced by the induced relay switch (Figure 7d), switching according to the rule: IF  $\text{abs}(\text{error}) > 0.2$  THEN  $u=u_1$  ELSE  $u=u_2$ . Notice that the relay switches in the point in where the two membership functions of selection intersect each other. The response of this system with relay is quite comparable to that of the fuzzy coordinator, except that the control output is changing in an abrupt manner, which is definitely not acceptable for the actuators. In Figures 6a to 6d, it can be observed that the state trajectory of the system with the fuzzy supervisor is again a combination of those of the individual PID controllers. We have achieved a fast response, which is bounded within certain practical limits.

### 3.5 Experiment 2

In this experiment the reference level is changed following a triangular wave. These results are shown in Figure 7. We carry out the simulation in a similar way, taking PID 1 first, then PID 2, next the fuzzy supervisor with both PID controllers, and the last graph is the response with the relay. From the response of the system with PID 1, it can be observed the effect of the nonlinearities and noise of the overall system. The amplitude of the oscillations is larger close to zero than close to the maximum (Figure 7a). From the response of PID 2 we can see that the velocity of response is a factor in the performance of this controller (Figure 7b). Again, the response of the system with the fuzzy supervisor is quite remarkable, the system is able to follow the reference despite the disturbances (Figure 7c). The output of the system with relay is comparable to that of the fuzzy supervisor except that we have a not acceptable control signal, due to the fast changes (Figure 7d). In Figures 8a to 8d, the state trajectories are shown, notice that the response of the system with the fuzzy supervisor is again a combination of those of the individual PID controllers.

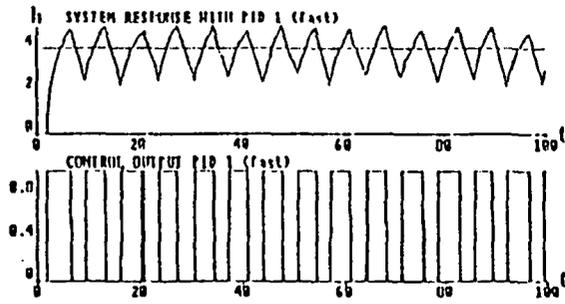


Figure 5a. Response with PID 1.

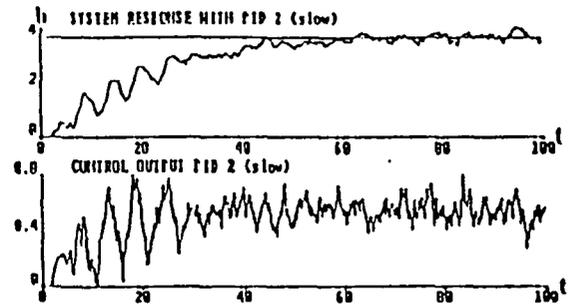


Figure 5b. Response with PID 2.

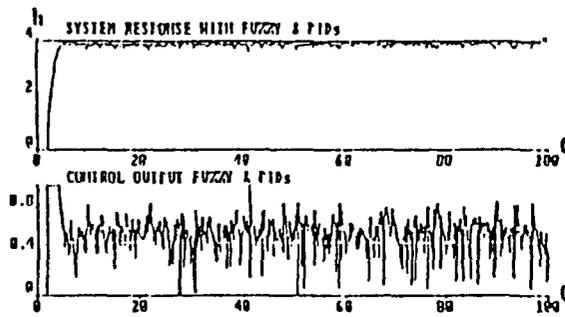


Figure 5c. Response with fuzzy coordinator.

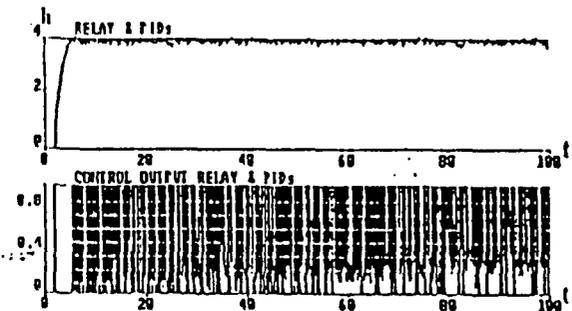


Figure 5d. Response with relay switch.

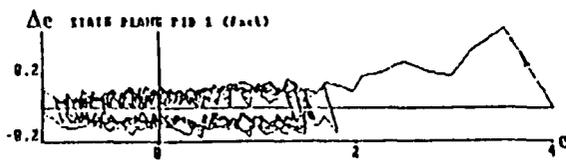


Figure 6a. State trajectory, PID 1.

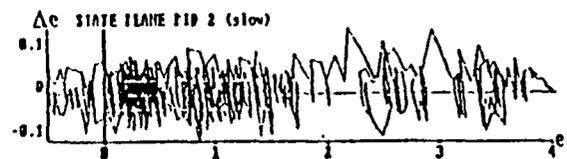


Figure 6b. State trajectory, PID 2.

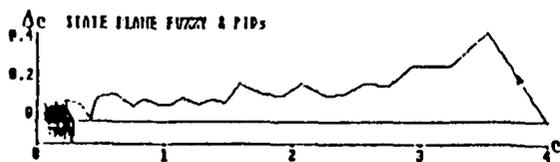


Figure 6c. State trajectory, fuzzy coordinator.

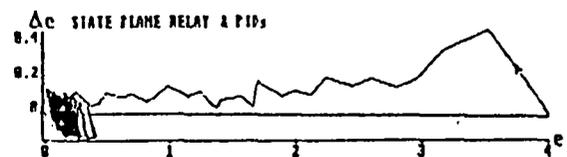


Figure 6d. State trajectory, relay switch.

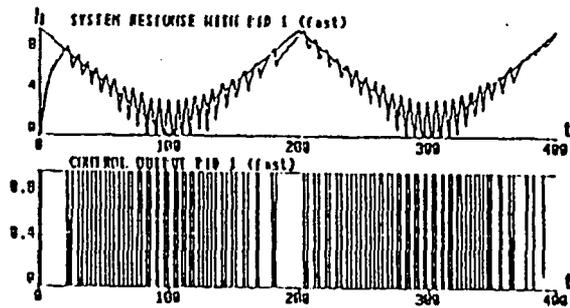


Figure 7a. Response with PID 1.

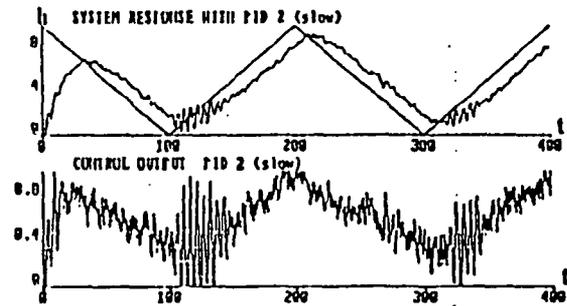


Figure 7b. Response with PID 2.

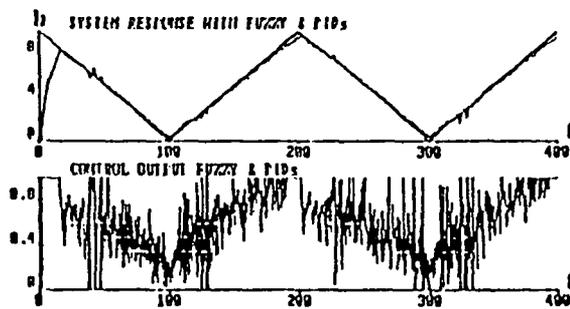


Figure 7c. Response with fuzzy coordinator.

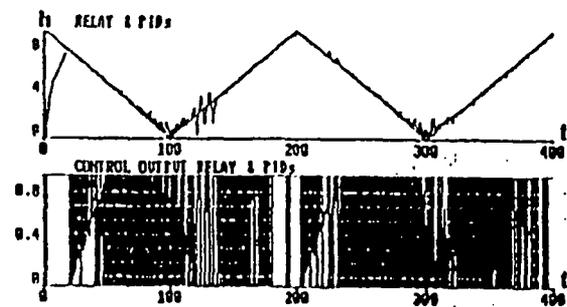


Figure 7d. Response with relay switch.

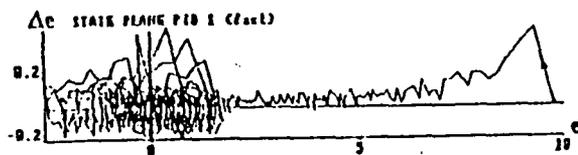


Figure 8a. State trajectory, PID 1.

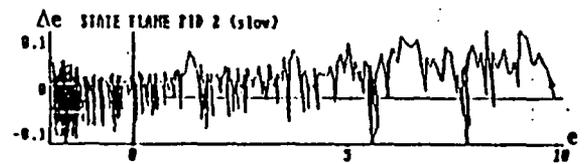


Figure 8b. State trajectory, PID 2.

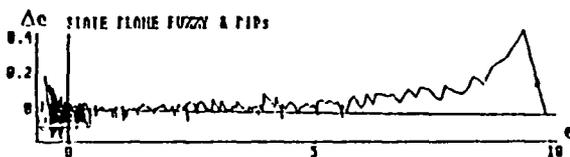


Figure 8c. State trajectory, fuzzy coordinator.

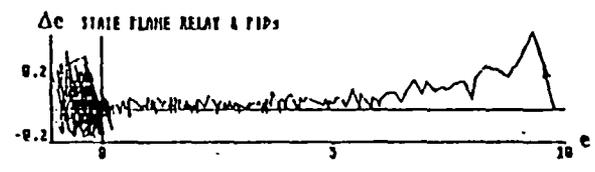


Figure 8d. State trajectory, relay switch.

## 4 CONCLUSIONS

We have discussed the hierarchical controller using a fuzzy coordinator. The results are encouraging. The fuzzy controller was found capable of combining control signals of individual PID controllers, so that the overall control characteristics are superior to those obtained for the single PID controller. The advantages of the coordinator over the relay switch were also highlighted. Further studies should lead toward enhancements in expressing control rules and calibrating the fuzzy sets included there.

## REFERENCES

- [1] Arabshahi, P., Choi, J. J., Mark II, J. J., Caudell, T. P., "Fuzzy Control of Backpropagation", IEEE Int. Conf. on Fuzzy Systems, San Diego 1992, 967-972.
- [2] Astrom, K. J., Hagglund, T., PID Controllers, Instrument Society of America, NY 1988.
- [3] Astrom, K. J., Wittenmark, B., Computer Controlled Systems, Prentice Hall, Englewood Cliffs, N. J. 1984.
- [4] Chow, K. H. F., A Self-Organizing Fuzzy Controller, M. Sc. Thesis, Dept. of Electrical Engineering, The University of Manitoba, Canada, 1984.
- [5] Elmqvist, H., SIMNON - An Interactive Simulation Program for Non-linear Systems, Proc. Simulation 77, Montreux 1977.
- [6] Gupta, M. M., Yamakawa, T. (ed.), Fuzzy Computing, Theory, Hardware, and Applications, Noth Holland, Amsterdam 1988.
- [7] Gupta, M. M., Yamakawa, T. (ed.), Fuzzy Logic in Knowledge-Based Systems, Decision and Control, Noth Holland, Amsterdam 1988.
- [8] Kickert, W. J. M., Mamdani, E. H., "Analysis of a Fuzzy Logic Controller", Fuzzy Sets and Systems, 1, 1978, 29-44.
- [9] Koivo, H. (ed), Int. Journal of Adaptive Control and Signal Processing, Special Issue: Expert Systems in Control Design and Tuning, vol. 5, no. 1, 1991.
- [10] Lee, C. C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller", Parts I and II, IEEE Trans. on Syst. Man and Cybern., vol. 20, no. 2, 1990, 404-435.
- [11] Mamdani, E. H., "Applications of Fuzzy Algorithms for Control of Simple Dynamic Plants", Proc. IEEE, 121, 1976, 1585-1588.
- [12] Mamdani, E. H., King, P. J., "The Application of Fuzzy Control Systems to Industrial Processes", Automatica, 13, 1977, 235-242.
- [13] Sugeno, M. (ed.), Industrial Applications of Fuzzy Control, North Holland, Amsterdam, 1985.
- [14] Pedrycz, W., Fuzzy Control and Fuzzy Systems (2nd edition), Research Studies Press, John Wiley, Taunton 1992.
- [15] Tong, R. M., "A Control Engineering Review of Fuzzy Systems", Automatica, 13, 1977, 559-569.
- [16] Van der Veen, J. C. T., Fuzzy Sets, Theoretical Reflections, Application to Ship Steering, M. Sc. Thesis, Dept. of Electrical Eng., Delft University of Technology, The Netherlands, 1976.
- [17] Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Trans. on Syst. Man and Cybern., vol. 3, no. 1, 1973, 28-44.

537-67  
161837

N 9 3 - 2 9 5 5 3

1-10

## Tuning a Fuzzy Controller using Quadratic Response Surfaces

Brian Schott and Thomas Whalen  
Georgia State University  
qmdbms@gsusgi2.gsu.edu

### Abstract

Response surface methodology, an alternative method to traditional tuning of a fuzzy controller, is described. An example based on a simulated inverted pendulum "plant" shows that with (only) 15 trial runs, the controller can be calibrated using a quadratic form to approximate the response surface.

