REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1996

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
An Intelligent Hierarchical Decision Architecture for Operational Test & Evaluation

5. FUNDING NUMBERS

6. AUTHOR(S)

Suzanne M. Beers

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFIT Student Attending:
Georgia Institute of Technology

8. PERFORMING ORGANIZATION REPORT NUMBER
96-009D

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DEPARTMENT OF THE AIR FORCE
AFIT/CI
2950 P STREET, BLDG 125
WRIGHT-PATTERSON AFB OH 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release IAW AFR 190-1
Distribution Unlimited
BRIAN D. GAUTHIER, MSgt, USAF
Chief Administration

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

19960531 073

14. SUBJECT TERMS

15. NUMBER OF PAGES
323

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. ***(If known)***

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

★U.S.GPO:1990-0-273-271

An
Intelligent Hierarchical Decision Architecture
for
Operational Test & Evaluation

A Dissertation

Presented to The Academic Faculty

by

Suzanne M. Beers

In Partial Fulfillment
of the requirements for the Degree
Doctor of Philosophy in Electrical Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia USA
May 1996

An

Intelligent Hierarchical Decision Architecture

for

Operational Test and Evaluation

Approved:

George J. Vachtsevanos, Chairman

Edward W. Kamen

Yorai Y. Wardi

Date Approved:

17 MAY 1996

To Paul & Mao ...

Acknowledgment

I would like to gratefully acknowledge the support, guidance, and encouragement of a number of people who have helped me to get to, and through, this stage of my education.

Major General George B. Harrison and Dr. Marion L. Williams, the Commander and Chief Scientist of the Air Force Operational Test and Evaluation Center provided me the incredible opportunity to take time away from my Air Force career to pursue a Ph.D. at Georgia Tech. I am grateful for the confidence that they showed, and continue to show, in me, and I hope that with my service at AFOTEC I can give back to them, and to the organization, at least a fraction of what they have given me.

My advisor, Dr. George J. Vachtsevanos, invited me into his graduate student group with open arms, giving me a place to begin to see what the Doctor of Philosophy concept really means. He has guided me through the process and provided the insight I needed to successfully navigate it.

The members of my committee -- Dr. Yorai Wardi, Dr. Edward Kamen, Dr. Samuel Blankenship, and Dr. Stephen DeWeerth -- took time from their busy schedules to read my work and provided valuable comments, contributing to the quality of the document.

Dr. Sam Baty, provided not only the data necessary to complete a portion of the research, but reaffirmed my (sometimes faltering) beliefs that what I was working on was really needed by the Air Force, the Department of Defense, and the operational testing community, in general.

The members of the staff at the Headquarters, Air Force Operational Test and Evaluation Center who provided support throughout this process; especially, Col T.D. Woodruff, Capt Terri Langenfelder, TSgt Tom Holloway, Mr. Steve Ganger, and Mr. Ron Gustafson.

My mother, Ms. Marcela M. Beers, provided the early foundation and emphasis on education that have led me through all my educational endeavors. Her never-ending support for whatever path I wanted to pursue, has made the trek down each a rewarding experience.

Our friend, Mr. Steven R. Jones, was an immeasurable help when I was learning the C-language. Many long nights were saved because of his insights and thorough understanding of the intricacies of the language. The help he provided me in those early days made the implementation of the methodology in this work much more straightforward.

And last, my husband, Mr. Paul A. Baye, who, with his strict Teutonic upbringing and his naval submariner training and operational experience, provided the "push" that made both the idea of pursuing a Ph.D. and the completion of all the requirements a reality. His unparalleled vision and unending perseverance made this possible.

TABLE of CONTENTS

LIST of TABLES

LIST of FIGURES

LIST of ACRONYMS

AFOTEC	Air Force Operational Test and Evaluation Center
ANOVA	Analysis of Variance
AR	Autoregressive
ARIMA	Integrated Autoregressive Moving Average
ARMA	Autoregressive Moving Average
ASPJ	Advanced Self Protection Jammer
Bel	Belief
BMF	Basic Membership Function
BPA	Basic Probability Assignment
COEA	Cost and Operational Effectiveness Analysis
COMMFFY	Composite Fuzzy Membership Function
DOC	Degree of Certainty
DoD	Department of Defense
D-S	Dempster-Shafer
DT&E	Developmental Test and Evaluation
EC	Electronic Combat
ECM	Electronic Countermeasures
E-FCM	Extended Fuzzy Cognitive Map
ESAMS	Enhanced Surface to Air Missile Simulation
FAM	Fuzzy Associative Memory
FCM	Fuzzy Cognitive Map
FMEA	Failure Modes and Effects Analysis
HWIL	Hardware-in-the-Loop
IBL	Increase in Break Locks

IDA	Institute for Defense Analysis
IHDA	Intelligent Hierarchical Decision Architecture
ITE	Increase in Track Error
JSOW	Joint Standoff Weapon
MA	Moving Average
MOFP	Measure of Functional Performance
MOTA	Measure of Task Accomplishment
M&S	Modeling and Simulation
OTA	Operational Test Agency
OT&E	Operational Test and Evaluation
P_k	Probability of Kill
Pl	Plausibility
RIG	Reduction in Guidance
RIH	Reduction in Hits
SAM	Surface-to-Air Missile
SRAM	Short Range Attack Missile
T&E	Test and Evaluation
TEWS	Tactical Electronic Warfare System
TOJ	Track on Jam
TSPI	Time, Space, Position Information
UFP-ONC	Unsupervised Fuzzy Partition-Optimal Number of Clusters
VV&A	Validation, Verification and Accreditation

Summary

Operational Test and Evaluation (OT&E) is testing conducted on a full-up system, to determine for a decision-maker, if it will be able to perform the tasks for which it was designed. Current methods used for the analysis of OT&E data include standard statistical methods that are adequate for summarizing the information gathered during a testing phase (information at a functional- performance level). However, these methods have proved inadequate in providing the type of information needed by the decision-maker in the OT&E context: information on the system's ability to accomplish the tasks for which it was designed.

The Intelligent Hierarchical Decision Architecture provides a neuro-fuzzy analysis methodology meant to take test data at the functional-performance level gathered in a laboratory, through modeling and simulation, or through field testing and aggregates/synthesizes it to provide information at the task-accomplishment level, where it is meaningful to the decision-maker. The analysis methodology is composed of four separate stages. Raw test data enters the Intelligent Hierarchical Decision Architecture as individual observations of the system's functional performance and the final output is a probabilistic system performance bound at the task-accomplishment level.

The first stage of the Intelligent Hierarchical Decision Architecture is the Clustering Methodology. Here, the raw test data are transformed into a Composite Fuzzy Membership Function -- a fuzzy distribution -- for further processing. A Composite Fuzzy Membership Function is formed for each test measurement by first, defining Basic Membership Functions that divide the universe of discourse for the variable into relevant fuzzy sets. These Basic Membership Functions are defined either heuristically or through a fuzzy clustering algorithm, depending on the amount of background information available for the variable. Once the Basic Membership Functions are defined, one of four

Compositional Methods is used to form the Composite Fuzzy Membership Function. The optimal Compositional Method to be used for each data set is chosen based upon the calculation of several Fuzzy-Statistical Similarity Measures. These Fuzzy-Statistical Similarity Measures were developed to relate the fuzzy distribution and a normal statistical distribution that would be derived from the same data.

Once the Composite Fuzzy Membership Functions for each test measure have been developed, they are used to stimulate a Fuzzy Associative Memory, the second stage of the Intelligent Hierarchical Decision Architecture. The Fuzzy Associative Memory transforms the Composite Fuzzy Membership Functions at the functional-performance level to a Composite Fuzzy Membership Function at the task-accomplishment level. The Fuzzy Associative Memory rules can be built from modeling and simulation data, previously gathered test data, or heuristically. The rules are written to relate each individual test measure at the functional-performance level to the measure at the task-accomplishment level. Once all the measures have been transformed to the task-accomplishment level, they are combined into a single Composite Fuzzy Membership Function at that level using the result of the Reduction Theorem. Thus, the output of the second phase of the Intelligent Hierarchical Decision Architecture is a single Composite Fuzzy Membership Function representing an aggregation of all the gathered test data.

The third stage of the Intelligent Hierarchical Decision Architecture makes use of a Fuzzy Cognitive Map to adjust the measured system performance at the task-accomplishment level to account for factors that could not be controlled or tested during the testing phase. In this stage, several expert-generated Fuzzy Cognitive Maps are combined to yield a single, global Fuzzy Cognitive Map representing the experts' consensus. Each factor is considered in turn, and its impact on the outcome of the testing effort is accessed. The best-case and worst-case adjustment to the test-measured performance are made using the output of the Fuzzy Cognitive Map as a fuzzy linguistic hedge. The result of the third stage, then, is two Composite Fuzzy Membership

Functions, at the task-accomplishment level, representing the Best-Case and Worst-Case Adjusted System Performance.

The final stage of the Intelligent Hierarchical Decision Architecture is to aggregate the information across all the logical divisions of the system performance. This is done using Dempster's Rule of Combination from the Dempster-Shafer Theory of Evidential Reasoning. The evidence from each logical division, in the form of fuzzy sets, is transformed to basic probability assignments using alpha-cut levels of the fuzzy sets. Each division's evidence is combined in turn, to provide the final probabilistic belief interval that is provided to the decision-maker as the final Intelligent Hierarchical Decision Architecture output.

As a proof of concept, the Intelligent Hierarchical Decision Architecture methodology has been applied to data from a program loosely based upon an Air Force Operational Test and Evaluation program. The data from that testbed program consisted of ten observations of six functional-performance measures against four separate enemy threat systems: 240 data points. The Intelligent Hierarchical Decision Architecture was used to aggregate and synthesize these data into a best-case probabilistic performance bound and a worst-case probabilistic performance bound at the task-accomplishment level.

Finally, an information content measure, based upon fuzzy entropy concepts was developed to measure the amount of ambiguity associated with decision-making throughout the stages of the Intelligent Hierarchical Decision Architecture. This measure shows that the difficulty associated with making a decision decreases as the data is processed through the Intelligent Hierarchical Decision Architecture.

CHAPTER ONE

INTRODUCTION

1. INTRODUCTION

Decision-makers long for meaningful information that will make their decision processes straightforward and accurate. They prefer not to have to wade through reams of computer printouts, mountains of data products, or page after page of statistical analyses to understand the information they need to make an informed decision. Decision-makers want information that is relevant, meaningful, and concise. This need for information at the appropriate level is universal -- from the individual trying to decide what money market fund to invest in, to the plant manager trying to decide whether the product being produced will meet market demands, to the military strategist trying to determine if the new piece of equipment will help the force accomplish their objectives.

Decision-makers in the information age are faced with a decision-making dilemma. They are bombarded with overwhelming quantities of information from a variety of sources, some relevant to the decision they are trying to make, and some not. How should the decision-maker separate the wheat from the chaff to make informed, yet timely, decisions? A mechanism needs to be developed through which basic information, or low-level data, can be brought together in a meaningful and systematic way, to provide information at a level that helps the decision-maker with his task. The mechanism needs to be flexible enough to cope with the inaccuracies and uncertainties in the data sources, yet structured enough to provide a deterministic method for decision-making. This research aims to develop a methodology through which low-level data can be aggregated and synthesized to produce information that is meaningful for making high-level decisions using ***intelligent techniques*** such as neural networks and fuzzy logic.

The intelligent techniques, including fuzzy logic and neural networks, are being considered for the attractive qualities they bring to bear on the problem. Neural networks and fuzzy systems estimate functions from sample data. However, unlike mathematical or statistical approaches, they do not require a mathematical model of the function -- they are model-free estimators [1]. While both can be used to represent input-output relations using a model-free approach, each has its strengths and weaknesses stemming from the mechanism used to represent the input-output associations. Neural networks are modeled after the physical architecture of the brain and depend on a vast number of interconnected neurons to encode and recall information, while recall and associations in a fuzzy logic system are based upon the way the brain deals with information, building heuristic rules to guide actions.

The human brain excels at certain tasks, yet has limited capabilities in others. For example, there is a limit to the number of mathematical calculations a person can do "in his head," yet the same person has no problem recognizing a familiar face in a crowd or picking out the conversation on a noisy telephone line. On the other hand, anyone who has done any computer programming knows that a computer can be easily programmed to perform mathematical calculations far beyond the abilities of most human beings. However, that same programmer has experienced the frustration associated with programming the computer to recognize characters that a three-year child can easily discern. Because the neural network structure is patterned after the brain's construct of interconnected neurons, it is most suited for those applications that the brain can easily accomplish, such as pattern recognition, signal classification, and prediction tasks [2]. The drawback of neural networks is their black-box mode of operation. Once a network has been trained to accomplish a task, it will accomplish that task dutifully. However, if the user wants to understand why the neural network made the decision that it did, he is faced with a tangle of interconnections and weights that provide little or no intuitive insight into the network's operation. Another drawback of the neural network structure, is its propensity to "relearn" new patterns or associations, replacing those already encoded into the network. This feature causes uncertainty in the knowledge of what the neural network has actually learned, and what the network will recall when it is used. Finally, a large amount of data is required to train a neural network. In the case being examining in this work, the quantity of data that would be required to adequately train the network, is seldom available.

On the other hand, fuzzy logic-based systems operate based on the human common-sense reasoning approach; such as, if the room is cold, turn on the heater for a while. Fuzzy logic provides a mechanism for quantifying the concepts "cold" and "awhile" and building a structure for determining which actions should be taken as a result of the current state of the system. Additionally, the mechanism of the fuzzy rule base which is used for determining the outputs/actions, allows the user to backtrack through the inference process and determine which input conditions caused the decision that is being made. This is an attractive property for a system used by a decision-maker who wants to understand the rationale for the decisions that are being made. The drawbacks mentioned above in neural network operation -- the inability to justify the decisions that are being made, the inability to determine what the network is truly trained to do, and the lack of an adequate amount of training data to train the neural network -- cause this research to rely more heavily on fuzzy-logic based techniques and use the neural network structures only in the form of Fuzzy Associative Memories. The Fuzzy Associative Memory falls in the category of fuzzy neural networks because of its neural-like structure, yet retains the attractive qualities of the fuzzy-logic based decision techniques -- allowing the inferencing to be justified by looking inside the system to determine which rules are being used to give current results.

1.1 TEST AND EVALUATION: A DECISION-MAKER'S TOOL

Test and Evaluation (T&E) is a process through which a system-under-test is assessed to determine its technical performance characteristics or its ability to perform certain tasks. It is conducted throughout a system's lifecycle for various reasons including circuit level testing to verify design concepts; component level testing in extreme environmental conditions to verify operational temperature constraints; and complete system testing to verify component interoperability. The results of the T&E process are a critical tool in the decision-maker's toolbox.

The information provided on system characteristics and task performance during T&E is used for varying decision-making processes throughout the system's lifecycle. These two distinct types of information (i.e., technical system characteristics and operational task performance characteristics) are the result of two distinct phases of the T&E process: ***Developmental Test and Evaluation (DT&E)*** and ***Operational Test and Evaluation (OT&E).***

DT&E is testing conducted by the system's developer in order to verify specification compliance and design maturity. OT&E, on the other hand, is testing conducted by the user of the system, or by an independent agent, in order to verify that the full-up system will be able to perform its designated tasks in its intended operational environment [3]. Both types of information are important for decision-making at various stages of the system's lifecycle. The information on fulfillment of system specifications is necessary to make decisions on design and production as the system is being built. Overall system performance information is necessary to determine if the finished product will meet customer expectations.

Determining if a product meets customers' requirements is an important aspect in the production and sale of any product. OT&E is meant to provide that type of information. Although a structured OT&E mechanism is most evident within the

Department of Defense testing community[1], similar testing is conducted by the producers and manufacturers of all types of products. The product testing conducted by the publishers of Consumer Reports magazine is an excellent example of OT&E conducted on consumer products [4]. The information provided as a result of the T&E conducted by the magazine's publisher, provides information to the consumer on the ability of the system-under-test to meet consumer requirements. This focus on providing information on the overall system performance, that is relevant to the product's end user, requires different test and evaluation tools than those used in laboratory testing, whose goal is to determine technical system performance characteristics.

1.1.1 Different Focus

The differing focuses of OT&E and DT&E (i.e., providing information on task accomplishment capabilities versus providing information on technical specification compliance) require differing sets of analysis tools. Three primary differences, described below, highlight the need for a new analysis technique to be added to the OT&E toolbox.

First and foremost, the *evaluation criteria* used in OT&E are the operational users' requirements. On the other hand, the evaluation criteria used for DT&E are the technical design specifications for the system. The contrast between DT&E and OT&E evaluation criteria can be illustrated in the considerations made in the purchase of an automobile. When an individual decides to buy a car, he is somewhat concerned that the car has been

[1] The Government Accounting Office, the Office of Management and Budget, and President Nixon's Blue Ribbon Panel recommended the creation of a ***structured operational testing mechanism*** within the military services in 1974. The four Operational Test Agencies (OTAs): the Air Force Operational Test and Evaluation Center (AFOTEC), the Army Operational Test and Evaluation Command (OPTEC), the Navy Operational Test and Evaluation Force (OPTEVFOR), and the Marine Corps Operational Test and Evaluation Agency (MCOTEA) were created in that year to provide a means for testing and evaluating system performance in as close to an operational environment as possible. The OTA's job is to determine if the system-under-test meets the warfighters' needs and to provide that information to the decision-maker who will decide whether or not the system will be acquired for use by the military services.

designed, built, and tested to an adequate set of *specifications*, but the real purchase determination isn't made until the individual has determined that the car meets all his *requirements* (e.g., has enough passenger and cargo room for his family, gets adequate gas mileage, provides enough safety features, etc.). Additionally, the buyer will want to take the car for a test drive to see how it "feels." Only after he is satisfied that the car meets all his requirements and feels good to him, will he make his purchasing decision. This car buyer has conducted an OT&E: testing the car to his *user requirements* and determining if they are satisfied. This OT&E was aimed at answering the "how does it feel" and "does it meet my needs" type of questions for the end user. Before the car made it to the showroom floor, the car's producer conducted a DT&E on the car. The DT&E was a series of tests aimed at determining if the car met its design specifications; answering the "does the engine provide the specified amount of torque to the drive train" and "does the battery output the specified voltage" type of questions. In conducting the DT&E, the manufacturer can take measurements of the system component's technical performance and compare those values to the systems' specifications: a straightforward measure and compare analysis. The OT&E analysis is a little trickier. The person conducting the OT&E has to be able to *quantify concepts that might not be measurable* (e.g., the knobs on the instrument panel do not feel right) and *aggregate the observed data into a meaningful conclusion* with respect to the decision he is trying to make on buying the car.

Closely related to the differing evaluation criteria, are the *different environments* in which DT&E and OT&E are conducted. Most of the testing for DT&E is done in a laboratory setting, or a highly controlled environment, with the system developers operating the equipment. OT&E, on the other hand, is conducted in as close to an operational environment as possible. This operational environment includes taking the system to the location where it would normally be operated, and letting the users operate the system. Frequently there is quite a difference in system performance between when a Ph.D. engineer is operating it in a sterile laboratory environment and when the end user puts it through its paces in its intended operational environment! Going back to the car purchasing example, the DT&E of the car radio might consist of looking at such things as the signal to noise ratios across the frequency bands, measuring the SNRs to insure that they meet or exceed the values that were given in the radio's specification. The OT&E on the other hand, might be a part of the customer's test drive. As he is driving the car through the dense city traffic that he has to negotiate on his way to work, he turns on the radio to see if he can receive his favorite radio station. If the reception is clear and steady, the radio will pass his OT&E. While the DT&E is a very controlled test, the performance that is being assessed during the OT&E is being *influenced by a number of factors which are not being considered or controlled.*

Finally, because DT&E is conducted on system components during the development stage and OT&E is conducted on the full-up system after it has been designed and built, there is a difference in the number of tests that each testing effort can afford to conduct. DT&E can perform numerous replications of typical experimental designs -- collecting enough data to satisfy sample size requirements to perform standard statistical analyses. While, on the other hand, because OT&E conducts tests of the full-up system in an environment meant to replicate the operational environment, the testing is typically expensive. Therefore, the *sample sizes* collected during OT&E are *frequently inadequate for standard statistical analysis techniques.*

So, the OT&E tester/analyst is faced with a number of challenges that do not concern the DT&E community. In OT&E, the tester/analyst must:

- *quantify concepts* that are not numerically quantifiable
- *aggregate observed data* to form meaningful conclusions that are at a higher information-level than where the data can typically be collected
- consider the results from experiments that include *unknown or uncontrollable factors*
- draw conclusions from *limited sample sized data sets.*

These challenges require a new set of analysis tools be added to the OT&E analysis toolbox.