### Introduction

#### Fuzzy Controller

Fuzzy controllers have received considerable attention in practice and in the literature because fuzzy rules can be framed by domain experts for narrowly defined systems. For example, Sugeno and Yasukawa said, "It supports the idea of a fuzzy model that human being can grasp input-output relations of a system qualitatively."<sup>1</sup> Although the general structure of such rules can be accomplished rather directly because of their linguistic flavor, tuning or calibration of the fuzzy variables can be very challenging. The purpose of this research is to explore an alternative method of calibration based on representing the performance of the system relative to the parameters of the controller by a sequence of quadratic functions.

We consider traditional fuzzy controllers in which the knowledge is encoded as rules comprised of combinations of subrules. The subrule  $i$  for rule  $k$  is of the form, "If  $x_i$  is  $X_i$  and  $y_i$  is  $Y_i$  then  $z_i$  is (should be)  $Z_i$ ," where lowercase letters  $x$  and  $y$  signify the names of two antecedent objects;  $X$  and  $Y$  are values of fuzzy linguistic variables describing their objects;  $z$  and  $Z$  are a consequent object and its fuzzy variable's value. The  $k$ th rule contains subrules  $i = 1, \dots, J$ , which are fused into rule  $k$  by the fuzzy operator minimum or maximum, depending on the multivalued logic employed in the system. The term set for the fuzzy values  $X$ ,  $Y$ , and  $Z$  commonly includes LARGE NEGATIVE, NEGATIVE, SMALL NEGATIVE, ZERO, SMALL POSITIVE, POSITIVE, AND LARGE POSITIVE. A typical subrule is, "If the error angle is SMALL NEGATIVE and the angular velocity is SMALL NEGATIVE, then the force of the push should be SMALL POSITIVE."

In operation the fuzzy controller is supplied the actual data values for the antecedent variables  $x$  and  $y$ ,  $x$  and  $y$ . As is usual in practice, these actual values

are assumed to be crisp numerical singletons, in this research. Also the operational controller defuzzifies rule  $k$ 's detached consequent value  $kZ$  into a crisp numerical singleton which is employed to control the "plant," the system which is being controlled. The current study uses a system that contains only one rule, with eleven subrules.

### Controller Tuning

Tuning a controller involves tweaking the several parameters which define the rules with the intention of optimizing or improving key system performance. Among the controllable parameters are the number of linguistic terms and linguistic hedges and conjunctives considered, the granularity of discretization, the method of defuzzifying, and the shape of the fuzzy variables. Many alternatives are available regarding shape: the width of the support and core; triangular vs. trapezoidal vs. sigmoidal shape; regularity vs irregularity among linguistic terms; and so on. The choices of these parameters are dependent on one another and on other system features. For example, systems based on possibilistic logics (such as Mamdani's popular system) can function well with triangular shaped fuzzy terms with slight gaps between the cores of adjacent terms (in subrules).<sup>2</sup> But a system based on Lukasiewicz' multivalued logic requires fuzzy terms with broad cores, and there must be no gaps between the cores of adjacent terms.<sup>3</sup>

Tuning can occur prior to employing the controller and adaptive learning can occur while the controller is in operation. Adaptive learning (re-tuning) is needed when the plant experiences extensive changes during use. In recent literature artificial neural networks have been suggested as tuners by several scientists, both for initial learning (see for example Kosko<sup>4</sup>, Keller & Tahani<sup>5</sup>) and adaptively (see for example Hayashi *et al*<sup>6</sup> and Berenji<sup>7</sup>). We consider an alternative tuning method based on Box and Wilson's<sup>8</sup> response surface methodology as explicated by Myers<sup>9</sup>.

### Controller Performance

The performance of the controlled system may depend on multiple factors. Common performance variables for mobile systems are fuel economy, smoothness of ride, and speed of recovery. Performance factors of the controller itself include speed, robustness, memory needs, physical dimensions, and cost. We are concerned in this study with performance factors which result from tuning decisions. We attempt to optimize system performance in relation to these criteria, or at least to satisfy the more important ones. The methodology employed assumes that the controllable factors and the performance variable are measured by continuous numeric values.

## Quadratic Response Surfaces

In theory neural network systems consider all computable functions compatible with their

$$\eta = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_{ii}^2 + \sum_{i,j:i < j} \beta_{ij} x_i x_j \quad (1)$$

architecture. In contrast, response surface methodology considers only quadratic functions. Although in using response surface methodology we reduce the quantity of alternative functions considered, we hope to take advantage of the well-studied nature of quadratic functions (based on quadratic “forms”) to improve the quality of the analysis. The rationale for using quadratic functions as approximations for unspecified functions is the Taylor series expansion of the function  $\eta$  about the point  $x_1 = x_2 = x_3 = \dots = x_k = 0$ . The assumed quadratic function is expressed algebraically in equation (1).<sup>10</sup>

The estimated quadratic function is expressed matrixly in equation (2).  $\mathbf{b}$  and  $\mathbf{x}$  are vectors with typical elements  $b_i$  and  $x_i$ ;  $\mathbf{B}$  is a symmetric matrix with typical elements  $b_{ij}/2$ . Each  $b$  in (2) is an estimate of the corresponding  $\beta$  in (1). The right side of equations (1) and (2) are called quadratic forms.

$$\hat{y} = b_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} \quad (2)$$

## Experimental Design

Experimental design is a time honored methodology cultivated by theoretical and applied statisticians.\* One of the achievements of experimental design methods is economy of sample size for multiple factor phenomena. This economy is of great interest to the tuning of fuzzy controllers, if it can be achieved without sacrificing prediction precision.

Perhaps the most naive design of a multifactor system is called “one-at-a-time”: each factor’s value is changed one at a time (holding the levels of all other factors constant). In contrast, “full factorial” experimental designs interweave the changes of all factors; if there are  $k$  factors and each factor is to be sampled at  $n$  levels, then a full factorial experiment requires a sample size of  $n^k$ . Full factorial designs are great improvements over the one-at-a-time method in reducing sample size. Even so, in practice  $n^k$  can rapidly escalate into a large quantity; the number of factors and levels are usually severely limited.

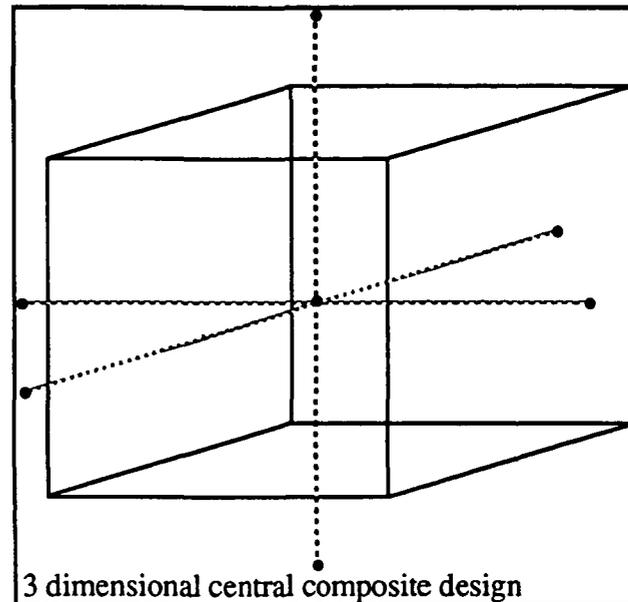
“Partial factorial” designs trim the sample size of full factorials by upwards of 50% by eliminating carefully selected sample points. But inevitably

---

\*“Control” treatments and randomization of “subjects” to treatments are among the key tenets of experimental design. Many of the desiderata of experimental design are shaded by the stochastic nature of the modelled system. In the present paper we downplay randomness and concentrate on the economical detection of dominant patterns.

partial factorial designs are unable to estimate all terms of the quadratic form; coefficients in pairs of terms are not distinguishable, but are "confounded."

To model a quadratic function every factor must be sampled by at least  $n=3$  levels in a full factorial design. But "central composite" designs (ccd) are based on an augmented  $2^k$  (not  $3^k$ ) full factorial design. Geometrically the  $2^k$  full factorial design samples all of the vertices of a  $k$ -dimensional rectangular solid. In addition to sampling points at the vertices, in the ccd the center point and "axial" points are sampled, thus augmenting the full factorial design. Axial points are found along the orthogonal lines which intersect at the centroid of the rectangular solid. With the ccd we consider, one axial point is selected outside of each face of the rectangular solid. That is, two axial points are selected along each axial line. One point is sampled where all the axial lines coincide in the center of the solid.



In a  $3^k$  full factorial design each factor is tested at three levels and in all combinations. In the ccd each factor is tested at five levels but not all factors are combined. In a  $3^k$  full factorial with  $k=3$ , the sample size is  $3^3=27$ . In the ccd the total number of sample points is  $2^k + 2k + 1$ . With  $k=3$ , the sample size is only 15. And the relative advantage of the ccd improves as  $k$  increases.

### Inverted Pendulum Example

Control of an inverted pendulum has become a common testbed among fuzzy researchers. A cart on a straight track is pushed according to the controller's instructions with varying degrees of force. A sensor detects the angle  $\varphi$  in radians that the pole makes with the vertical plumb line. The angular velocity of the pole angle is computed approximately based on the change in  $\varphi$ . Another sensor determines the cart's position  $\phi$  relative to its starting position. A pushing force  $f$  is applied to the cart.  $\varphi$ ,  $\phi$  and  $f$  can take on positive and negative values.

The fuzzy controller was constructed with eleven sub-rules containing  $\varphi$  and  $\dot{\varphi}$  as antecedent variables and with  $f$  as the consequent variable. Five terms were defined for each variable: NEGATIVE, SMALL NEGATIVE, ZERO, SMALL POSITIVE, AND POSITIVE. All fuzzy (linguistic) variables were represented as symmetrical trapezoids. The scale of the all trapezoids on each universe of

discourse were uniform relative to one another; but, the scales on different universes were independent.

Tuning of this controller was done by calibrating the scale of the axes of the three universes:  $\varphi$ ,  $\dot{\varphi}$ , and  $f$ . The criterion variable was the absolute value of the cart position  $|\phi|$  at the end of an experimental trial of 5 seconds. If the pole fell during the trial, the ending cart position was a very large number. The farther the cart moved away from its starting position, the less likely that it was in an equilibrium state. Ending cart positions near 0 were considered ideal.

The steps below are referred to as "response surface methodology." RSM is a branch of experimental design which searches for the optimal values of the explanatory variables: values of each factor which together produce the best (maximum or minimum) value of the criterion variable.

### Step 1 Select the initial set of sample points

The triads ( $\varphi$ ,  $\dot{\varphi}$ ,  $f$ ) for each of the 15 sample points in this study were set according to the central composite design. Each point corresponds to specifying the scale\* values of the 3 variables: pole angle in radians, pole angular velocity in radians per second, and pushing force in newtons. As a practical matter the factor levels were standardized so that the vertices values were expressed as +1 and -1; the centroid value is (0, 0, 0). The standardized values of the axial points were selected to produce an orthogonal design matrix,  $\pm\alpha=1.21541$ . The initial range for the variables were as follows. Pole angle: 0...0.15625. Angular velocity: 0...2. Pushing force: 0...8. The smaller the scale for  $\varphi$  and  $\dot{\varphi}$ , the more sensitive is the input sensing of the controller; and the larger the scale for  $f$ , the stronger the output of the controller.

### Step 2 Perform the experiment

We ran the controller with the simulated\*\* cart-pole "plant" 15 different times. Every experiment was run with a starting angle  $\varphi = 0.01$ , and all other transient variables set to 0. We recorded the absolute value of the final cart position for each experiment. Time was incremented every 0.02 seconds, cart mass was 1.0 Kg, pole mass was 0.1 Kg, pole length was 0.5 m, and acceleration due to gravity was 9.8 m/s<sup>2</sup>.

---

\*Each (continuous) variable's axis was discretized at 17 equally spaced values, nominally -8, -7, ..., -1, 0, 1, ..., 8. The "scale" value is the distance between adjacent discretization points.

\*\*The simulation was based on equations provided by Hamid Berenji. The differential equations can be found in Berenji's article cited in the references. The simulation assumed a frictionless plant.

### Step 3 Fit a quadratic form to the experimental results

We used the least squares criterion to fit a regression surface. In the case of  $k=3$ , there are 10 regression coefficients to be estimated. The form of the fit regression function is expressed matrically in equation (2).

### Step 4 Find the "stationary point" of the quadratic form

The stationary point is  $\mathbf{x}_0 = -\mathbf{B}^{-1}\mathbf{b}/2$ . The stationary point may be inside or outside of the convex envelope enclosing the experimental region. The stationary point may correspond to a maximum, a minimum, or to a saddle point.

In the example reported on here, the stationary point was typically a saddle point. A typical value of  $\mathbf{x}_0$  is  $(-0.129, -0.503, -0.0822)$  and was near the centroid.

### Step 5 Reduce the response surface to canonical form

"Canonical analysis" is used to reduce the response surface to canonical form by determining the eigenstructure of the matrix  $\mathbf{B}$ . If all of the eigenvalues (characteristic roots) are positive, the stationary point indicates a minimum; if all are negative, the stationary point indicates a maximum; otherwise, a saddle point has been found. A typical case produced eigenvalues 9.71706, -4.90507, and -6.70762. This suggests a saddle point.