1.1.2 CURRENT ANALYSIS METHODS

Current analysis methods within the OT&E community are limited to standard statistical methods and a limited use of Modeling and Simulation. Although both have proved inadequate in providing information to the decision-maker at the appropriate level, they continue to be used, and in fact, endorsed as the preferred analysis methods. Mr. Don Giadrosich, in [5], states "... statistical methods offer a sound and logical means of treatment; there is no equally satisfactory alternative." Ironically, within the same section of his work, Mr. Giadrosich also states

> **We should not overlook the fact that in certain instances a qualitative description of what happened may be more valuable than large amounts of quantitative data. ... Thus, there are instances when unquestioning acceptance of quantitative measurements would mask or even misrepresent the actual performance of an item or system undergoing testing.**

seemingly, acknowledging the need for a more satisfactory analysis method. However, throughout the rest of his work, one of the few written addressing the task of OT&E analysis, Mr. Giadrosich concentrates on the standard statistical methods currently in use by the OT&E community. Appendix A provides a brief review of those methods, broken into three broad categories: Statistical Analysis Tools, Statistical Model Building Techniques, and Modeling & Simulation.

1.1.2.1 CURRENT TOOL INADEQUACY

After their initial analysis of all of the statistical and analytical methods used in OT&E, the National Research Council affirmed those factors described in Section 1.1.1, when they listed the four aspects of operational testing contributing to its difficulty and complexity [68]:

- **The operational testing paradigm often does not lead to a pass/fail decision. Instead, testing can involve redesign, iteration on concepts, or changes in subcomponents. This aspect especially characterizes the operational testing of complex systems for which no competing capability exists. The statistical methodology appropriate for one-at-a-time pass/fail decisions is inappropriate for sequential problems; thus there is a need for more proper sequential methods that will increase the information derived from tests of this type.**

- **Operational testing involves realistic engagements in which circumstances can be controlled only in the broadest sense. Human intervention, training, and operator skill level often defy control, and can play as important a role in the performance outcome as the system hardware and software.**

- **Operational tests are often expensive. With increasingly constrained budgets, there is enormous pressure to limit the amount of operational testing solely because of cost considerations. Experiments with sparse data cannot produce information with the associated levels of statistical uncertainty and risk traditionally used to support decision-making.**

- **When attempted, the incorporation of additional sources of relevant data -- before, during, and after operational testing -- in the evaluation of complex systems poses methodological and organizational challenges. Methodological challenges arise from the difficulty of combining information from disparate sources using standard evaluation techniques. Such sources include training data on operators involved in field tests and observational data on in-use situations when they present themselves. Organizational challenges can arise where there is disagreement about the validity of certain types of information or when**

attempting to gather information in settings (e.g., combat) in which the primary objective is not data collection.

The different focus of OT&E and the inadequacy of the current set of tools, drives the need for a development of a new set of tools to meet the analysis challenge.

1.1.3 DIFFERENT TOOLS

As discussed in Sections 1.1.1 and 1.1.2, different analysis tasks require different analysis tools to adequately perform OT&E. However, the current OT&E analyst's toolbox is filled with the same statistical methods as the DT&E analyst's. These statistical methods are inadequate for the challenges faced by the OT&E community. While statistical methods provide a mechanism for describing and summarizing the collected data, they provide no means for aggregating or synthesizing the information to higher information-content levels or for quantifying concepts that cannot be measured numerically. Additionally, statistical methods such as Analysis of Variance (ANOVA) or Hypothesis Testing, depend on tight controls on the test variables and adequately-sized data samples [6] that are not possible in the dynamic environment of OT&E. A systematic means for analyzing the results of OT&E, considering the challenges described in Sections 1.1.1 and 1.1.2 is desperately needed. This research focuses on the development of a hierarchical analysis and decision methodology based upon ***intelligent techniques***, which will allow information gathered at the functional performance level to be aggregated to a higher information-content level where it is more meaningful to the end users of the information. These intelligent techniques also provide a flexible mechanism for dealing with the dynamic/uncertain OT&E environment.

It should be noted at this point, that although the decision-making dilemma facing the OT&E community has spawned this research, and that the testbed case examined throughout is a specific application to the military OT&E community, the work described here can be applied universally. In a general decision-making setting, decision-makers want information at a level where it is meaningful to the decision being made. The methodology developed here can be used by any decision-maker who has low-level data and wishes to base high-level decisions on the information it contains.

1.2 THE INTELLIGENT HIERARCHICAL DECISION ARCHITECTURE

The development of the ***Intelligent Hierarchical Decision Architecture (IHDA),*** illustrated in Figure 1, to address the OT&E analysis challenges discussed in Section 1.1, is the focus of this research. In addition to the development of the hierarchical structure, several new tools and methods were developed which advance the state of the art in both the areas of systems analysis and fuzzy logic. These advances are described as each portion of the proposed structure is discussed in more detail. Each function of the *Intelligent Hierarchical Decision Architecture* will be briefly described here to present an overview of its operation. Further detail on each portion of the architecture's development and use will be described in the following chapters.

1.2.1 SYSTEM-UNDER-TEST PERFORMANCE MEASURES

Before proceeding to a discussion of the components of the *Intelligent Hierarchical Decision Architecture*, two definitions are required. Measures of the system-under-test's performance at two information-content levels will be used in the progression through the aggregation/synthesizing process of the *Intelligent Hierarchical Decision Architecture.* These measures will be given names to associate them with their level of information-content. First, a ***Measure of Functional Performance (MOFP)*** is a measure of the system's performance at the technical performance level. The MOFP is the lowest level of information derived from a testing effort, and represents the starting point for the IHDA framework. Typically, the MOFP will be a measure that can be derived directly from test measurements in the field or in a laboratory. The MOFP characterizes the system performance at a technical level, but does not provide very much insight for the decision-maker on the system's overall worth. Going back to the car stereo system evaluation example of Section 1.1.1, one of the MOFPs used to measure the technical performance might be the maximum dynamic range of the speakers.

The information-content level above the MOFP is the ***Measure of Task Accomplishment (MOTA)***. This is a measure of how well the operational task is being accomplished by the system-under-test. Just as accomplishing one task requires the completion of several sub-tasks (e.g., the task of making a sandwich requires the subtasks of taking out the bread, spreading the mayonnaise,....), the MOTA is composed of several MOFPs. System performance at the MOTA-level usually cannot be directly measured by a testing activity; thus, information on the MOTA must be derived from an aggregation of the underlying MOFPs. The MOTA gives the decision-maker an insight into how well the required tasks are accomplished by the system-under-test. Continuing the car stereo example, a MOTA might be the system's ability to adequately reproduce FM quality sound from the FM signals transmitted by the radio station while the car travels in all types of terrain that it may encounter during operational use.

1.2.2 INTELLIGENT HIERARCHICAL DECISION ARCHITECTURE OVERVIEW

The *Intelligent Hierarchical Decision Architecture* provides the mechanism to aggregate and synthesize information gathered on low-level technical system performance (at the MOFP-level) to provide information on the system-under-test's operational task accomplishment capabilities (at the MOTA-level). Additionally, the *Intelligent Hierarchical Decision Architecture* provides a mechanism through which other factors that could not be controlled or included in the testing can be considered in the system performance outcome. With this information, the decision-maker can determine a system's worth in meeting higher level objectives and goals.

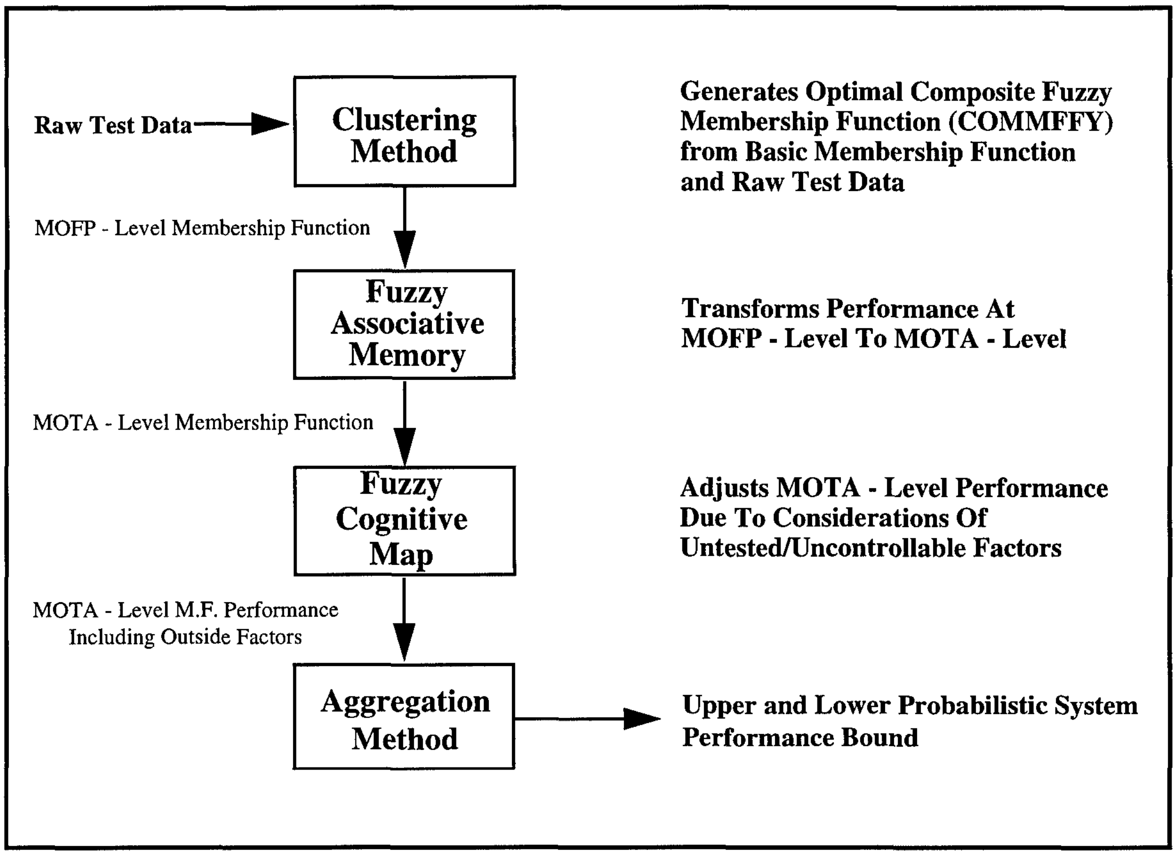

Figure 1 The Intelligent Hierarchical Decision Architecture (IHDA)

Although Figure 1 suggests a linear structure, the *Intelligent Hierarchical Decision Architecture* is actually more of a tree or funnel structure. It proceeds from multiple observations of a single MOFP at its beginning stages and ends with a single bounded performance measurement at the operational task level. A brief overview of The *Intelligent Hierarchical Decision Architecture's* functional blocks is given below. The following chapters provide descriptions of each block in greater detail.

Test data gathered either on the test range or in a laboratory enter the *Intelligent Hierarchical Decision Architecture* as raw input. These data have been pre-processed to provide measures of the system's technical performance at the functional-performance level. The ***Clustering Method*** function serves to form a ***Composite Fuzzy Membership Function (COMMFFY)*** based upon pre-defined ***Basic Membership Functions (BMFs)*** and the observed test data for each *Measure of Functional Performance* (MOFP). Each MOFP, will be processed separately, with a MOFP-level COMMFFY developed for each. Once the MOFP's COMMFFY has been formed, it will be used to stimulate the ***Fuzzy Associative Memory (FAM).*** The FAM serves to aggregate the MOFP-level performance, providing information at the MOTA-level. The output of the FAM will also be a COMMFFY, now at the MOTA-level, which indicates the system's performance at the operational task level based solely upon the information gathered at the MOFP-level. Since every factor that could potentially affect system performance is not included in the testing effort, the ***Fuzzy Cognitive Map (FCM)*** function will allow the measured MOTA-level performance to be adjusted to take into consideration factors that could not be tested or controlled during the testing phase. The Clustering, FAM, and FCM phases are accomplished for each logical division of the system performance. The ***Aggregation Method*** performs the aggregation of the system performance across these logical boundaries. The aggregation will be performed using the Dempster's Rule of Combination from the Dempster-Shafer Theory of Evidential Reasoning. The COMMFFYs from the first three phases of the *Intelligent Hierarchical Decision Architecture* will be used to form basic probability assignments through the use of alpha-cut sets. The bpa's are then combined to form belief intervals, which is the final piece of information provided to the decision-maker.

1.3 TESTBED CASE DESCRIPTION

The *Intelligent Hierarchical Decision Architecture*, once developed, will be applied to data from an OT&E program to illustrate the methodology. The best illustrations of OT&E programs can be taken from the Department of Defense (DoD).[2] An actual DoD program's data cannot be directly used in this work because of security classification and sensitivity issues associated with data gathered on system performance against enemy threat systems. However, the data used for the testbed case is based upon measures used, and data gathered, from an actual Air Force program. The type of program selected for the testbed case is an Electronic Combat (EC) program because of the poor track record within all the services' Operational Test Agencies on performing OT&E on this type of programs. The Navy's Advanced Self Protection Jammer (ASPJ) and the Air Force's F-15 Tactical Electronic Warfare System (TEWS) programs are but two recent examples of tumultuous test and evaluation activities of EC programs [7]. Additionally, a new EC system is currently under development for use on the B-1B bomber. Once this methodology is developed, it can be used in the OT&E analysis of the B-1B Electronic Countermeasures System to help the Air Force avoid the pitfalls experienced by the ASPJ and TEWS programs.

The following sections provide information on the case study: the structure of the evaluation framework, the measures of performance used to judge the system-under-test, and a sample of raw data to be used in the analysis. The complete set of raw data measurements can be found in Appendix F.

---

[2] The DoD has the most structured OT&E programs due to the DOD's inability, in most cases, to get any informal feedback on system performance of the systems they are acquiring. The basis of the testbed case is a Department of the Air Force test program, conducted by the Air Force Operational Test and Evaluation Center.

1.3.1 Strategy-To-Task Evaluation Framework

Since their creation in 1974, the services' Operational Test Agencies have conducted OT&E on systems being acquired for their services using available analysis tools. The information discovered during the conduct of OT&E is presented to high-level DoD decision-makers who are charged with making system procurement decisions: the defense acquisition decision-makers. Because of the lack of adequate analysis tools, the information presented as a result of OT&E to-date, has essentially been a summary of low-level test results. Current analysis methods did not provide a mechanism to aggregate or synthesize the test results into more meaningful information for the decision-maker. A growing dissatisfaction with the information that was being provided, coupled with a landmark report written by the RAND Corporation created a major upheaval in the conduct of OT&E in 1992.

The acquisition decision-makers were being presented with "nuts and bolts" type of information and being expected to piece the information together themselves to determine the system's worth. OT&E had become simply a replication of DT&E in an operational environment. Each individual aspect of the system's performance was being examined for its compliance with the user requirement. OT&E results were being presented as a series of pass/fail determinations against individual user requirements, with no effort by the testers to provide information on how these performance measures combined to indicate an adequate or inadequate system. The decision-makers were getting no information on the system's impact on the overall mission or how the system was going to contribute to the services' missions. They were left to determine system contributions to overall goals and objectives on their own from the pieces of test data presented to them.

Simultaneously, Lieutenant General Glenn Kent, at RAND, published a report suggesting that in an era of diminishing defense budgets and uncertain threat environments, it is extremely important to have a link between national security objectives and tasks that a system should be able to perform in order to contribute to those objectives [8]. This link would allow the acquisition decision-makers to understand where each system that was being acquired fit into the national security strategy picture. Thus, their decisions on how to allocate precious resources to the acquisition programs could be based upon national security needs. The report suggested a hierarchy of objectives from national security objectives through subordinate objectives and finally to accomplishing specific military tasks, the fundamental building blocks of military capability.

The hierarchical structure suggested by General Kent is illustrated in Figure 2. In the figure it can be seen that once the desired goals are defined, then objectives at the various levels are defined in order to meet those goals. Once the objectives at the national security, national military, and regional levels have been defined, tasks that need to be accomplished in order to meet those objectives can be delineated. Finally, once the specific tasks that need to be accomplished have been defined, functions that a system needs to perform to accomplish those tasks can be determined. The concept was, that once the hierarchical structure has been defined, information on a system's performance can be gathered at the functional performance level during the testing phase. This functional performance data can then be used to provide information to the decision-maker by flowing the information upward through the hierarchy.

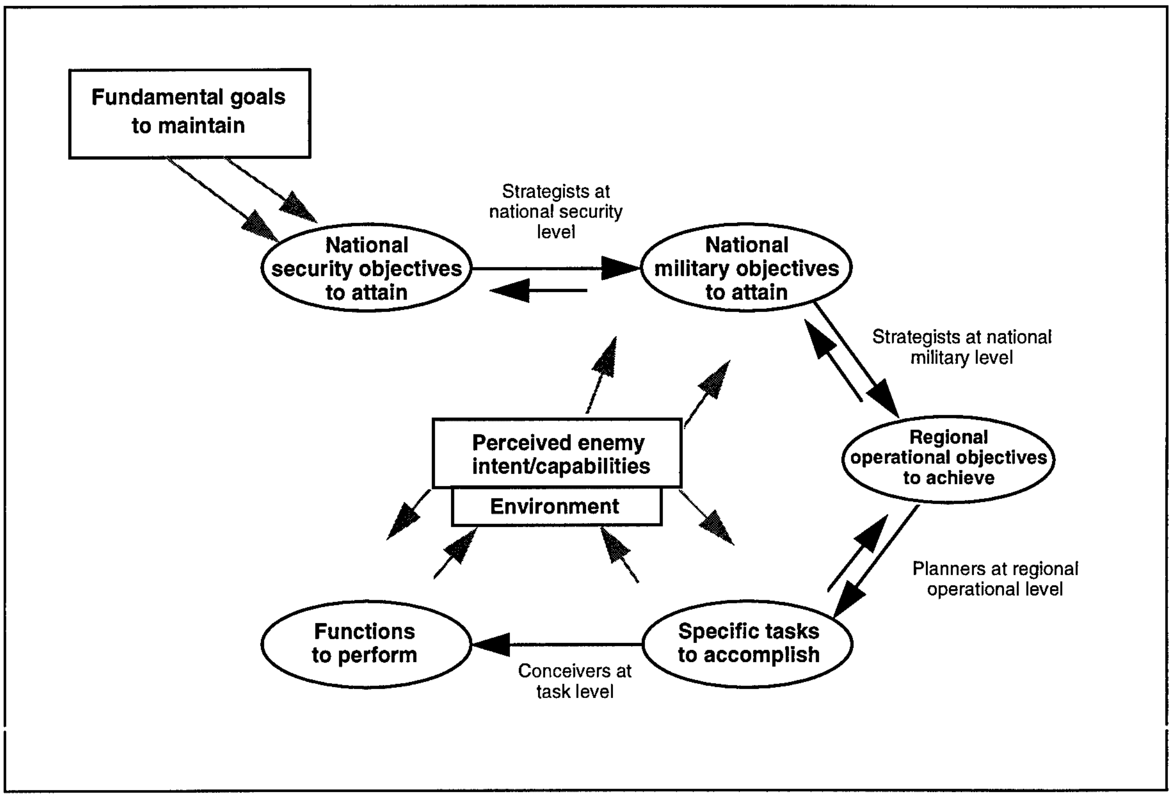

Figure 2 Hierarchy of Security Objectives: Precursor to Strategy-to-Task

This report, coupled with the already present dissatisfaction with the information being provided by the Operational Test Agencies, spawned the acquisition decision-makers to demand information from the testers on system performance at the operational task level. With information on the system's task-accomplishment capabilities, the decision-makers would then easily be able to determine how procuring the system would contribute to national security objectives. The Operational Test Agencies embraced this concept of a hierarchical linkage and termed the philosophy the ***Strategy-to-Task Evaluation Framework***. The implementation of this evaluation framework in test programs began in 1992; however, *to date it has not been successfully accomplished.* The development of the hierarchical structure of national security objectives flowing down to operational tasks which subsequently flow down to system performance requirements has been accomplished successfully on a number of programs. However, *no OT&E program has been able to develop a method through which the test data and information gathered at the functional performance level can be aggregated or synthesized to provide information at the operational task level.* The OT&E community is currently working with statistical and analytical tools that only allow a summary of the phenomena seen, with no systematic method to combine the information to take functional performance level information and provide task accomplishment level information to help the decision-maker determine if the system-under-test is worth buying.

The testbed case, termed Jammer-X, will be evaluated using this Strategy-to-Task Evaluation Framework. Under this framework the low-level measures of functional performance are used to guide the data collection in the field. These measurements are then aggregated/synthesized using the *Intelligent Hierarchical Decision Architecture* to determine operational task-level performance of the system. This is the information level that is ultimately reported to the acquisition decision-makers. The decision-maker would then use the information developed within the Strategy-to-Task Framework to determine how the operational task accomplishment, determined as a result of analysis with the *Intelligent Hierarchical Decision Architecture*, contributes to higher level objectives and goals. For the testbed case, six MOFPs will be aggregated to a single MOTA. The *Measure of Task Accomplishment* that will be demonstrated is "*Percentage Reduction in Probability of Kill.*" Each of the MOFPs described in Section 1.3.1.1 contribute to the determination of that MOTA. In the overall Jammer-X Strategy-to-Task Evaluation Framework, there would be other operational tasks that would contribute to mission accomplishment. However, for the testbed case, only the portion of the overall hierarchy shown in Figure 3 will be used to demonstrate the *Intelligent Hierarchical Decision Architecture* methodology.