The stationary point and the response surface can be interpreted in terms of its canonical form. If, for example, we are seeking a minimum and the stationary point indicates a minimum and the stationary point is inside the experimental region, interpretation of the results are relatively straightforward. If, on the other hand, we are seeking a minimum and the stationary point does not indicate a minimum or the stationary point is outside the experimental region, interpretation of the results is more complex.

The signs and magnitudes of the eigenvalues of matrix  $\mathbf{B}$  provide considerable information about the region of the surface in the vicinity of the stationary point. This information is oriented not to the original reference axes, but to the axes described by the eigenvectors. Each eigenvalue has a corresponding eigenvector. If an eigenvalue is negative, then movement in either direction along the corresponding defined axis, produces a decrease in the value of the response variable. An opposite, analogous interpretation applies for a positive eigenvalue. If the magnitude of the eigenvalue is large relative to other eigenvalues, then movement away from the stationary point along the corresponding axis has greater sensitivity than movement along other axes. If one of the eigenvalues is very close to 0, then the stationary point may resemble

more of a near-stationary ridge. This may afford the decision maker considerable latitude in controller tuning.

Although the experiment is supposed to be designed so that the stationary point is inside the experimental region or at least close by, the system may not behave as expected. Evaluation of the eigenstructure may provide import clues regarding the location of additional experimentation.

#### Step 6 Use ridge analysis to further interpret the response surface

Often analysis of the canonical form suggests that additional experimentation is needed because, for example, the stationary point appears to be a saddle point.\* If additional experimentation is indicated, a "ridge analysis" may suggest the direction in which to move in order to select future sampling points. Myers suggests references by Hoerl<sup>11</sup> and Draper<sup>12</sup>.

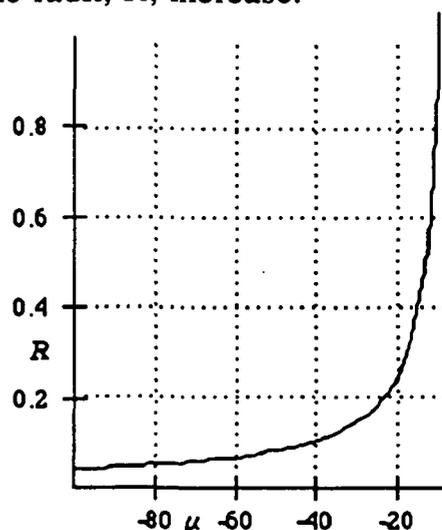
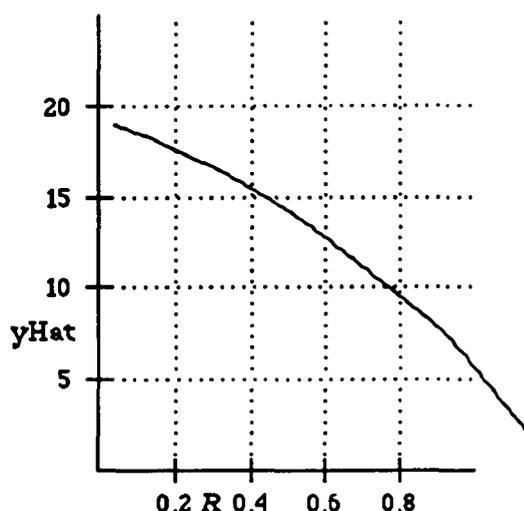
To perform a ridge analysis is to perform a constrained optimization; optimize the quadratic function restricting the solutions to being on (hyper)spheres of varying radii. The spheres are centered at the stationary point. To minimize the response, then for each different radius, plot the values of  $\hat{y}$  against  $R$ . Also plot the values of the  $x$  which correspond to each radius. For example, to minimize when the stationary point suggests a saddle point, move in the direction of decreasing response along a "ridge" defined by the series of radii.

The ridge analysis can be modeled using the method of Lagrangian multipliers. The constraint can be expressed  $x'x - R^2 = 0$ . The function  $F = \hat{y} - \mu(x'x - R^2)$  can be optimized. In practice, the plotting of the solutions of this optimization is a parametric plot.  $\hat{y}$  is a function of  $x$ , as is  $R$ ; in addition  $R$  is constrained by (is a function of)  $\mu$ ,  $R(\mu)$ . Each value of  $\mu$  determines a radius  $R$ , and the optimal value of  $\hat{y}$  is determined by that radius. This can be done by selecting values of  $\mu$  first, then determining the values of  $x_j = b_j/2\mu$  which follows from requiring the partial derivatives in the Lagrangian method to equal 0. The range of possible values of  $\mu$  is determined by whether you wish to maximize or minimize. For maximization, the values of  $\mu$  must be larger than the largest eigenvalue; for minimization, the values of  $\mu$  must be smaller than the smallest eigenvalue. With the eigenvalues 9.71706, -4.90507, and -6.70762,  $\mu$  must be less than -6.70762. The plots below show that the predicted value of the

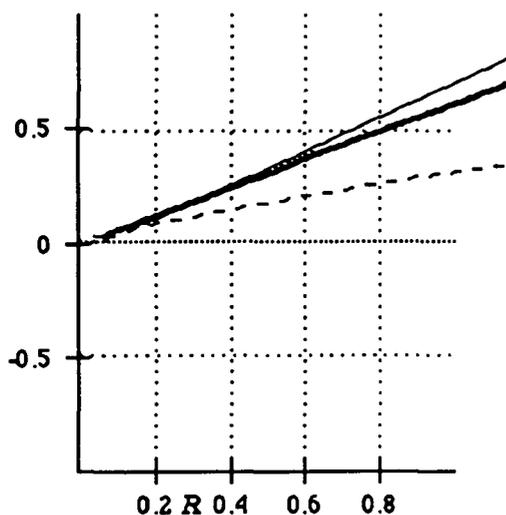
---

\*A saddle point may be an indication of multiple extrema; such a phenomenon is not consistent with models of the quadratic form.

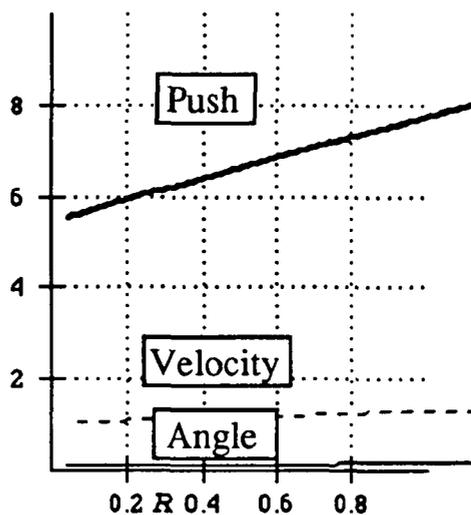
response surface,  $\hat{y}$ , reduces relatively steady as the radii,  $R$ , increase.\*



In relation to  $R$ , the plots of the variables angle, velocity, and push show that push, velocity, and push increase slightly. By telescoping in to get more accuracy, the value of the pole angle is found to be between 0.08 and 0.13 radians; angular velocity is between 1 and 1.28 radians per second; and push force is between 5.55 and 7.8 newtons. These ranges provide a narrower range within which to calibrate the three scales.



Standardized angle, velocity, push

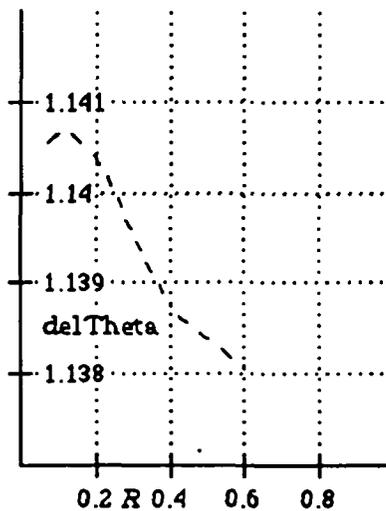
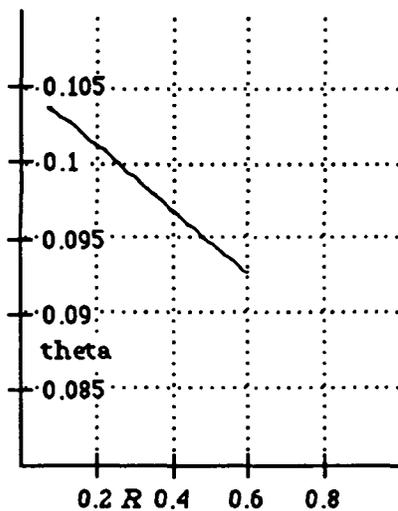
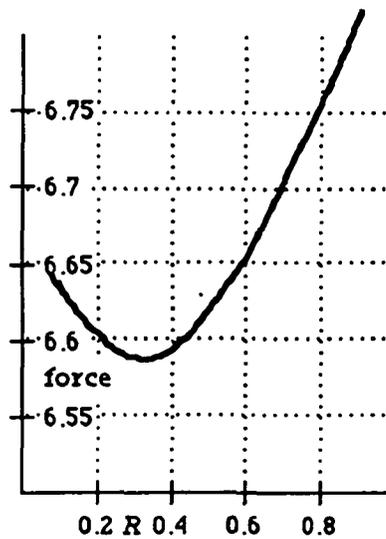
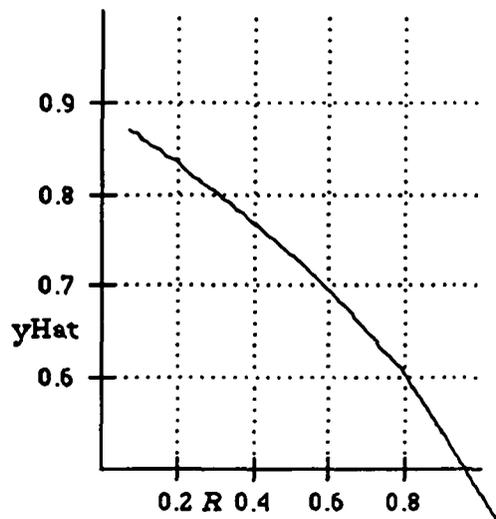


Angle, velocity, push variables

The controller experiments were performed again with the variables limited to these narrower limits. The results of the repeat experiment suggest the controller is able to balance the cart-pole system; the final position of the cart in

\*In fact the plot shows  $\hat{y}$  becoming negative, which is impossible for the true response value, since only absolute values are considered. But this anomaly is a result of the approximate nature of the fit of the quadratic form, and is not critical.

half of the trials was less than 0.8 m from its starting position and always between 0.29 and 1.42 m. Below we show plots of each key variable relative to the radii R.



By applying a similar analysis to alternative criteria, a fuller assessment of the controller performance can be had. Using plots similar to those for cart position, the alternative criteria's optima can be viewed in relation to the analysis demonstrated here.

## References

1. Michio Sugeno and Takahiro Yasukawa, "Linguistic modeling based on numerical data," in the Proceedings of the 4th IFSA Congress, in Brussels, 1991, pp. 264-267.
2. Z. Cao and A. Kandel, "Applicability of some fuzzy implication operators," *Fuzzy Sets and Systems*, **31**, 2 (1989), pp 151-186.
3. Roger Martin-Clouaire, "Semantics and computation of the generalized modus ponens: the long paper," *International Journal of Approximate Reasoning*, **3**, 2 (1989), pp 195-218.
4. Bart Kosko, *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs, 1992.
5. James M. Keller and Hossein Tahani, "Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks," in the *International Journal of Approximate Reasoning*, **6**, 2, pp. 221-240, 1992.
6. Isao Hayashi, Hiroyoshi Nomura, Hisayo Yamasaki, and Noboru Wakami, "Construction of fuzzy inferences rules by neural network driven fuzzy reasoning and neural network driven fuzzy reasoning with learning functions," in the *International Journal of Approximate Reasoning*, **6**, 2, pp. 221-240, 1992.
7. Hamid R. Berenji, "A reinforcement learning-based architecture for fuzzy logic control," in the *International Journal of Approximate Reasoning*, **6**, 2, pp. 267-292, 1992.
8. G. E. P. Box and K. B. Wilson, "On the experimental attainment of optimum conditions," in the *Journal of the Royal Statistical Society*, **B13**, 1, 1951.
9. Raymond H. Myers, *Response Surface Methodology*, Allyn and Bacon, Inc., Boston, 1971.
10. Raymond H. Myers, *Response Surface Methodology*, Allyn and Bacon, Inc., Boston, 1971.
11. A. H. Hoerl, "Optimum solution of many variables equations," in *Chemical Engineering Progr.* **55**, (11), 69 (1959).
12. N. R. Draper, "Ridge analysis of response surfaces," *Technometrics* **5** (4), 469 (1963).

538-67  
161838

N 93-29554

P-10  
**THE COGNITIVE BASES FOR THE DESIGN OF A NEW CLASS OF FUZZY LOGIC CONTROLLERS: THE CLEARNESS TRANSFORMATION FUZZY LOGIC CONTROLLER\***

**Labib Sultan**

Director of R&D  
Mentalogic Systems Inc., and  
Associate Professor  
Department of Computer Science  
Glendon Campus, York University  
Toronto, Ontario M4N 3M6, Canada

**Talib Janabi**

Director of Technology  
Mentalogic Systems Inc.  
7500 Woodbine Ave.  
Markham, Ontario L3R 1A8,  
Canada

**ABSTRACT**

This paper analyses the internal operation of fuzzy logic controllers as referenced to the human cognitive tasks of control and decision making. Two goals are targeted. The first goal focuses on the cognitive interpretation of the mechanisms employed in the current design of fuzzy logic controllers. This analysis helps to create a ground to explore the potential of enhancing the functional intelligence of fuzzy controllers. The second goal is to outline the features of a new class of fuzzy controllers, the Clearness Transformation Fuzzy Logic Controller (CT-FLC), whereby some new concepts are advanced to qualify fuzzy controllers as "cognitive devices" rather than "expert system devices". The operation of the CT-FLC, as a fuzzy pattern processing controller, is explored, simulated and evaluated.