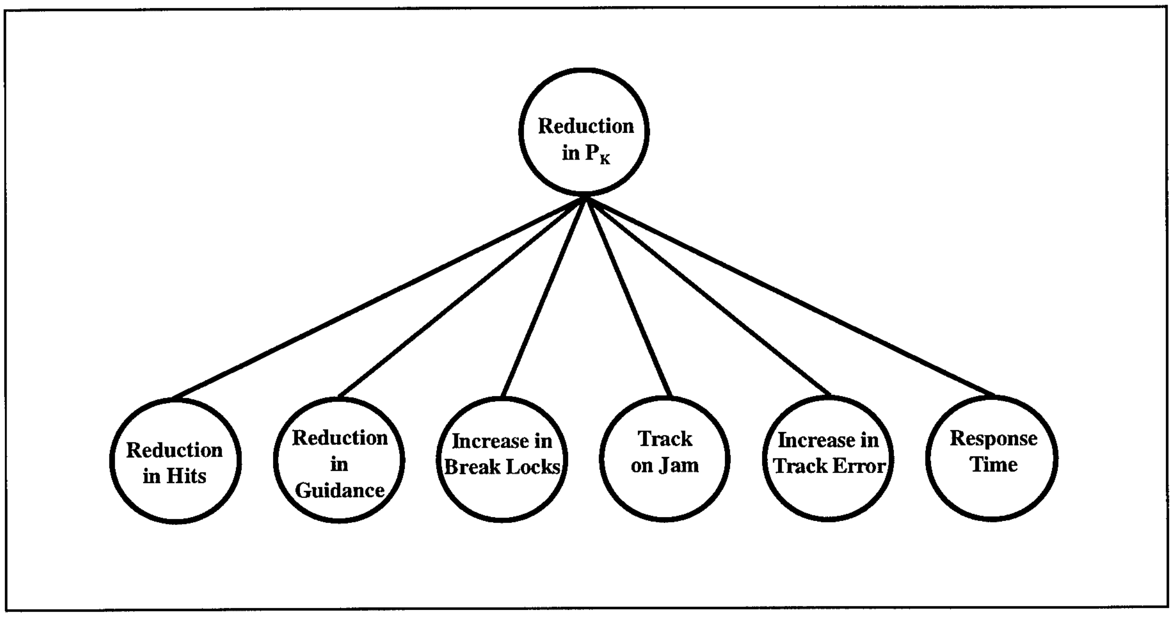

Figure 3 Testbed Case Evaluation Framework

1.3.1.1 TESTBED CASE MEASURES OF FUNCTIONAL PERFORMANCE

The lowest-level in the Strategy-to-Task framework are the Measures of Functional Performance (MOFPs) which are used to guide the data collection activities during the OT&E. The MOFPs that are used in the testing of Jammer-X are described below [9], along with the evaluation criteria that would make each of the individual MOFPs a success.

<u>MOFP 1. Reduction in Hits (RIH).</u> RIH is defined as a measure of the (percentage) reduction in the number of missile or bullet hits per pass in dry (no jamming) versus wet (jamming) conditions. Evaluation Criteria: Greater than or equal to 50%.

Method: Missile fly-out models associated with each threat simulation are used to generate shot miss distance information. If the miss distance is less than one-half the maximum length of the aircraft plus the lethal warhead radius, the shot is considered a hit.

$$\%RIH = \left[\frac{hits / pass(dry) - hits / pass(wet)}{hits / pass(dry)} \right] x100\% \qquad \textbf{(1-1)}$$

<u>MOFP 2. Reduction in Guidance (RIG).</u> Percent reduction in command guidance. Evaluation Criteria: Greater than or equal to 50%.

Method: Missile seeker guidance signals are observed for periods of interruption (nonguidance). Comparisons are made between wet and dry runs to determine if electronic countermeasures caused significant amounts of degradation. Guidance signals are only observed during time periods when the missile seeker is attempting to track a target. Percent reduction in guidance is calculated as

$$\%RIG = \left[\frac{nonguidance / pass(wet) - nonguidance / pass(dry)}{nonguidance / pass(dry)} \right] x100\% \qquad \textbf{(1-2)}$$

<u>MOFP 3. Increase in Break Locks (IBL).</u> Percentage increase in number of break locks. Evaluation Criteria: Greater than or equal to 50%.

Method: To calculate the percentage increase in break locks, the number of break locks is counted in both wet and dry conditions. The percentage change caused by the jammer for each pass is determined as

$$\%IBL = \frac{\#BL/pass(wet) - \#BL/pass(dry)}{\#BL/pass(dry)} x100\% \qquad \textbf{(1-3)}$$

<u>MOFP 4. Track-on-Jam (TOJ).</u> Percentage Track on Jam. Evaluation Criteria: Greater than or equal to 50%.

Method: To calculate the percentage TOJ, the TOJ signal in the seeker is instrumented and the signal recorded digitally or on a strip chart recorder. During periods of track, missile seekers are monitored to determine if TOJ status is reported. The percentage of TOJ is determined as

$$\%TOJ = \left[\frac{TotalTOJTime}{TotalTrackTime}\right]_{WET} x100\% \qquad \textbf{(1-4)}$$

<u>MOFP 5. Increase in Tracking Errors (ITE).</u> Percentage increase in the tracking error above a threshold. Evaluation criteria: Greater than or equal to 50%.

Method: Errors in threat system tracking are determined by comparing time-space-position-information (TSPI) target location generated by the test range instrumentation with threat target location. The percentage increase in the dry and wet tracking error is calculated as

$$\%ITE = \left\{ \frac{\left[\frac{TrackTimeGTThreshold}{TotalTrackTime}\right]_{WET} - \left[\frac{TrackTimeGTThreshold}{TotalTrackTime}\right]_{DRY}}{\left[\frac{TrackTimeGTThreshold}{TotalTrackTime}\right]_{DRY}} \right\} x100\% \qquad (1\text{-}5)$$

MOFP 6. Response Time. Difference in time between when a threat engages the aircraft and when the jammer responds. Evaluation Criteria: Less than or equal to 10 seconds.

Method: The assessment requires determining the earliest response against a threat. To calculate the response time, spectrum analyzers are attached to the jammer input and output during testing. Center frequencies and band passes were set to show only the threats of interest. The elapsed time between when a signal is observed at the input and when the response is observed is measured by a stop watch and recorded.

1.3.1.2 TESTBED CASE MOFP DATA

Table 1 shows one sample of the data that will be used for the analysis of Jammer-X's capabilities, with the complete data set contained in Appendix F. The raw data collected on the test range or in the laboratory setting are pre-processed in accordance with the equations shown in Section 1.3.1.1, to give the MOFP-level values shown here, arranged in each table by MOFP, threat system, and test run number. It should be noted from these tables that each MOFP is evaluated against four separate threat systems, and a total of ten observations is made for each MOFP/threat system combination. As mentioned previously, current OT&E analysis methods would provide a statistical mean for each of the MOFP/threat combinations (i.e., 24 separate statistical mean values) to the decision-maker, and require that he determine the system's worth from that information. The result of the *Intelligent Hierarchical Decision Architecture* methodology will provide a single performance bound across all the MOFPs and all the threat systems, with the additional consideration of other factors which could not be tested. The decision-maker's job is substantially easier when presented with that type of information versus that which he is currently provided.

The values given here and in the Appendix F tables were drawn using the uniform random number generator contained within MATLAB® because actual system performance values would reveal classified information about defense systems and, therefore, cannot be used in a research effort such as this. The data has been chosen from those generated by the MATLAB® *rand(m,n)* command such that the performance against each of the threats can be easily inferred into much better than the user requirement, just barely better than the user requirement, just barely worse than the user requirement, and both better and worse than the user requirement. Threat A is represents data that are *much better* than the user requirement for each measure, the performance against Threat B is *just barely better* than the user requirement for each measure, the performance against Threat C is *just barely worse* than the user requirement for each measure, and the performance against Threat D is *scattered throughout the range of better and worse* than the user requirement for each measure. These choices of values will allow the performance of the *Intelligent Hierarchical Decision Architecture* to be easily checked to ensure it is drawing valid inferences.

Table 1 MOFP #1, Reduction in Hits Performance for Jammer-X

	Percent Reduction in Hits			
Run Number	**Threat A**	**Threat B**	**Threat C**	**Threat D**
1	92.88	51.67	44.76	71.43
2	75.22	52.11	30.00	32.59
3	100.00	57.28	39.15	80.16
4	81.68	64.62	40.34	71.68
5	85.14	61.54	41.67	60.34
6	82.80	63.94	32.78	61.54
7	87.45	54.62	32.58	49.15
8	81.36	59.05	33.33	50.88
9	79.21	50.91	35.00	41.36
10	87.63	62.80	32.50	37.63

1.4 DISSERTATION ORGANIZATION

With this introduction, the problem this research aims to solve has been highlighted: the development of an intelligent, hierarchical decision architecture to aggregate low-level information in order to provide high-level information. The analysis method will provide a systematic means of dealing with the decision-making dilemma faced by decision-makers in the information age. The remainder of the dissertation is organized as follows.

Chapter Two is an overview of the relevant fuzzy set and fuzzy logic theories required to understand the work.

Chapter Three is a description of the Clustering Methodology, the first step in the *Intelligent Hierarchical Decision Architecture.* The Clustering Method takes the test data at the functional performance level and forms it into a *Composite Fuzzy Membership Function* (COMMFFY) -- a fuzzy distribution -- to be used in the aggregation/synthesis stages that follow.

Chapter Four describes the Fuzzy Associative Memory of the *Intelligent Hierarchical Decision Architecture*. The Fuzzy Associative Memory provides the transition from the functional-performance level to the task-accomplishment level. It takes the functional-performance level COMMFFY as input and aggregates across all the Measures of Functional Performance to yield a COMMFFY at the task-accomplishment level.

Chapter Five outlines the theory and use of a Fuzzy Cognitive Map within the *Intelligent Hierarchical Decision Architecture*. A Fuzzy Cognitive Map provides a means for quantifying and manipulating expert-provided cause-and-effect relationships. After a review of the relevant FCM theory, the FCM's use within the *Intelligent Hierarchical Decision Architecture* is described. The FCM adjusts the task-accomplishment level COMMFFY generated from the previous stages for factors that could not be controlled or included in the testing effort.

Chapter Six describes the *Intelligent Hierarchical Decision Architecture's* Aggregation Methodology, which aggregates the information generated from the first three stages across the logical divisions of the system performance. The aggregation method is based upon the Dempster's Rule of Combination taken from the Dempster-Shafer Theory of Evidential Reasoning.