**1. INTRODUCTION**

Methodologically, fuzzy logic controllers implement digital control method which simulates the human thinking in handling the imprecision inherent in the control of physical systems. They can be classified as control expert systems capable of interpreting fuzzy statements of human knowledge such as "Temperature is high" or "Increase flow slightly", etc. Fuzzy controllers employ the approximate reasoning procedure called the compositional rule of inference (CRI), introduced by Zadeh [8], which represents the core of the deduction mechanism of the controller. Following the CRI scheme, the control actions are deduced by the composition of the fuzzy sets which are generated from the measured values of process variables (the input to the controller), and the matrices of fuzzy rules (knowledge on the input-output relationship) using the relational algebra operations of Max and Min. Fuzzy logic controllers propagate numerical data of the process variables into fuzzy linguistic terms (this phase is called fuzzification), deduce the control actions as fuzzy sets using the CRI, and translate fuzzy actions into crisp data (this phase is called defuzzification) to be applied to the controlled process to keep it within the desired limits. Hence, the overall operation of the controller can be looked upon as a numerical to numerical mapping mechanism whereby compositional relations of fuzzy sets and fuzzy rules are handled by the compositional rule of inference while the controller is provided with two converters: numerical to linguistic (fuzzifier) and linguistic to numerical (defuzzifier) to facilitate its communication with real world processes.

---

\* This work is supported by Mentalogic Systems Inc. and the National Science and Engineering Research Council of Canada (NSERC).

In this paper the operation of fuzzy logic controllers is analysed within a cognitive framework based on two concepts. The first uses the Rasmussen model of the cognitive task analysis of control and decision making in a supervisory control environment [1, 4]. The second uses the concept of a fuzzy pattern and the measure of its clearness degree to describe the tasks of the fuzzy controller. These two concepts have been used in developing a new class of fuzzy logic controllers called the CT-FLC, or the Clearness Transformation Fuzzy logic Controller [5]. The CT-FLC is characterized as fuzzy patterns assessment and processing device. The paper discusses theoretical issues of the CT-FLC, and presents some simulation results on its performance.

## **2. THE FRAMEWORK OF THE COGNITIVE TASKS ANALYSIS OF FUZZY LOGIC CONTROLLERS**

Fuzzy controllers can be looked upon as cognitive devices which comply in their operation with the cognitive tasks achieved by skilled operators involved in decision making in a supervisory control environment. As such, we will follow the step-layered model of the control and decision making of Rasmussen [4] and Cacciabue [1] to establish and describe the tasks performed by fuzzy logic controllers. Following the step-ladder diagram, the operator behaviour in a supervisory control environment is described in terms of the cognitive tasks to be performed at three ladders: skill-based, rule-based and knowledge-based, depending on the complexity of the task to be handled by operators. Within this framework fuzzy logic controllers cover the skill-based and most of the rule-based decision-making functions of skilled operators. The knowledge based behaviour, where decisions are elaborated as a compromise between purposive policies such as safety and production policies, etc., falls beyond the task of the fuzzy controller as a parameter driven system of control.

The cognitive tasks achieved by the operator in handling the rule-based functions are:

- observation, detection and perception of process situations and status.
- assessment and evaluation of the current process situation.
- actions planning.
- actions execution.

Following the Rasmussen task analysis ladder diagram, it is obvious that the first and the last tasks correspond to the fuzzification and defuzzification tasks of the fuzzy controller, respectively, while the second and third tasks are related to the approximate reasoning procedure employed by the controller.

Further, we will intensively use the concept of fuzzy patterns to elaborate the definition of the tasks of the fuzzy controller. The rationale behind using of fuzzy pattern instead of its synonym fuzzy set is that patterns are the basic cognitive entities manipulated by humans in the decision making practice. The fuzzification task of the fuzzy controller corresponds to the perception phase of the human cognition whereby the observed numerical values of the process variables (such as, for example, the value of the temperature = 30° c) is mapped into fuzzy patterns such as NORMAL, SLIGHTLY HIGH, etc. The next task of the controller is to generate action(s) to react to the observed situation to recover the process to its normal/ desired operation. This phase is performed by the operator by activating an associative referencing to his/her long term memory to consult and select the proper action(s). This task is conveniently called " the associative pattern matching " activity, whereby the pattern(s) generated by the fuzzification phase are used to activate patterns of the control action(s). The translation of these patterns to numerical values to be applied to the system will be the task of the defuzzification. Hence, the three tasks : fuzzification, pattern matching and defuzzification are the major tasks performed by the fuzzy controller. These are the same tasks performed by operators in their usual practice in the supervisory control environment. They are consistent with Rasmussen cognitive task analysis also.

However, the approximate reasoning task of the fuzzy logic controller has a different meaning from the cognitive approximation achieved by skilled operators in the implementation of their decision making policy. The CRI scheme currently applied in fuzzy controllers can be given the following interpretation. The overall output of the controller is quantified as averaging of all the possible control actions deduced by firing all the fuzzy rules. The deductions are performed by Max and Min quantifiers to produce the action of each rule. The final action is generated by the defuzzifier as averaging all the actions to the process. Obviously, the human approximate reasoning is not limited, if at all applied, to this context. It is not necessary for the operator will be using all of his knowledge (fuzzy rules) to deal with each process situation. Rather, operators might activate the knowledge which is most relevant to the current context of a process situation. One of the schemes which has been developed recently and making use of this fact is called the clearness transformation mechanism for approximate reasoning [6, 7]. By this mechanism it is supposed that the human performs an assessment of the clearness degree of the perceived fuzzy patterns and activates the relevant rules on how to react, rather than calling all the rules (knowledge) about the process. He/she then qualifies and quantifies actions to be taken based on his/her assessment of the the detected patterns. The clearer the detected pattern of the process state are the more confident and relevant actions will be taken by the operator to recover the process to its normal operation. The approximation taking place here has the following context: to which extent the detected patterns are clear enough for the operator to initiate certain actions and how this clearness will affect the extent to which these actions will be performed. This interpretation has been formalized as the clearness transformation mechanism for approximate reasoning applied in the design of a new class of fuzzy controllers called the Clearness Transformation Fuzzy Logic Controller (CT-FLC). The outlines of the cognitive tasks implemented by this controller is presented in Figure(1).

The following features characterize the cognitive approximation performed by the controller:

1. The decision maker uses his/her long term memory to deduce the pattern of the required action (through the pattern matching activity) while applying an approximate reasoning mechanism to assess the clearness degree of the deduced fuzzy pattern of the control action.
2. The clearer the patterns of the process situations are the clearer the action patterns are and the more confident actions will be applied to the process. By this mechanism the "Strength" and "Weakness" measures of the detected patterns of process situations are mapped to affect the extent to which the fuzzy patterns of control actions will be applied to the controlled system.

The table below describes the cognitive tasks of the operators and the counterpart mechanisms employed by the CT-FLC.

THE OPERATOR COGNITIVE TASK	THE RELEVANT TASK OF THE CT-FLC
Detect and assess patterns of process variables and the current process context	Fuzzify the measured values of process variables into fuzzy patterns and determine the clearness of each pattern
Select most relevant set of actions to recover the process to its normal operation□	Pattern matching the fuzzy patterns with the rules to deduce the patterns of the control actions
Prioritize actions and assess the extent to which each action must be performed to achieve the goal	Approximate reasoning using the clearness Transformation mechanism of inference
Quantify the control action values and apply to the process	Defuzzification of the control action patterns into crisp control actions

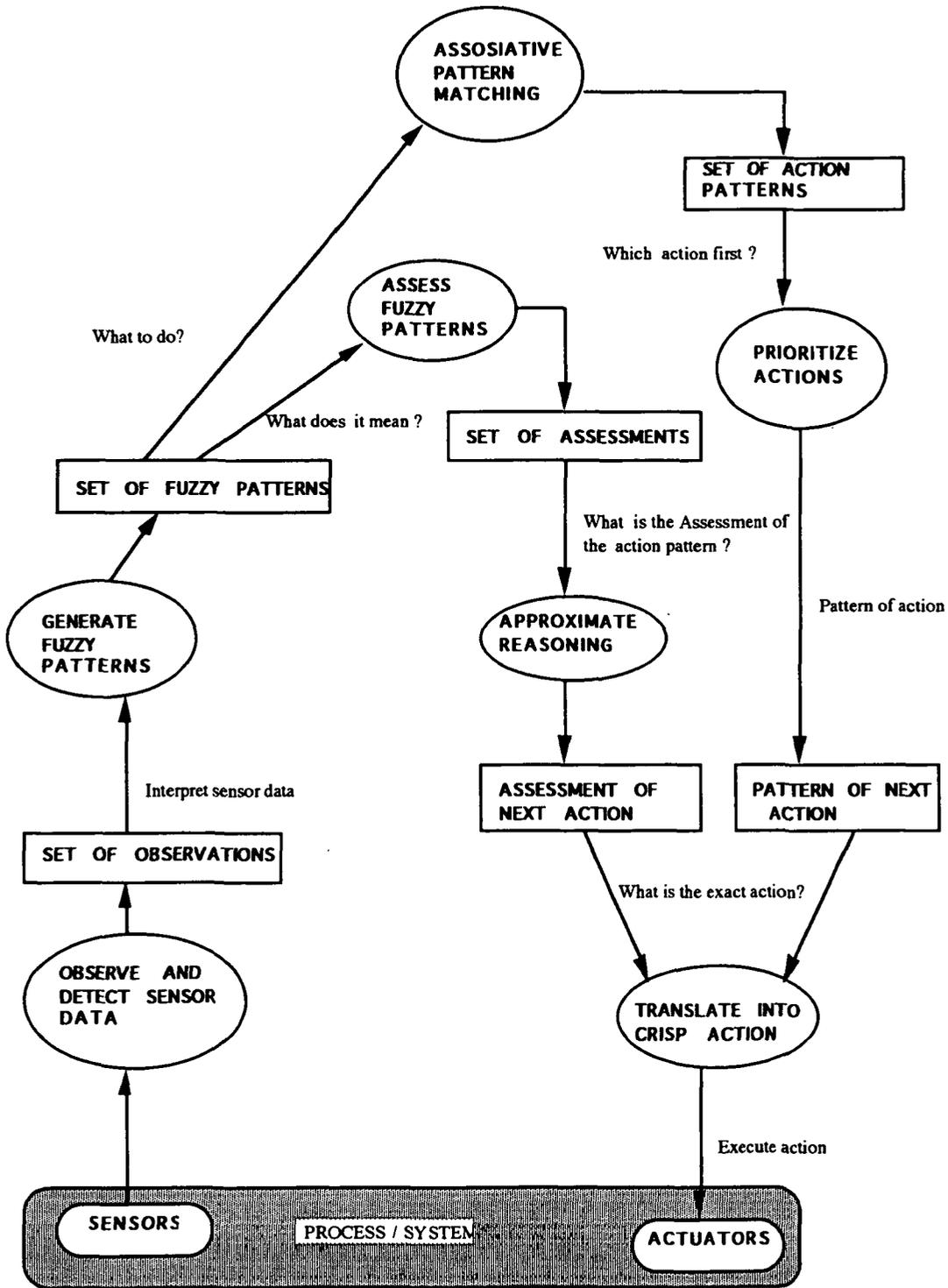


Figure 1. The Cognitive Model of CT-FLC Fuzzy Controller

### 3. THEORETICAL BASES OF THE CT-FLC

We bear in mind that the CT-FLC is a system which operates and makes its decision at the level of fuzzy pattern processing. Hence, two fundamental theoretical concepts have been used in the development of the CT-FLC: the concept of a fuzzy pattern and the formulation of the clearness transformation mechanism for approximate reasoning.

A *fuzzy pattern (FP)* is defined by a triple  $\langle S, D, A \rangle$ , where:

- S - is the syntactical description of a fuzzy pattern;
- D - is the domain to which a fuzzy pattern is attached; and
- A - is the clearness assessment of a fuzzy pattern.

We proceed with formal definition of each of these components.

**S - component** characterizes the syntactical description of the fuzzy pattern. We have utilized the logic of fuzzy predicates to describe the fuzzy patterns of the real world situations. In this context, the notion of a fuzzy predicate as an atomic formula of this logic is considered as an elementary fuzzy pattern. Other complex fuzzy patterns can be described as well-formed formulas (WFF) of this logic using the logical operators AND, OR, etc. The syntax of a fuzzy predicate (elementary fuzzy pattern), denoted as  $P_A, P_B$ , etc., is as follows:

$$P_A : L_x \text{ is } A$$

where,  $L_x$ - is a linguistic variables of Zadeh [8] and A- is its attribute value defined as fuzzy subsets of the universe of discourse X . As an examples of elementary fuzzy patterns is:

$$P_A : \frac{\text{THE STATE OF TEMPERATURE}}{L_x} \text{ is } \frac{\text{HIGH}}{A}$$

**D-component.** The domain of a fuzzy pattern  $P_A$  , denoted as  $D_{A,X}$ , is composed of three attributes  $\langle L_x, X, \sigma_x \rangle$ , where:

- $L_x$ - is the domain variable;
- X - is the space of all the instant models and objects  $(x_1, x_2, \dots)$  that can be substituted as values for  $L_x$ ;
- $\sigma_x$  - is the set of substitutions of the form  $\{x_i/L_x\}$  which define the allowed substitutions  $x_i$  for  $L_x$  from X.

As example of the domain of  $P_A$  :

$$D_{A,X} = \left[ \begin{array}{l} L_x = \text{THE STATE OF TEMPERATURE ;} \\ X = [ 0 , 50 ] \\ \sigma_x = \{ 20 ; 30 ; 35 ; 45 ; 50 \} \end{array} \right]$$

Figure (2) illustrates the definition of the domain of the fuzzy predicate  $P_A$ .

The next component is the *assessment of the clearness measure* of a fuzzy pattern by employing the clearness measures built in the closed interval  $[0,1]$  divided into a finite number of truth values  $\{ a_k \}$ . The "clearness" of a fuzzy pattern, is assessed when the variable (e.g.  $L_x$ ) of a fuzzy predicate (such as  $P_A$ ) is substituted by instancial models (such as  $x_j$  of the variable  $L_x$ )

from the domain  $D_{A,X}$ . Two measures,  $T$  and  $\Gamma$  are developed to estimate the clearness of fuzzy patterns. The local clearness  $T(P_A)$  and the global clearness  $\Gamma(P_A)$  of the fuzzy patterns. Figure 2 illustrates the concept of these two measures for the assessment of fuzzy patterns.

The local clearness measure is used to assess the clearness of a pattern at given domain variable values and formulated as:

$$T: P_A \rightarrow [0, 1] \mid \text{for } Lx_i = x_i$$

The global clearness measure, denoted as  $\Gamma$ , is used to assess the "global" clearness of a fuzzy pattern and formulated as:

$$\Gamma(P_A) = \{T_1(P_A), T_2(P_A), \dots, T_n(P_A)\} \mid \text{for all the substitutions } \{x_i/Lx_i\}.$$

In the CT-FLC system all the three components  $\langle S, D, A \rangle$  are represented in three knowledge blocks of the controller. The fourth knowledge block is used to represent the fuzzy rules (the control protocol). The control protocol of the fuzzy controllers is composed of a finite set of fuzzy rules of the form:

*IF < Fuzzy Pattern of Process Situation > THEN < Fuzzy pattern of Control Actions >*

Both the patterns of the "Process Situations" and the patterns of the "Control Actions" are specified as complex fuzzy patterns. A general form of a situation-Action rule of the control protocol is as follows:

*IF  $PA_1$  and  $PA_2$  ... and  $PA_n$  THEN  $PB_j$*

where:  $PA_i, PB_j$ - are elementary fuzzy patterns of the rules.

The next basic theoretical concept used in the development of CT-FLC is the approximate reasoning mechanism of the Clearness Transformation Mechanism of Inference(CTMI). Fuzzy patterns can be classified as "dynamic" or "static" to denote the patterns detected in real dynamic operation ( the output of the human perception) and the patterns represented in the controller knowledge base (the patterns established in the human long-term memory), respectively. The static and dynamic patterns have the same syntactical description but may differ in their clearness evaluation in terms of the "strength" and "weakness", as it is defined in the following:

If  $G'$  is a dynamic pattern of  $G$ , then we say that the pattern  $G'$  is "clearer" or "stronger" than  $G$  if  $\Gamma(G') > \Gamma(G)$ , and  $G'$  is "less clear" or "weaker" than  $G$  if  $\Gamma(G') < \Gamma(G)$ , for the same instant models of its domain, where  $\Gamma$  is the clearness measure of a fuzzy pattern. The CTMI has been developed and established theoretically and in experimental studies on the analysis of approximate reasoning of the Transformation Mechanism. It is a Modus Ponens based rule of inference which uses the  $T$  and  $\Gamma$  measures to generate an estimation of the local clearness degree of fuzzy patterns of the control actions [ 6,7]. Some two mechanisms are involved in the CTMI: the Pattern Matching and the Transformation Mechanism.

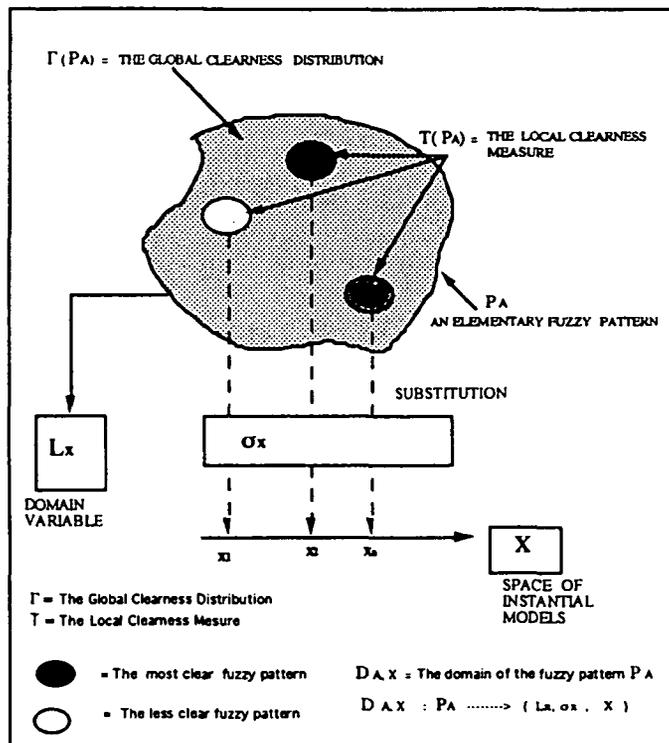


Figure 2. The Clearness and Domain Interpretation of a Fuzzy Pattern

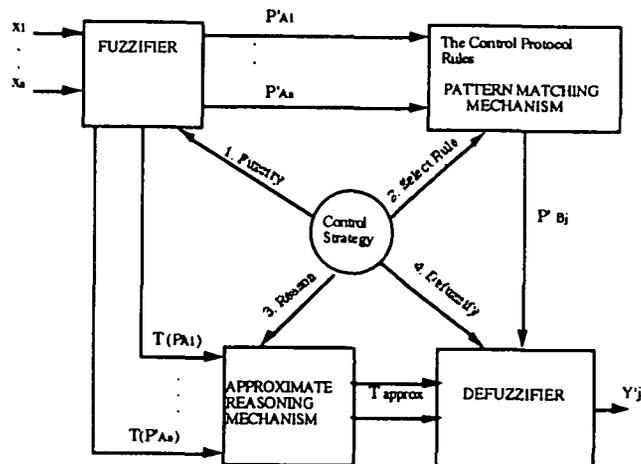


Figure 3. The basic modules and operation phases of CT-FLC

#### 4. THE CONCEPTUAL DESIGN AND OPERATIONAL PHASES OF THE CT-FLC

The CT-FLC is designed following the cognitive model of fuzzy control described above. It has a modular architecture consisting of four operational modules: The Fuzzifier, The Controller pattern matching mechanism, The Approximate Reasoning Mechanism and the Defuzzifier. The flow of data and control between these four modules is coordinated by the Control and Inference module. The controller operates in four phases labeled in Figure 3 as: the Fuzzification Phase, the Rule Selection and Inference Phase, the Approximate Reasoning Phase and the Defuzzification Phase.

The abbreviation on the block diagram of the controller are:

$P'_{A1}, \dots, P'_{An}$  - fuzzy patterns of the process input variables ( $X_1, \dots, X_n$ ),

$T(P'_{A1}), \dots, T(P'_{An})$  - the local clearness of fuzzy patterns  $P_{A1}, \dots, P_{An}$ .

$P'_{Bj}$  - the deduced fuzzy patterns of the control action for the output variables ( $Y_j$ )

Tapprox- the local clearness of the fuzzy patterns of the control actions  $P'_{Bj}$ .

#### 5. APPLICATION EXAMPLE

This is a simulation example to illustrate the performance of the CTFLC. The system in this example is a closed loop single-input single-output system consisting of two parts, a linear element and a nonlinear element. The linear element is a second order system with a transfer function

$$G(S) = \frac{1}{s^2 + 0.2s + 0.1}$$

and the nonlinear element is a dead-zone equal to 0.3 with a slope of 1.0 as shown in figure (4). Two variables are selected to represent the process. These are the error in the output response and the change of this error. The control rules used in the fuzzy controller are shown in figure (5). The fuzzy patterns implemented in the controller knowledge-base are: positive high, positive-normal big, positive-normal small, positive low and similar patterns for the negative estimation of the error patterns. The global clearnesses of these patterns, as well as those of the patterns of the control actions were embodied in the Fuzzifier and Defuzzifier knowledge-bases of the controller (figure 6).

The digital simulation response for a unit step input before and after the fuzzy controller in the loop is illustrated in Figure 7. It is evident that the controlled system has a smooth response with no steady state error. The elimination of the steady-state error despite the presence of the dead-zone nonlinearity in this system is a remarkable achievement of this controller. It illustrates the capacity of the TTFC and reflects the effectiveness of the design approach of this generation of controllers.

#### 6. EVALUATION

1. A new class of fuzzy controllers : *The Clearness Transformation Fuzzy Logic Controller* is developed. This controller is designed based on a cognitive model of control. It is capable of performing the tasks of approximate reasoning at the level of fuzzy patterns. It incorporates knowledge for fuzzy pattern clearness assessment and utilizes approximate reasoning mechanism based on the Clearness Transformation Mechanism of Inference .
2. The fuzzy controller has been simulated and analysed through applications with difficult control problems. The results were extremely satisfactory in terms of performance and robustness when compared with the existing designs of fuzzy logic controllers.

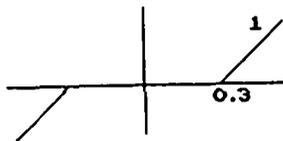
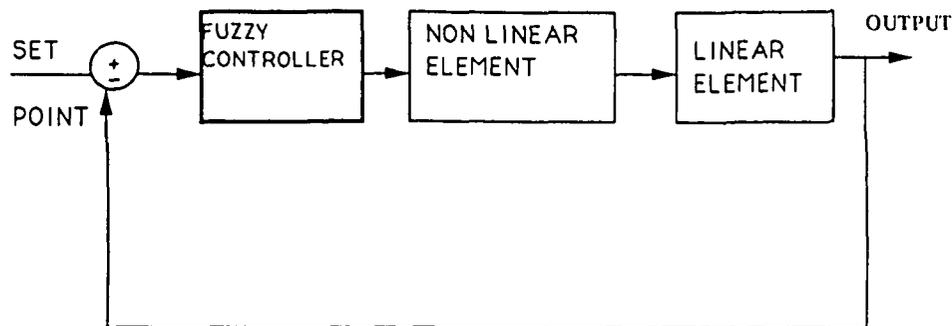


Figure 4. Block Diagram and Dead-zone Nonlinearity

IF	THEN IF	THEN
E = NH	CE = NH or CE = NNB or CE = NNS or CE = NL	CA = NNBR
E = NH	CE = PL or CE = PNS or CE = PNB or CE = PH	CA = NL
E = NNB	CE = NH or CE = NNB or CE = NNS	CA = NNBR
E = NNB	CE = NL	CA = NNSR
E = NNB	CE = PL or CE = PNS or CE = PNB or CE = PH	CA = NL
E = NNS	CE = NH or CE = NNB or CE = NNS or CE = NL	CA = NH
E = NNS	CE = PL or CE = PNS	CA = NNSR
E = NNS	CE = PNB or CE = PH	CA = NNBR
E = NL	CE = NH or CE = NNB or CE = NNS or CE = PL	CA = NL
E = NL	CE = NL	CA = NH
E = NL	CE = PNS or CE = PNB or CE = PH	CA = NNBR

E = PL	CE = NH	CA = NL
E = PL	CE = NNB or CE = NNS	CA = PL
E = PL	CE = NL	CA = PNSL
E = PL	CE = PL or CE = PNS or CE = PNB or CE = PH	CA = PH
E = PNS	CE = NH or CE = NNB or CE = NNS or CE = NL	CA = PL
E = PNS	CE = PL or CE = PNS or CE = PNB or CE = PH	CA = PH
E = PNB	CE = NH or CE = NNB or CE = NNS	CA = PL
E = PNB	CE = NL	CA = PNSL
E = PNB	CE = PL or CE = PNS or CE = PNB or CE = PH	CA = PH
E = PH	CE = NH or CE = NNB or CE = NNS	CA = PL
E = PH	CE = NL or CE = PL or CE = PNS or CE = PNB or CE = PH	CA = PH

Figure 5. Control Rules

The abbreviations used are:  
 E = Error  
 CE = Change in Error  
 CA = Control Action  
 NH = Negative High  
 NNB = Negative Normal Big  
 NNBR = Negative Normal Big Right (right side of the curve)  
 NNS = Negative Normal Small  
 NNSR = Negative Normal Small Right (right side of the curve)  
 NL = Negative Low  
 PH = Positive High  
 PNB = Positive Normal Big  
 PNSL = Positive Normal Big Left (left side of the curve)  
 PNS = Positive Normal Small  
 PNSL = Positive Normal Small Left (left side of the curve)  
 PL = Positive Small