Finally, Chapter Seven concludes the discussion by illustrating the *Intelligent Hierarchical Decision Architecture's* merit with a fuzzy entropy measure of the information content at each stage of its processing and through a comparison with currently available analysis methods. Also in the final chapter, the contributions of the work are summarized, and suggestions for future research efforts that would extend this work are proposed.

Appendices provide additional background information for the reader on Current OT&E Analysis Methods (Appendix A), Fuzzy Sets and Fuzzy Logic (Appendix B), Fuzzy Cognitive Maps (Appendix C), and Dempster-Shafer Theory (Appendix D). Finally, Appendix E provides the source code which implements the first three stages of the *Intelligent Hierarchical Decision Architecture* and Appendix F contains the complete details of the application of the methodology to the testbed case.

CHAPTER TWO

FUZZY LOGIC

2. FUZZY LOGIC

In Lofti Zadeh's 1973 "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," he stated [10]

> **... it is this fuzzy, and as yet not well-understood, logic that plays a basic role in what may well be one of the most important facts of human thinking, namely, the ability to summarize information -- to extract from the collections of masses of data impinging upon the human brain those and only those subcollections which are relevant to the performance of the task at hand.**

This is exactly the decision-making problem being addressed with this work -- how to determine the factors that are important to a system's ability to perform its operational tasks, when faced with masses of system performance data.

The classes of objects encountered in the physical world do not fit into clean, crisp classes or have precisely defined criteria for membership. Dr. Zadeh developed fuzzy logic in 1965 as a means of dealing with the analysis of complex systems -- providing a mechanism through which systems could be dealt with realistically without making a "fetish of precision, rigor, and mathematical formalism, and which employ[s] instead a methodological framework which is tolerant of imprecision and partial truths." In the past, Operational Test and Evaluation analysis has relied on statistical methods to summarize the aspects of the system performance demonstrated during the testing phase. It has lacked a formal method that allows the human's perception of system performance to be melded with the measured system performance to form an overall picture of the ability of the system to perform its operational mission. Pilots who fly a new plane should be able to express their opinions in such terms as "it flies too slow" or it "turns too wide" without having to give a mathematically precise formula for the system performance, and this information should be able to be incorporated into the evaluation process. Additionally, system performance in a dynamic environment should not be graded by arbitrary or artificial lines marking good and bad performance -- there should be a mechanism for gradual transitions from good to bad performance. Finally, OT&E analysis lacks a mechanism through which system performance data can be summarized, and through which the relevant pieces can be separated from the masses of data to flow forward to aid in the decision-making process. Fuzzy logic provides these needed mechanisms.

Appendix B describes the basics of fuzzy set and fuzzy logic theory. In it, how fuzzy sets and crisp sets differ is discussed; the concepts of fuzzy membership functions and fuzzy set operations are defined; linguistic variables and their modification through hedges, connection and negation operators are described; fuzzy inference mechanisms and defuzzification schemes are described; fuzzy logic's use in control applications are briefly highlighted. This information is provided for the reader unfamiliar with fuzzy set and fuzzy logic concepts. This chapter describes properties and examples of fuzzy similarity, distance, and entropy measures, which provide an important foundation for much of this research.

2.1 FUZZY MEASURES

Various measures have been proposed to measure the information content of fuzzy sets. Here, two of those concepts are briefly discussed: ***fuzzy entropy*** and ***fuzzy similarity*** measures. The entropy measures have been suggested to define the amount of uncertainty contained within a fuzzy set. By decreasing the entropy the uncertainty associated with the set also decreases. Similarity measures have formed the basis of comparisons made between fuzzy sets. They are most applicable to tasks such as clustering and classification. The current uses and definitions of these concepts, although diverse, do not completely suit the *Intelligent Hierarchical Decision Architecture's* purposes. Therefore, as this work progresses, entropy and similarity measures will be developed to meet the needs established by the work.

2.1.1 FUZZY SIMILARITY MEASURES

A ***distance measure*** of two fuzzy sets measures the difference between the two sets [11]. The ***similarity measure*** of two fuzzy sets, on the other hand, measures the similarity between the two sets. Distance and similarity measures are dual concepts.

Below the definitions of these concepts are provided, such that the measures derived in this work can be tested to ensure that they satisfy the axiomatic properties given here. In these properties, the following definitions are used:

$$\begin{gathered} R^{+} = [0, +\infty) \\ X \text{ is the universal set} \\ \mathscr{F}(X) \text{ is the class of all fuzzy sets of } X \\ \mathscr{P}(X) \text{ is the class of all crisp sets of } X \end{gathered} \qquad (2\text{-}1)$$

Thus, the definition of a fuzzy distance measure is [11]:

Definition: A real function d: $\mathscr{F}^2 \rightarrow R^+$ is called a ***distance measure*** on $\mathscr{F}$ if d satisfies the following properties:

(DP1) $d(A, B) = d(B, A), \quad \forall A, B \in \mathscr{F}$;

(DP2) $d(A, A) = 0, \quad \forall A \in \mathscr{F}$;

(DP3) $d(D, D^c) = \max_{A, B \in \mathscr{F}} d(A, B), \quad \forall D \in \mathscr{P}(X)$;

(DP4) $\forall A, B, C \in \mathscr{F}$, if $A \subset B \subset C$, then $d(A, B) \leq d(A, C)$ and $d(B, C) \leq d(A, C)$.

An example of a distance measure satisfying these properties is:

$$d_p(A,B) = (\int_0^1 |\mu_A(x) - \mu_B(x)|^p)^{1/p} \quad \forall A,B \in \Im \qquad \textbf{(2-2)}$$

The definition of a fuzzy similarity measure is [11]:

Definition: A real function s: $\mathscr{F}^2 \rightarrow R^+$ is called a ***similarity measure*** on $\mathscr{F}$ if s satisfies the following properties:

(SP1) $s(A, B) = s(B, A), \quad \forall A, B \in \mathscr{F}$;

(SP2) $s(D, D^c) = 0, \quad \forall D \in \mathscr{P}(X)$;

(SP3) $s(C, C) = \max_{A, B \in \mathscr{F}} s(A, B), \quad \forall C \in \mathscr{F}$;

(SP4) $\forall A, B, C \in \mathscr{F}$, if $A \subset B \subset C$, then $s(A, B) \geq s(A, C)$ and $s(B, C) \geq s(A, C)$.

An example of a similarity measure satisfying these properties is:

$$s_p(A,B) = 1 - \left(\int_0^1 \left|\mu_A(x) - \mu_B(x)\right|^p\right)^{1/p} \quad \forall A,B \in \Im \tag{2-3}$$

2.1.2 FUZZY ENTROPY MEASURES

Entropy is a measure that expresses the average difficulty/ambiguity in making a decision as to whether an element belongs to a set or not [12]. It can be interpreted as the uncertainty associated with a fuzzy event. Therefore, in information theory an appropriate metric in determining if the information content is moving in the correct direction, could be the fuzzy entropy. As with the fuzzy distance and similarity measures described in Section 2.1.1, various fuzzy entropy measures have been proposed. To be an entropy measure, certain properties should be satisfied. These properties are given in the definition below. The properties of entropy given below require an additional definition, in addition to those given in (2-1).

$$\left[\frac{1}{2}\right]_X \text{ is the fuzzy set of X for which } \mu_{\left[\frac{1}{2}\right]_X}(x) = \frac{1}{2} \tag{2-4}$$

Using the definitions given in both (2-4) and (2-1), the definition of fuzzy entropy is [11]:

Definition: A real function e: $\mathscr{F}^2 \rightarrow R^+$ is called an ***entropy measure*** on $\mathscr{F}$ if e satisfies the following properties:

(EP1) $e(D) = 0, \quad \forall D \in \mathscr{P}(X)$;

(EP2) $e(\left[\frac{1}{2}\right]_X) = \max_{A \in \mathscr{F}} e(A)$;

(EP3) $\forall A, B \in \mathscr{F}, e(A) \geq e(B)$, if

$\mu_B(x) \geq \mu_A(x)$ when $\mu_A(x) \geq \frac{1}{2}$ and $\mu_B(x) \leq \mu_A(x)$ when $\mu_A(x) \leq \frac{1}{2}$

(EP4) $e(A^c) = e(A)$, $\forall A \in \mathscr{F}$;

An example of an entropy measure satisfying these properties is:

$$e_p(A) = 1 - (\int_0^1 \left|\mu_A(x) - \mu_{A^c}(x)\right|^p)^{1/p} \quad \forall A \in \Im \qquad \textbf{(2-5)}$$

CHAPTER THREE

CLUSTERING METHODOLOGY

---

3. CLUSTERING METHODOLOGY

The first step in the *Intelligent Hierarchical Decision Architecture* involves taking the observed test data and creating a ***Composite Fuzzy Membership Function (COMMFFY)*** from it. This COMMFFY represents, in terms of relevant fuzzy sets for the given variable, the system performance at the functional-performance level observed during the testing effort. This composite membership function will be used in further stages of the hierarchy's processing to assist in high-level decision-making. Ultimately, the final decisions on overall system performance will be based upon the entire spectrum of low-level system performance observed during the testing phase, described by the COMMFFYs developed in this stage.

This chapter describes this first phase of the *Intelligent Hierarchical Decision Architecture*, labeled the ***Clustering Methodology***. The label for this portion of the hierarchy was chosen because the clustering of the raw test data into like groupings is similar to the clustering methodologies used in the fuzzy literature. However, typical clustering methods described in the fuzzy literature do not completely accomplish what needs to be done at this stage of the *Intelligent Hierarchical Decision Architecture*. Therefore, a new clustering methodology, that uses current methods as a starting point, has been developed. The discussion here has been broken into four separate topics: the development of the *Basic Membership Functions* that serve as the foundation for the COMMFFY building, the *COMMFFY Compositional Methods* developed to compose a COMMFFY from raw test data, the *Fuzzy-Statistical Similarity Measures* derived to assess the compositional methods' performance, and the *Optimization Method* used to determine the optimal COMMFFY for each data set.

3.1 FUZZY CLUSTERING TECHNIQUES

Fuzzy clustering techniques examine the elements of some universal set and group them according to similarity [13]. Using these techniques, each cluster becomes a fuzzy set in which the grades of membership represent the similarity between elements within the cluster. The membership function values for the points or feature vectors within each cluster are typically defined by an inverse distance measurement from the cluster center to the point or feature vector, as [14]

$$\mu_{ij} = \frac{1/d(X_j, V_i)}{\sum_{k=1}^{K} 1/d(X_j, V_k)} \qquad \textbf{(3-1)}$$

where μ_{ij} = membership value of point X_j in cluster V_i

K = total number of clusters

$d(X_j, V_i)$ = distance from sample X_j to the centroid of cluster V_i

The most commonly used clustering algorithm is the *fuzzy k-means* or *fuzzy c-means*, both by Bezdek [15]. Both algorithms require the pre-specification of the total number of clusters (represented by either the value of k in the *fuzzy k-means* or c in the *fuzzy c-means*) in the domain. The *fuzzy k-means* is based upon minimization of the objective function with respect to U, a fuzzy K-partition of the data set and V, a set of K prototypes

$$J_q(U,V) = \sum_{j=1}^{N}\sum_{i=1}^{K}(\mu_{ij})^q d(X_j,V_i) \tag{3-2}$$

where q = any real number greater than 1

N = number of data points

and the rest are as defined in (3-1).