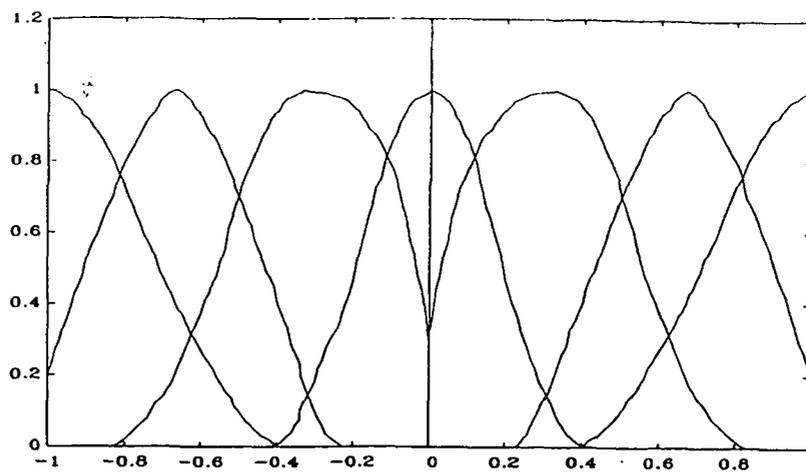


Figure 6. Clearness set

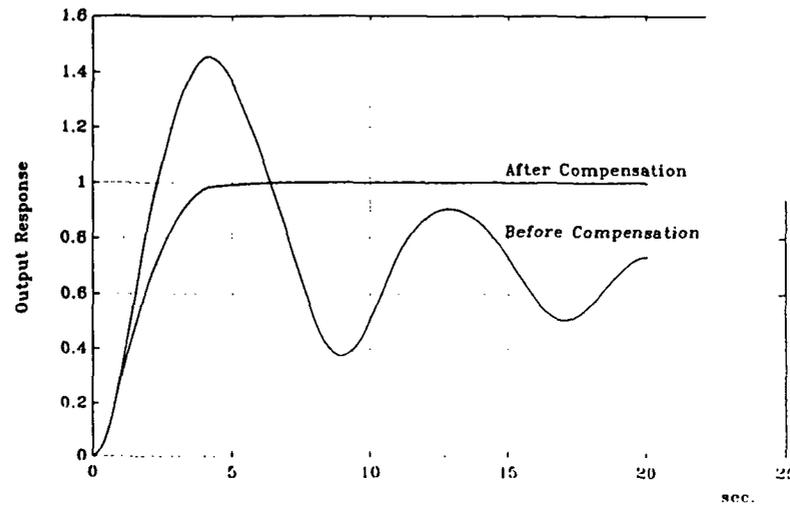


Figure 7. Unit Response for Closed Loop

## REFERENCES

1. P.C. Cacciabue, G. Mancini and G. Guida , A Knowledge Based Approach to Modelling the Mental processes of Operators. Proc. IEAEA/CEC/OECD/NEA Int. Conf. on Man-Machine Interface, Tokyo, Japan, , pp.71-78, 1988.
2. E.H. Mamdani, Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis, IEEE Trans. Computers, C - 26, no.12, pp. 1182-1191, (1977).
3. M. Mizumoto, " Fuzzy Control Under Various Approximate Reasoning Methods", Second IFSA Congress, pp. 143-146, (1987).
4. J.Rasmussen, Information processing and Human-Machine Interaction. An Approach to Cognitive Engineering. North- Holland, (1986).
5. M. Sugeno, The Industrial Applications of Fuzzy Controllers, Elsevier Science Publishers B.V., (1985).
6. L. Sulatn and T. Janabi , " The Truth Transformation Fuzzy Logic Controller: Outline of the design of a new generation of fuzzy logic controllers", Proc. of AI-91- Frontiers in Innovative Computing for the Nuclear Industry", Jakson, Wyoming, pp. 156-164, 1991.
7. L.Sultan and T. Janabi, " The Organization and Architectural Design of Knowledge-Based Fuzzy Controllers". Submitted to IEEE Trans. on System, Man and Cybernetics, 1991.
8. L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," IEEE Trans Systems, Man and Cyber., vol. SMC-3, pp.28-44, 1973.

539-67  
161839

N 93-29555

0-10

## A FUZZY CONTROL DESIGN CASE: THE FUZZY PLL

H.N. Teodorescu, I. Bogdan  
*Center for Research in Fuzzy Systems & Artificial Intelligence  
Polytechnic Institute of Iași, Romania*

### ABSTRACT

The aim of this paper is to present a typical fuzzy control design case. The analyzed controlled systems are the phase-locked loops -- classic systems realized in both analogic and digital technology. The crisp PLL devices are well known.

### Introduction

To evidence the requirements of the analyzed case, in this first part of the paper, a review of the PLL systems and their applications is made.

The phase-locked loops (PLL) are devices that perform the phase control of an oscillator (see Figure 1). As any crisp control can be turned into a fuzzy control, the idea of the fuzzy-controlled PLL (FPLL) [2], [3], [4] is natural. Of course, one has to analyze if such a control is beneficial or not. This last problem is only partly analyzed here, more details being given in papers [2], [3], [4], to which the reader is referred.

The PLL concept dates to the early days of radio technology.

Phase-Locked Loops (PLLs) devices are systems primarily aimed to generate signals in phase with the input (control) signal phase, while the input signal is (slowly) changing. If the input signal is noisy, the output signal should follow the carrier (basic signal) phase. Thus, the PLL can act as a nonlinear bandpass filter tuned by the incoming signal. In fact, the PLL recreates the original signal rather than to just filter the input signal.

The PLL basically consists in two circuits: a

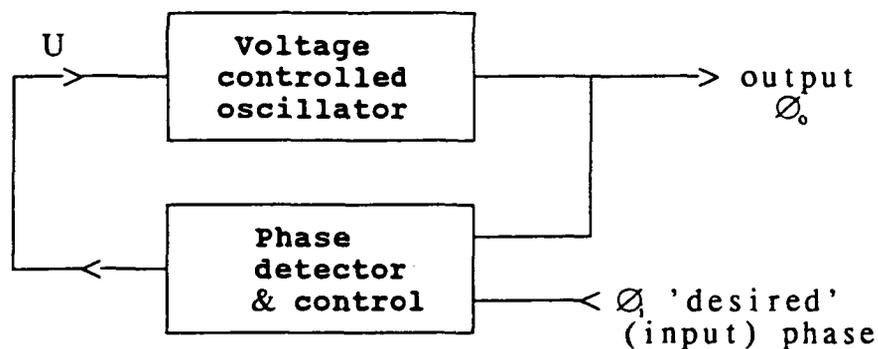


Figure 1: Basic PLL device

controlled signal generator (voltage controlled oscillator - VCO) and a phase detector & control circuit (PD-C). As the control signal is an estimate of the phase (or a function of it), the PLL can be used in demodulation purposes (frequency or phase demodulation). The PLL can also be used in amplitude demodulation, as it generates a constant level output signal, as required by the amplitude demodulators. Moreover, PLLs are used in frequency synthesizers. In this application, a fixed precise generator provides the input signal and the control loop includes a frequency divider to allow for frequency changes. Industrial applications such as motor-speed control were also announced [7]. Other applications include signal synthesis [8].

In many such applications, the dynamical characteristics of the PLL play an important part, mainly the acquisition time and the noise immunity. The time needed to reach the quasi-stationary regime, for a given hop in frequency/phase is most usually determined in terms of equivalent number of periods. This characteristics is important in frequency demodulators and in fast switching frequency synthesizers that must often change the output frequency. (Such devices are used for example in frequency hopping system). Noise output spurious signal suppression power versus noise input power is important in (tele)communications applications such as carrier recovery [9].

In the last two decades PLLs turned from the analog technology to the digital one, due to some important advantages: high frequency range (up to 30 MHz in monolithic integrated circuits), insensitivity to changes in temperature and power-supply voltage, programmable bandwidth and center frequencies.

Moreover, in the digital technology, very high quality factors (i.e. narrow - bandwidth) loops can be achieved, and high order loops are easy to construct by simple cascading operation. Unlike the analog PLLs, where the error signal provided by the phase detector (PD) corrects the (analog) VCO frequency, in usual, digital PLLs the error signal controls the direction of on up - down counter.

Much used are devices from the class of integrated (monolithic) hybrid PLLs. These devices include an analog VCO and low pass filter (LPF), and a digital PD and digital dividers.

Such devices are usually manufactured in CMOS (Complementary-symmetry Metal-Oxide-Semiconductor) or TTL (Transistor-Transistor-Logic) technology and a classical example is the 4046 circuits. (Such devices are often named "digital PLLs" although they are hybrid, while the true digital PLLs are named "all - digital PLLs").

### The classic PLL device

In the usual analogic PLLs, the phase control is got by a linear (P) control loop, i.e.

$$U = k*(\varnothing_o - \varnothing_i) \quad (1)$$

$$\Delta \varnothing_o = \gamma * U \quad (2)$$

where  $\Delta \varnothing_o$  is considered as the phase shift per second. (Indeed, the frequency change is controlled by U, rather than by the phase).

More exactly, in an analogic PLL, the relations are:

$$U = k * \langle (\varnothing_o - \varnothing_i) \rangle \quad (3)$$

$$\Delta \varnothing_o = \gamma * U \quad (4)$$

where  $\langle .. \rangle$  stands for the mean value, obtained by integration over a fixed time period. Thus, the control is of proportional-integral type (PI).

The difference  $\varnothing_o - \varnothing_i$  is performed by the block named 'phase detector'. The integration (average value) in eq. (3) is realized by a block named 'low-pass filter'. The complete block diagram of the basic PLL system is sketched in Figure 2.

### Turning the crisp control into a fuzzy control

Obviously, such a control as described by eqs. (1) and (2) can be performed in a quasi-linear, or in a nonlinear manner, by using a simple fuzzy control system followed by a defuzzifier block (Figure 3).

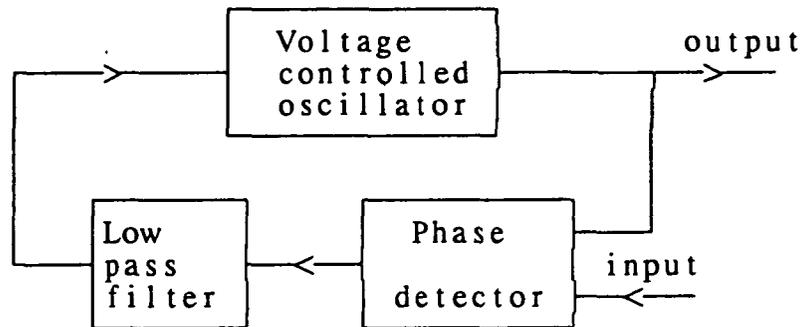


Figure 2: Basic diagram of the classic PLL

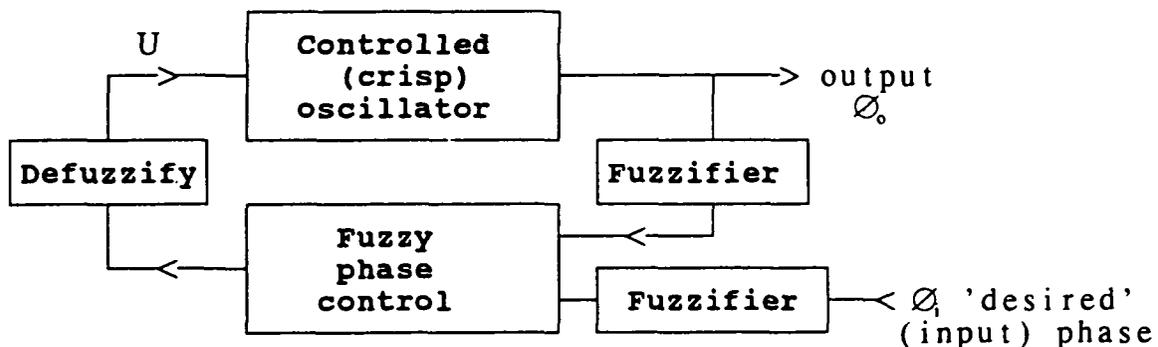


Figure 3: Basic fuzzy PLL device (FPLL)

The linguistic -- and, with appropriate definitions of the membership functions, the fuzzy -- control for a classic, linear PLL system can be described by such simple rules as below:

- |  |                                 |
|--|---------------------------------|
| <b>If <math>\varnothing_o - \varnothing_i</math> is Negative Big</b>   | <b>THEN U is Positive Big</b>   |
| <b>If <math>\varnothing_o - \varnothing_i</math> is Negative Small</b> | <b>THEN U is Positive Small</b> |
| <b>If <math>\varnothing_o - \varnothing_i</math> is Zero</b>           | <b>THEN U is Zero</b>           |
| <b>If <math>\varnothing_o - \varnothing_i</math> is Positive Small</b> | <b>THEN U is Negative Small</b> |
| <b>If <math>\varnothing_o - \varnothing_i</math> is Positive Big</b>   | <b>THEN U is Negative Big</b>   |

If the membership functions assigned to the above linguistic (input, and respectively output) degrees are equal, isosceles triangles, then the performed control is almost linear. If the triangles have unequal bases, given by a nonlinear law (e.g.  $B_i = \exp(a \cdot i)$ ), then the control is nonlinear, approximating the according law. For more details on the characteristic functions of defuzzified fuzzy systems, see [5] and the following chapters. Fuzzy control of the PLLs change them into intelligent devices: they behave much similar as if a human operator controls the phase locking process. This has some benefits and some costs. Nonlinear type fuzzy control can be beneficial in PLLs because it can improve the convergence rate of the phase-locking process, and also can improve the noise rejection performance [2], [3], [4]. On the other hand, using fuzzy control increases the complexity and cost of the systems and can lower the maximum operating frequency of the loop, due to the high amount of computation required by the fuzzy control.

A more complex control, taking into account both the phase and its variation (got by means of the difference between the actual and previous values of the phase) is increasing the loop performance. Such a control is illustrated in Figure 4.

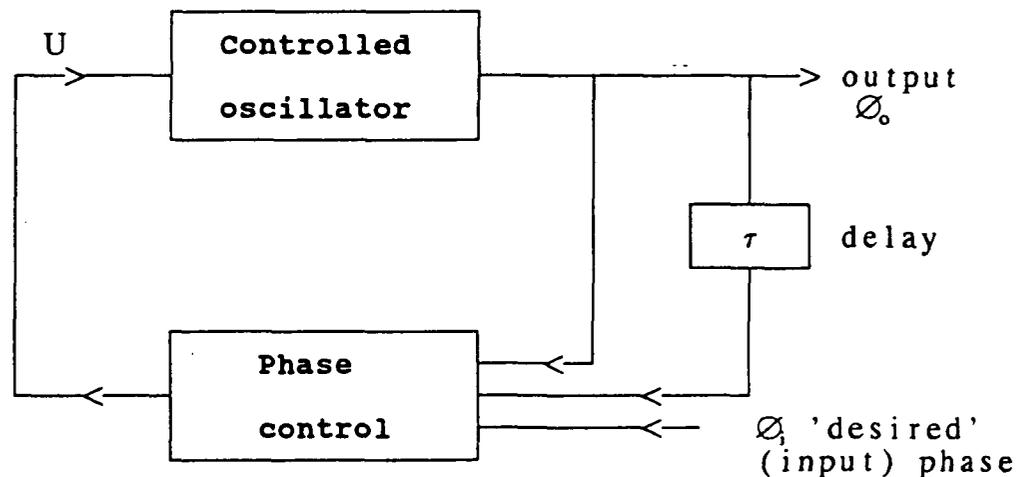


Figure 4: PLL device with double input control

An example of natural control rules for such a control is ( $\Delta\varnothing_n$  and  $\Delta\varnothing_{n-1}$  mean the differences  $\varnothing_n - \varnothing_{n-1}$  at the moments  $t_n$  and  $t_{n-1}$ , respectively):

- If  $\Delta\varnothing_n$  is Negative Big AND  $\Delta\varnothing_{n-1}$  is Negative Big  
THEN U is Positive Very Big**
- If  $\Delta\varnothing_n$  is Negative Big AND  $\Delta\varnothing_{n-1}$  is Negative Small  
THEN U is Positive Big**
- If  $\Delta\varnothing_n$  is Negative Big AND  $\Delta\varnothing_{n-1}$  is Zero  
THEN U is Positive Low**
- .....
- If  $\Delta\varnothing_n$  is Positive Low AND  $\varnothing_{n-1}$  is Positive Low  
THEN U is Negative Very Big**
- If  $\Delta\varnothing_n$  is Positive Big AND  $\varnothing_{n-1}$  is Positive Big  
THEN U is Negative Very Big**
- If  $\Delta\varnothing_n$  is Very Big and  $\Delta\varnothing_{n-1}$  is Very Big  
THEN U is Negative Very Very Big**

Even at the linguistic description level, the controlled system can behave in an unstable (e.g. oscillating) manner. The global, linguistic stability is very easy to check: the system is stable iff the state transition graph does not include any cyclic sub-graph.

### The all digital PLL fuzzy control schematic

Although analogic PLLs are largely used, for demanding applications, they are surpassed by the all-digital PLLs. In what follows, only digital PLL type will be addressed.

An all-digital PLL presented in [1] is claimed to have a good dynamic behavior and a very good rejection of the input phase noise because of the adaptive phase detector it contains. Its transfer characteristic (figure 5) is non-linear so that the phase detector output is zero for phase error absolute values greater than  $2\phi_R$ . Keeping  $\phi_R = \pi/20$  as long as the loop is locked, the PLL completely rejects the input phase noise greater than  $\pi/10$ , and strongly reduces the one smaller than this value. The phase detector adaptivity consists in changing  $\phi_R$  in accordance with the actual phase error value and maintaining the characteristic top corner abscissa close to it. The characteristic may, also, be translated along the vertical axis in order to cope with the phase detector input signals frequency difference.

The phase noise rejection reported in [1] was confirmed by our computer simulation of the all digital PLL, that yields a curve  $Z_{out} = f(Z_{in})$  very closed to that presented in [1]. The same computer simulation shows an about 25 iterations phase locking process for a 3 radian step in the input phase error (figure 6). The transient regime is considered to end when the input

phase error becomes smaller than 0.01 radian (about one tenth from the minimum value of the crisp PLL transfer characteristic turning point abscissa - figure 5).

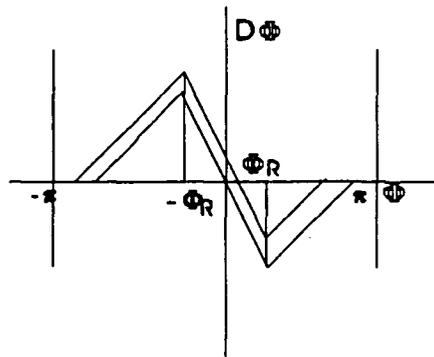


Figure 5: Crisp PD transfer characteristic

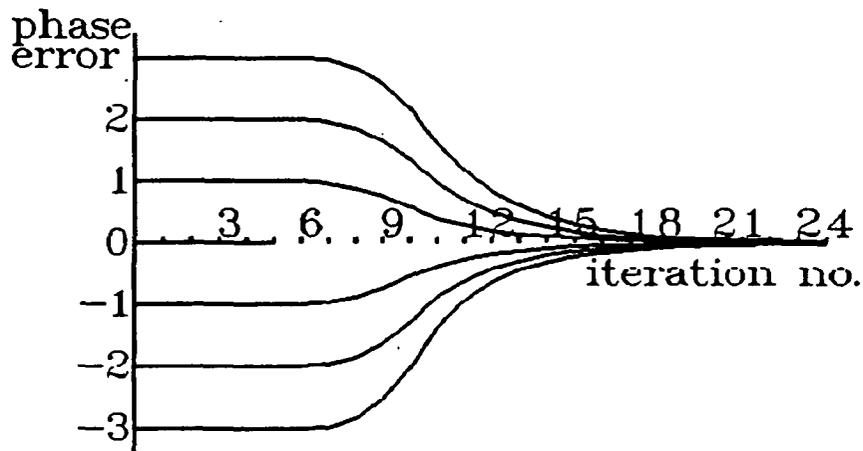


Figure 6: Crisp PLL phase locking transient response

This performant phase detector with non-linear adaptive characteristic is ideally suited to be replaced by a fuzzy control circuit, which is more flexible in design and operation, and may improve the PLL parameters. A possible way to introduce the fuzzy control (figure 7) is suggested in [4]. A fuzzifier circuit yields a 5 degree linguistic variable both for the actual and the previous phase error values. The fuzzy control circuit outputs the truth values for the 11 degrees of an linguistic variable by using inference rules of the above mentioned type:

IF  $\phi_{n-1}$  is NB AND  $\phi_n$  is NS THEN  $D\phi$  is NVB

The phase error is denoted as  $\phi$ , the output correction - as  $D\phi$ , and the linguistic variable degrees - as NVB (from Negative Very Big), NS (Negative Small) a. s. o. The all 25 rules used by the inference machine and presented in figure 8 are a "fuzzy model" for the phase detector operation in accordance with the authors' "feeling". A defuzzifier circuit produces a crisp correction value by means of the gravity center method.

The transfer characteristic of the phase control circuit from the actual phase error input to the crisp correction output is a rational fraction of 3 degree polynomials [3]. Its expression

[4] (eq. 5 for  $\phi_{n-1} = 0$ ) shows that the fuzzy control circuit has a strongly non-linear characteristic and its shape is easily controlled by means of the inference machine architecture. The actual shape induced by the inference rules from figure 8 is presented in figure 9.

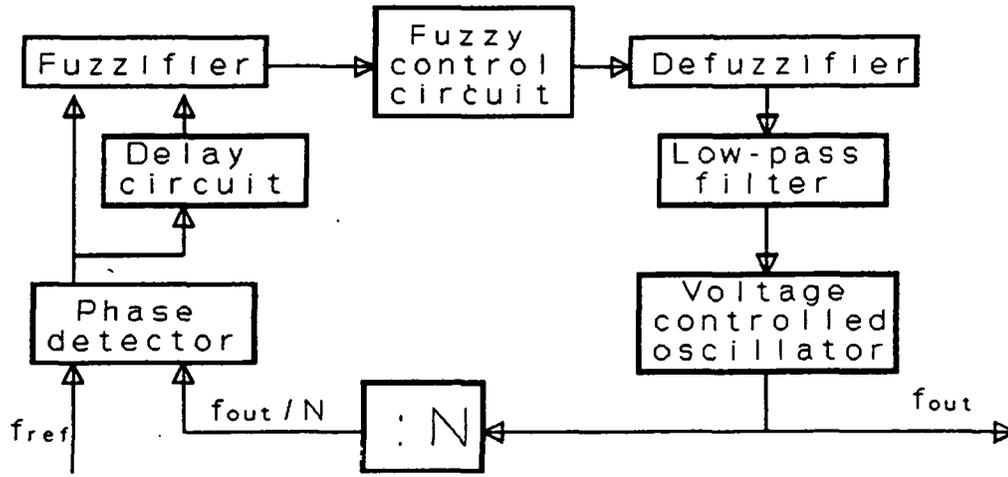


Figure 7: FPLL skeleton diagram

$$\phi_G = D\phi(0, \phi_n) = \begin{cases} \frac{\pi}{5} \frac{14\phi_n^2 + 23\pi\phi_n + 5\pi^2}{4\phi_n^2 + 6\pi\phi_n + \pi^2}, & -\pi \leq \phi_n \leq -\frac{\pi}{2} \\ \frac{12\pi}{5} \frac{\phi_n^2 + \pi\phi_n}{8\phi_n^2 + 4\pi\phi_n - \pi^2}, & -\frac{\pi}{2} \leq \phi_n \leq 0 \\ -\frac{12\pi}{5} \frac{\phi_n^2 - \pi\phi_n}{8\phi_n^2 - 4\pi\phi_n - \pi^2}, & 0 \leq \phi_n \leq \frac{\pi}{2} \\ -\frac{\pi}{5} \frac{14\phi_n^2 - 23\pi\phi_n + 5\pi^2}{4\phi_n^2 - 6\pi\phi_n + \pi^2}, & \frac{\pi}{2} \leq \phi_n \leq \pi \end{cases} \quad (5)$$

		phase error n-1				
		NB	NS	Z	PS	PB
phase error n	NB	PUB	PUB	PB	PM	PS
	NS	PUB	PB	PM	PS	PUS
	Z	PS	PUS	Z	NUS	NS
	PS	NUS	NS	NM	NB	NUB
	PB	NS	NM	NB	NUB	NUB

Figure 8: Inference rule set

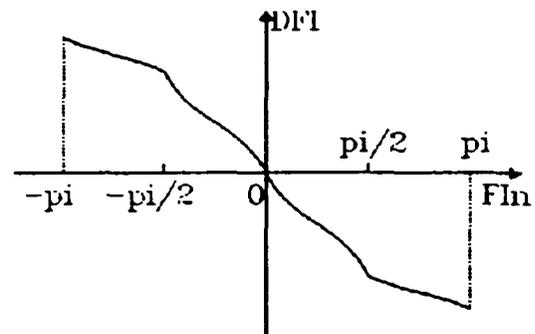


Figure 9: Fuzzy PD transfer characteristic

## 2. Fuzzy controlled PLL (FPLL) parameters

The FPLL dynamic behavior and noise properties are checked by means of the computer simulation. As figure 10 shows the FPLL needs only 10 iterations to get phase lock for the same step in input phase error, while maintaining the same great input phase noise rejection (figure 11 -  $Z_{out}$  and  $Z_{in}$  are the input and output phase noise effective values, respectively).