An iterative optimization is carried out by first choosing initial guesses at the centroid prototypes, then iteratively (1) computing the degree of membership of all points (or feature vectors) to the chosen centroids and (2) computing new centroids based upon the calculated degrees of membership, until the termination criterion in the optimization is reached. The difference between the *fuzzy k-means* and *fuzzy c-means* algorithms are the functions used to calculate the degree of membership value within the iterative optimization. For the *fuzzy k-means*

$$\mu_{ij} = \frac{\left[\dfrac{1}{d(X_j,V_i)}\right]^{1/(q-1)}}{\sum_{k=1}^{K}\left[\dfrac{1}{d(X_j,V_k)}\right]^{1/(q-1)}} \tag{3-3}$$

and for the *fuzzy c-means* [16]

$$\mu_{ij} = \left[\sum_{j=1}^{c}\left(\frac{x_k - z_i}{x_k - z_j}\right)^{2/(q-1)}\right]^{-1} \tag{3-4}$$

where z_i = mean value of the fuzzy clusters (fuzzy mean) and is calculated as

$$z_i = \frac{\sum_{k=1}^{n} x_k (\mu_{ij})^q}{\sum_{k=1}^{n} (\mu_{ij})^q} \tag{3-5}$$

Both the *fuzzy c-means* and *fuzzy k-means* methods satisfy the following conditions

$$\sum_{c=1}^{n} \mu_{ij} = 1, \quad k = 1,2,\ldots n \tag{3-6}$$

$$0 < \sum_{k=1}^{n} \mu_{ij} < n, \quad c = 1,2,\ldots.c \tag{3-7}$$

The *fuzzy c-means* and *fuzzy k-means* clustering methods, although the most widely used, have been criticized for their inability to generate membership function values that are independent measures of the membership within a cluster. The membership function values, due to the "conservation of total membership law" given in (3-6), are spread across all the available classes -- making the values dependent on the number of clusters present.

To address this concern, a *Possibilistic Clustering Method* is suggested in [17] that defines a possibility distribution function that depends only on the distance from each point to each cluster center, as

$$\mu_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i} \right)^{1/m-1}} \tag{3-8}$$

where d_{ij} = distance from sample X_j to the centroid of cluster V_i

η_i = a parameter that specifies the distance at which the membership function equals 0.5; essentially the 3dB point of the cluster

m = a weighting exponent called the fuzzifier

This function satisfies the conditions

$$0 < \sum_{j=1}^{n} \mu_{ij} \leq n \ \text{ for all } i \tag{3-9}$$

$$\max_i \mu_{ij} > 0 \ \text{ for all } j \tag{3-10}$$

yet is not constrained with the "conservation law" given in (3-6). Therefore, the membership function values in a cluster are not dependent on the number of other clusters. Additionally, this formulation ensures, with condition (3-10), that the entire domain of interest will be covered with fuzzy classes.

The problem with all of the clustering methods discussed thus far, is that the number of clusters must be specified before the clustering procedure begins. In some instances this is not a desirable attribute, for it may be very difficult to achieve *a priori.* To solve this, Gath and Gena proposed the *Unsupervised Fuzzy Partition-Optimal Number of Clusters* (UFP-ONC) clustering method [18]. In their method, they suggest an unsupervised tracking of cluster prototypes through which they begin with one cluster at the mean of the data distribution, then continuously add cluster prototypes until a pre-determined maximum number of clusters is reached. Three performance measures are proposed to compare different clustering schemes. The three measures are the Fuzzy Hypervolume, a measure of the size of the cluster, which is to be minimized, and the Average Partition Density and the Partition Density, measures of the number of members within each cluster, which is to be maximized.

For the *Intelligent Hierarchical Decision Architecture*, a different sort of clustering method is required. Rather than developing clusters with each element carrying a degree of membership representing its compatibility within the cluster, a clustering method that will take all the raw data measurements for a given test variable and form a representative fuzzy distribution is needed. These fuzzy distributions, named *Composite Fuzzy Membership Functions (COMMFFYs)*, will represent, in fuzzy terms, the distribution of test observations seen for an individual test measurement. The COMMFFYs, once formed, will be used as input to further stages in the *Intelligent Hierarchical Decision Architecture*, such that the high-level decisions derived as an output of the *Intelligent Hierarchical Decision Architecture* are firmly based upon the low-level test measurements seen during the testing phase.

The following sections describe the *compositional methods* used to generate the COMMFFYs, several *similarity measures* used to optimize the COMMFFY generation process, and the results from the COMMFFY generation of the testbed data.

3.2 THE IHDA CLUSTERING METHODOLOGY

The methods proposed here will yield a *Composite Fuzzy Membership Function* from the test data collected in the testing phase at the functional performance level. That functional performance level COMMFFY is then used in subsequent processing by the *Intelligent Hierarchical Decision Architecture*.

3.2.1 Basic Membership Function Derivation

Just as with any fuzzy modeling approach, the first step required in generating a *Composite Fuzzy Membership Function* is to divide the universe of discourse into relevant fuzzy sets. Each of these fuzzy sets will be given a linguistic tag and a region of support within the universe of discourse. Using these fuzzy sets, termed ***Basic Membership Functions (BMFs)***, as the foundation, the COMMFFY is derived from the test data. Depending on the amount of background information/data that is available for a given application, one of two approaches to defining the BMFs can be adopted.

If enough data exist to characterize the input space, a fuzzy clustering method will be employed to define the BMFs. Unless the application dictates the number of clusters that should be used to make up the BMF set (in which case a strict *fuzzy k-means* or *fuzzy c-means* clustering method should be employed), the *Unsupervised Fuzzy Partition-Optimal Number of Clusters (UFP-ONC)* method will be employed to define the clusters that will comprise the BMF set. The UFP-ONC is a two-stage clustering method. In the first stage, a variation of the fuzzy k-means algorithm is used, however, no initial conditions on the number of clusters or centroid location(s) need be made. In the second stage, the cluster prototypes generated in the first stage are used in a second clustering algorithm to optimize the partitioning. These two stages are repeated for increasing numbers of clusters, until the maximum allowed is reached. The final determination of the optimal clustering is made by examining the values of three performance measures that will be used to optimize the choice of clusterings [18].

Specifically, the development of the BMFs, given that enough data exist to use a clustering method, will proceed as follows.

(1) Compute primary centroids, V_i (prototypes) using

(a) Compute the mean and standard deviation of the whole data set

(b) Choose the initial cluster prototype at the data set mean value, subsequent cluster prototypes in further iterations will be placed one standard deviation from the current prototypes

(2) Compute the degree of membership of all points in all clusters, as

$$\mu_{ij} = \frac{\left[\dfrac{1}{d(X_j, V_i)}\right]^{1/(q-1)}}{\sum_{k=1}^{K}\left[\dfrac{1}{d(X_j, V_k)}\right]^{1/(q-1)}} \qquad (3\text{-}11)$$

(3) Compute new centroid values

$$\hat{V}_i = \frac{\sum_{j=1}^{N}(\mu_{ij})^q X_j}{\sum_{j=1}^{N}(\mu_{ij})^q} \qquad (3\text{-}12)$$

(4) Update degree of membership μ_{ij} to $\hat{\mu}_{ij}$ using (3-11)

(5) Iterate steps (3) and (4) until stopping criteria is reached

$$\max_{ij}\left[\left|\mu_{ij} - \hat{\mu}_{ij}\right|\right] < \varepsilon \qquad (3\text{-}13)$$

(6) Compute performance measures. The performance measure of Fuzzy Hypervolume is defined by [18]

$$F_{HV} = \sum_{i=1}^{K} \left[\det(F_i)\right]^{1/2} \tag{3-14}$$

where F_i is the fuzzy covariance matrix of the ith cluster, given by

$$F_i = \frac{\sum_{j=1}^{N} h(i|X_j)(X_j - V_i)(X_j - V_i)^T}{\sum_{j=1}^{N} h(i|X_j)} \tag{3-15}$$

where $h(i \mid X_j)$ is the posterior probability (the probability of selecting the ith cluster given the jth point or feature vector), given by

$$h(i|X_j) = \frac{1/d(X_j, V_i)}{\sum_{k=1}^{K} 1/d(X_j, V_i)} \tag{3-16}$$

The performance measure for Average Partition Density is calculated as [18]

$$D_{PA} = \frac{1}{K} \sum_{k=1}^{K} \frac{S_i}{[\det(F_i)]^{1/2}} \tag{3-17}$$

where S_i is the "sum of central members." This takes into account the membership function values of only those members within a standard deviation of the cluster center. S_i is calculated as

$$S_i = \sum_{j=1}^{N} \mu_{ij} \tag{3-18}$$

Finally, the Partition Density measure is calculated from [18]

$$P_D = \frac{S}{F_{HV}} \tag{3-19}$$

where F_{HV} is the Fuzzy Hypervolume measure given by (3-14) and S is the total sum of members, given by

$$S = \sum_{i=1}^{K} \sum_{j=1}^{N} \mu_{ij} \tag{3-20}$$

(7) Increase the number of clusters and repeat steps (2) through (6) until the maximum number of clusters to be considered is reached.

(8) Examine the performance criteria to determine the optimal number of clusters. The F_{HV} criteria, indicating the volume of each cluster, should be a minimum indicating small, tightly bounded clusters. The D_{PA} and P_D criterion, indicating the density of the points or feature vectors within the clusters, should be a maximum indicating dense clusters.

The combination of these three measures will ensure clear separation between the clusters, minimal volume of the clusters, and maximal number of points concentrated in the vicinity of the cluster centers.

If no, or very little, relevant data are available to define the BMFs using the clustering method described above, a heuristic approach should be adopted. The basis of the heuristic approach is the intuition of the person building the system, or that of experts in the field. One consideration in developing heuristic-based BMFs should be a look to the future, to the second phase of the *Intelligent Hierarchical Decision Architecture* where rules relating performance at the functional-performance and task-level are built. The functional-performance level BMFs should be developed such that they facilitate the MOFP-to-MOTA transformation. Additionally, if one region within the universe represents a more interesting area of concentration for the variable, more BMFs should be defined in that region. Each input variable can have a different set of BMFs defined in order to facilitate a logical description that variable.