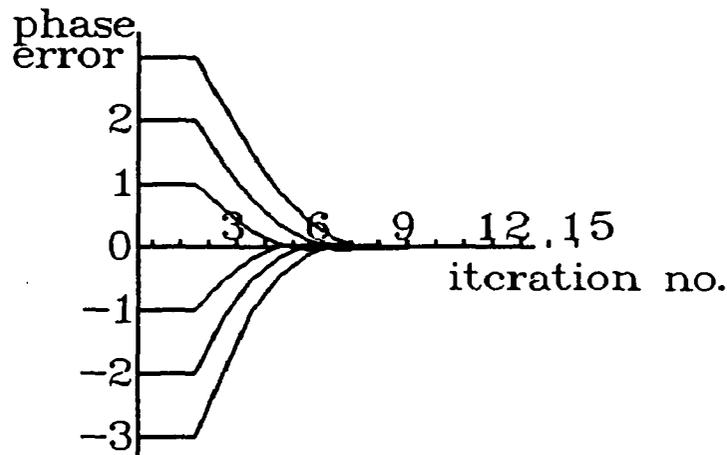


Figure 10: FPLL phase locking transient response

The dynamic behavior is further improved by changing the membership function shape. For a square root function, the number of the iterations till phase locking decreases to about 9. The same is the result of unequal base triangular membership functions.

The FPLL frequency acquisition regime is, also, greatly improved by the fuzzy control [4].

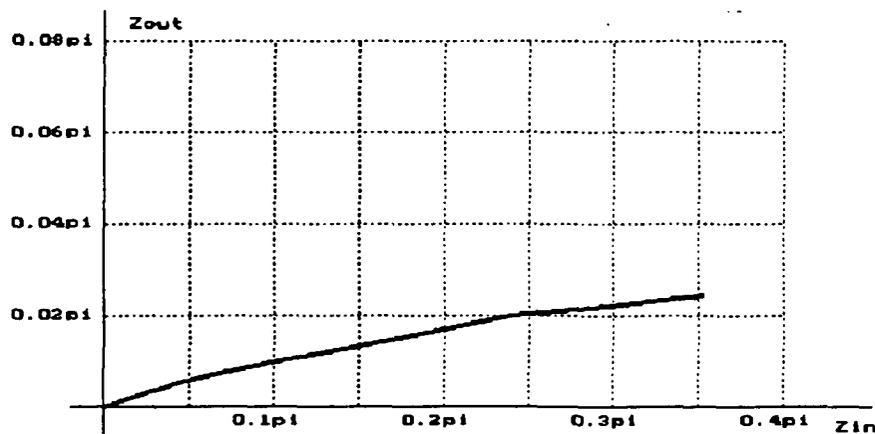


Figure 11: FPLL input phase noise rejection

### Some practical design hints: turning a crisp control into a fuzzy control

Fuzzy control of a crisp system asks for a fuzzifier block in front of the fuzzy control block, and of a defuzzifier block at the output. In other words, the overall control is a crisp control, and the fact that the way the control is performed is fuzzy is not seen by the controlled system. Supposing the defuzzification is realized by the center of gravity method, it is easy to determine the crisp input-to-output (characteristic) function of the equivalent crisp control.

Suppose now that the control characteristic function has to pass through a given number of fixed points in the input-to-output (xy) plane. (Only the problem of one-input, one-output control is discussed here, for sake of brevity). Let these points be:

$$\{(x_k, y_k) / k = 1, 2, \dots, n\}.$$

Also suppose that the type of membership functions is fixed, and all the membership functions  $x \sim_k$ ,  $y \sim_k$  are unimodal, and they attain the value 1 in just one point:

$$\mu_{x \sim_k}(u) = 1 \iff u = x_k; \mu_{y \sim_k}(v) = 1 \iff v = y_k.$$

For example, the membership functions can be triangular, sinusoidal, Gaussian a.s.o. Then, the control system is simply designed by using the following rules:

1. choose the membership functions width such as they overlap only two by two;
2. choose the membership functions vertices such as their coordinates are  $(x_k, y_k)$ ;
3. establish the rules describing the system in the form:

**If input is  $x \sim_k$ , Then output is  $y_k$ .**

Then, the defuzzified output will pass through the given points.

If a two-input system is to be designed being given the points:

$$\{(x_{1k}, x_{2k}; y_k) / k = 1, 2, \dots, n\},$$

the same procedure has to be observed.

Usually, the final step of your design must be the computer simulation, to check for the results.

## Conclusions

An analysis example of fuzzy control design problem was presented. The analysis was applied to the concepts of fuzzy controlled PLL.

The fuzzy control of classic analog PLLs is easy to design because the control system has to be a monotonic one. Then, the rules are derived in a very natural manner. The control can be easily changed, either by changing the rules, or the membership functions. The rules can be changed either by introducing new linguistic degrees, or by re-defining the input-to-output mapping of the linguistic degrees. Thus, this design case is most suitable in the classroom.

In the case of adaptive PLLs, the control is more intricate, and an adaptation of the control system configuration, rules and membership functions is needed.

It was shown by computer simulation that the suitably designed fuzzy control greatly improved the dynamic behavior of all digital adaptive PLL, while maintaining the input phase noise suppression properties of the original crisp PLL

## References

1. O. Nakajima, H. Hikawa, S. Mori -- Performance Improvement of an All Digital PLL with Adaptive Multilevel Quantized Phase Comparator. Proceedings ISCAS 1988, IEEE, pp 603 - 606.
2. I. Bogdan, H. N. Teodorescu, D. Galea, E. Sofron -- Analysis of Fuzzy Control Techniques. AMSE Int. 90 Conference on Signals and Systems, Cetinje, Yugoslavia.
3. H. N. Teodorescu, I. Bogdan, D. Galea, -- Fuzzy PLLs, 1990 International Symposium on Fuzzy Systems and Applications, Iasi, Romania.
4. I. Bogdan, H. N. Teodorescu, D. Galea, -- Further Analysis of the FPLL Dynamical Behaviour. RSFS Magazine, no. 3 - 4 / 1991, Iasi, Romania.
5. H.N. Teodorescu: Equivalence between Crisp and Fuzzy Systems with Defuzzified Output. In: E.P. Klement, H.N. Teodorescu: An Introduction to Fuzzy Systems
6. I. Bogdan, H. N. Teodorescu - Optimal Fuzzy Controlled Frequency Detector - AMSE '92 International Conference on Signal & Systems, Geneva, Switzerland (to be published)
7. H. Hikawa, S. Mori - DC Motor Control with Digital PLL Regulation - Proceedings of the 31<sup>st</sup> Midwest Symposium on Circuits & Systems, August 9-12, 1988, pp 809-812
8. R.P. Gilmore - Complex Waveform Generation Using DDS Techniques - Qualcomm, Application Note
9. J. Klapper, J.T. Frankle - Phase Lock and Frequency-Feedback Systems - Academic Press, 1972

OMIT TO  
END

AUTHOR INDEX

Aazhang, Behnaam .....	589	Laczko, J. ....	545
Barone, Joseph M. ....	205	Lara-Rosano, Felipe .....	304
Bezdek, James C. ....	98	Lea, Robert N. ....	154
Bogdan, I. ....	350		398
Bosc, Patrick .....	478		515
Buhusi, Catalin V. ....	360	Lee, Hon-Mun .....	125
Carreno, L. A. ....	627	Lee, Jonathan .....	185
Chang, Ching-Chuang .....	418	Leigh, Albert B. ....	69
Chen-Kuo Tsao, Eric .....	98	Lim, P. Y. W. ....	427
Chiang, Jung-Hsien .....	257	Littman, David .....	248
Chiu, Stephen .....	239	Ma, Yibing .....	49
Chung-Hoon Rhee, Frank .....	81	Madyastha, Raghavendra K. ....	589
Cohen, M. E. ....	535	Mansfield, W. H. ....	496
Copeland, Charles .....	154	McAllister, Luisa .....	555
Cutello, Vincenzo .....	215	Mitra, Sunanda .....	166
Dasarathy, Belur V. ....	368	Mohamed, Magdi .....	257
Daugherty, Walter C. ....	143	Montero, Javier .....	215
Dekorvin, Andre .....	398	Nair, Satish S. ....	118
Deutsch-McLeish, Mary .....	266	Nasraoui, Olfa .....	59
Dockery, John T. ....	248	Natarajan, Swami .....	185
Filev, Dimitar P. ....	135	Nauck, Detlef .....	296
Fleischman, Robert M. ....	496		388
Fleming, J. W. ....	524	Niskanen, Vesa A. ....	625
Flikop, Ziny .....	378	Nolan, Adam .....	488
Frigui, Hichem .....	59	Nystrom, L. ....	408
Gader, Paul .....	257	Ostaszewski, Krzysztof .....	1
Gantner T. E. ....	427	Pagni, Andrea .....	195
Gebhardt, Jorg .....	296	Pal, Nikhil R. ....	98
Gorzalczany, Marian B. ....	266	Pal, Sankar K. ....	69
Grantner, Janos .....	312	Parlos, Alexander .....	125
Gutierrez-Martinez, Salvador .....	437	Parviz, Behzad .....	598
Henson, Troy F. ....	589	Patyra, Marek J. ....	312
Holden, A. V. ....	545		565
Hu, Yong Lin .....	488	Pavel, Sandy .....	575
Huang, Song .....	108	Pedrycz, W. ....	322
Huang, X. H. ....	524		581
Hudson, D. L. ....	535	Pemmaraju, Surya .....	166
Huxhold, Wendy L. ....	589	Pflugger, Nathan .....	185
Janabi, Talib H. ....	340	Pivert, Olivier .....	478
	449	Poluzzi, R. ....	195
Joslyn, Cliff .....	458	Prade, Henri .....	7
Juang, C. H. ....	524	Quintana, C. ....	618
Karpovsky, Efin Ja. ....	224	Ramamoorthy, P. A. ....	108
Karr, C. Lucas .....	506	Ramer, Arthur .....	14
Karwowski, Waldemar .....	1	Reznik, L. ....	418
Keller, James M. ....	49		618
Keller, Jim .....	468	Rizzotto, G. G. ....	195
Klir, George .....	598	Roventa, Eugene .....	581
Koczy, Laszlo T. ....	608	Rueda, A. ....	322
Kraslawski, A. ....	408	Ryjov, Alexander P. ....	21
Kreinovich, V. ....	418	Schott, Brian .....	330
	618	Seniw, David .....	175
Krishnapuram, Raghu .....	49	Shehadeh, Hana .....	515
	59	Shieh, C-Y. ....	118
	81	Shiple, Margaret F. ....	398
Krowidy, Srinivas .....	488	Solopchenko, G. N. ....	418
Kruse, Rudolf .....	296	Stachowicz, Marian S. ....	312
	388	Steel, R. A. ....	627
Labos, A. S. ....	545	Steinlage, Ralph C. ....	427
Labos, E. ....	545		

Sultan, L. H. ....	340
	449
Sztandera, Leszek M. ....	89
Tahani, Hossein.....	468
Tavakoli, Nassrin.....	175
Teodorescu, H. N. ....	350
Tian, Y. ....	29
Tsai, Wei K. ....	125
Turksen, I. B. ....	29
	276
	286
Vaidya, Nitin .....	39

Villarreal, James A. ....	154
Wang, Haojin .....	143
Wee, Willaim G. ....	488
Whalen, Thomas .....	229
	330
Willson, Ian A. ....	276
	286
Yager, Ronald R. ....	7
Yashvant, Jani .....	154
Yen, John .....	39
	143
	185

# REPORT DOCUMENTATION PAGE

*Form Approved*  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE <b>December 1992</b>	3. REPORT TYPE AND DATES COVERED <b>Conference Publication</b>	
4. TITLE AND SUBTITLE <b>NAFIPS '92 North American Fuzzy Information Processing Society Volume 1</b>		5. FUNDING NUMBERS	
6. AUTHOR(S) <b>James Villarreal, compiler (NASA)</b>		8. PERFORMING ORGANIZATION REPORT NUMBER <b>S-702</b>	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Lyndon B. Johnson Space Center Houston, Texas 77058</b>		10. SPONSORING / MONITORING AGENCY REPORT NUMBER <b>CP 10112</b>	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) <b>National Aeronautics and Space Administration Washington, D.C. 20546</b>		11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT  <b>Unlimited/Unclassified</b>  <b>Subject Category 59</b>		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <b>This document contains papers presented at the NAFIPS '92 North American Fuzzy Information Processing Society Conference, held at the Melia Hotel Paseo de la Marina Sur Marina Vallarta in Puerto Vallarta, Mexico, on December 15-17, 1992. More than 75 papers were presented at this Conference, which was sponsored by NAFIPS in cooperation with NASA, the Instituto Tecnológico de Morelia, the Indian Society for Fuzzy Mathematics and Information Processing (ISFUMIP), the Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), the International Fuzzy Systems Association (IFSA), the Japan Society for Fuzzy Theory and Systems, and the Microelectronics and Computer Technology Corporation (MCC).</b>  <b>The fuzzy set theory has led to a large number of diverse applications. Recently, interesting applications have been developed which involve the integration of fuzzy systems with adaptive processes such as neural networks and genetic algorithms. NAFIPS '92 was directed toward the advancement, commercialization, and engineering development of these technologies.</b>			
14. SUBJECT TERMS <b>Fuzzy Systems, Neural Networks, Genetic Algorithms, Optimization, Pattern Recognition, Path Planning, Robotics, Information Processing and Vision, Decision Analysis, Control Systems</b>		15. NUMBER OF PAGES	
17. SECURITY CLASSIFICATION OF REPORT <b>Unclassified</b>		16. PRICE CODE	
18. SECURITY CLASSIFICATION OF THIS PAGE <b>Unclassified</b>	19. SECURITY CLASSIFICATION OF ABSTRACT <b>Unclassified</b>	20. LIMITATION OF ABSTRACT <b>Unlimited</b>	