Once the BMFs are defined for each test variable using either the clustering or heuristic approach, each BMF is given a linguistic tag, to facilitate the common-sense reasoning approach of the methodology. With the *Basic Membership Functions* defined, the next step is the formation of the *Composite Fuzzy Membership Functions* using the compositional methods described in the following section.

3.2.2 COMMFFY Compositional Methods

Once the *Basic Membership Functions (BMFs)* for each functional performance measure have been developed, the observed test data is introduced to develop the COMMFFY. The COMMFFY can be derived from the BMFs and the test data in a number of ways. Here nine different ***COMMFFY Compositional Methods*** are defined. The nine compositional methods initially considered are permutations of xxx-ALL, xxx-MIN, xxx-MAX and PROD-xxx, MAX-xxx, and MIN-xxx. As with all fuzzy operations, the operation closest to the operand is conducted first, followed by the external operation. Thus, the ALL, MIN, or MAX are performed, followed by the PROD, MAX, or MIN.

The compositional methods begin by looking at where each data point activates the BMFs. For the internal operation, each data point is used to activate the BMFs with which it comes in contact. The degree of activation, and which BMF is activated by a given data point depends on the operation being performed. For example, the xxx-ALL methods (e.g., PROD-ALL, MAX-ALL, and MIN-ALL) let each data point activate *all* the BMFs with which it comes in contact. For the xxx-MIN methods, only the BMF that is the *minimum* of those activated by the data point is activated. For the xxx-MAX methods, only the BMFs for which the data point is a *maximum* is activated. Once the inner operation has been performed, that of activating the BMFs using each data point, these activated BMFs are taken into the external operation to form the final COMMFFY. The external operation considers all the activations within a given BMF and multiplies, maximizes, or minimizes (for PROD-xxx, MAX-xxx, or MIN-xxx, respectively) the individual activation levels to determine the final activation level for that BMF. Finally, the COMMFFY is formed as the distribution of the resulting BMFs. Using the BMF set composed of five triangular-shaped BMFs and five sample data points, illustrated in Figure 4, the proposed compositional methods are illustrated below.

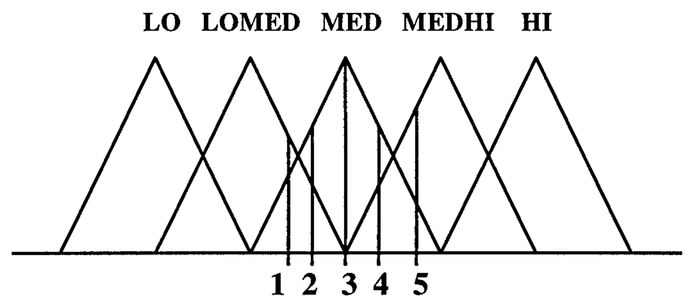

Figure 4 Sample Basic Membership Functions and Data Points

The ***MAX-ALL Compositional Method*** would let each data point activate ***all*** relevant BMFs, then the ***maximum*** value within each BMF would be used to form the COMMFFY. An illustration of the result derived from this method on the sample data points shown in Figure 4, is shown in Figure 5.

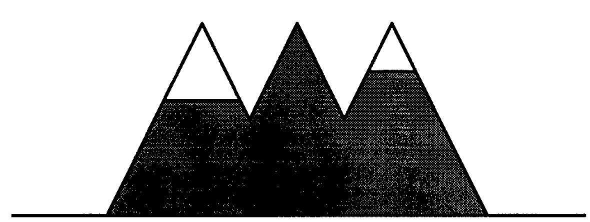

Figure 5 MAX-ALL, MAX-MAX, and MIN-MAX COMMFFY Compositional Methods Result

The ***MAX-MAX Compositional Method*** would let each data point activate only the membership function for which it is a ***maximum***, then the ***maximum*** value within each BMF would be used to form the COMMFFY. In the sample problem, the MAX-ALL and the MAX-MAX methods yield the same COMMFFY, shown in Figure 5.

The ***MAX-MIN Compositional Method*** would have each data point activate only the BMFs for which the data point is a ***minimum***, then the ***maximum*** of all the BMF activations would be used to form the COMMFFY. The result of the MAX-MIN Compositional Method for the sample problem is shown in Figure 6.

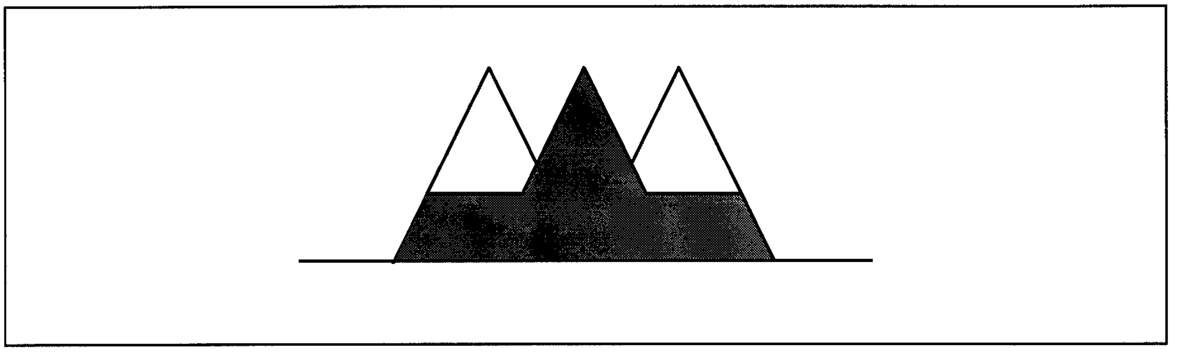

Figure 6 MAX-MIN COMMFFY Compositional Method Result

The ***MIN-ALL Compositional Method***, similar to the MAX-ALL method described above with the MAX replaced by the MIN, is shown in Figure 7. This method uses each data point to activate ***all*** possible BMFs, then uses the ***minimum*** value within each BMF as the contribution to the COMMFFY.

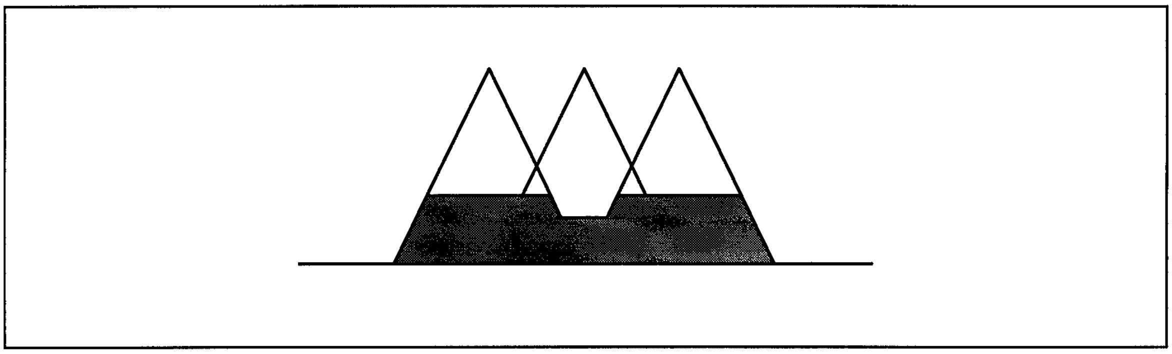

Figure 7 MIN-ALL and MIN-MIN COMMFFY Compositional Methods Result

The ***MIN-MAX Compositional Method*** lets each data point activate the membership functions for which it is the ***maximum***, then takes the ***minimum*** within each BMF for the contribution to the COMMFFY. In this example, the MIN-MAX, MAX-ALL, and MAX-MAX methods all yield the same COMMFFY, shown in Figure 5.

The ***MIN-MIN Compositional Method*** lets each data point activate the membership function for which it is a ***minimum***, then takes the ***minimum*** value within each BMF as the contribution to the COMMFFY. In this example, the MIN-ALL and MIN-MIN methods yield the same COMMFFY, illustrated in Figure 7 above.

Finally, the *product compositional methods*, denoted **PROD-ALL**, **PROD-MAX**, and **PROD-MIN** result from allowing the data point to activate the BMFs associated with ***all***, the ***maximum***, or the ***minimum***, respectively. Then the resulting COMMFFY is formed by ***multiplying*** the component values together to get the maximum value of the membership function for that region. Examples of the BMF activation values and the resulting COMMFFYs for each method are shown in Figure 8 through Figure 10.

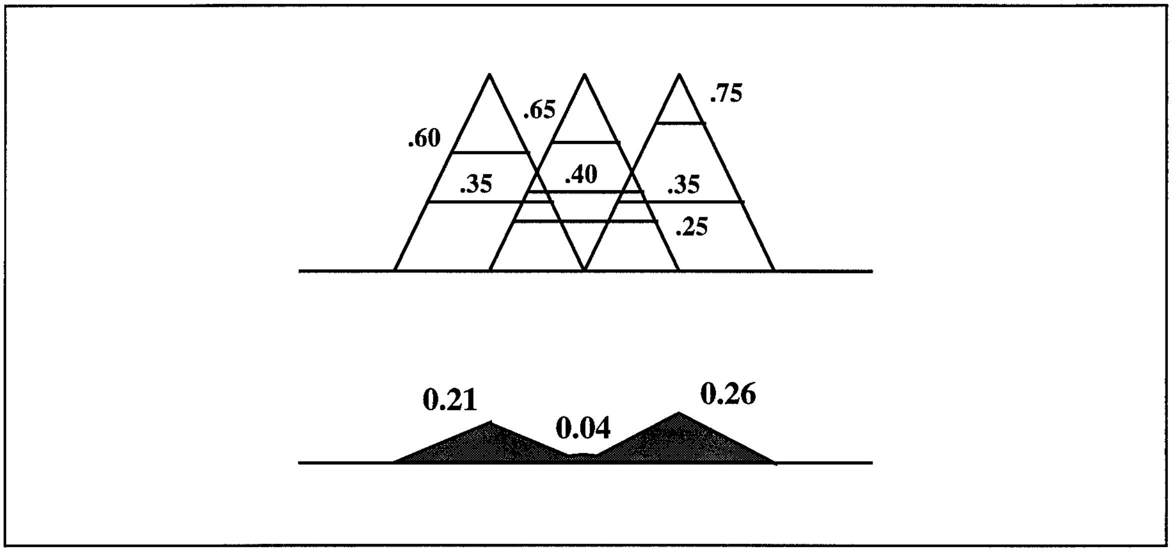

Figure 8 PROD-ALL COMMFFY Compositional Method

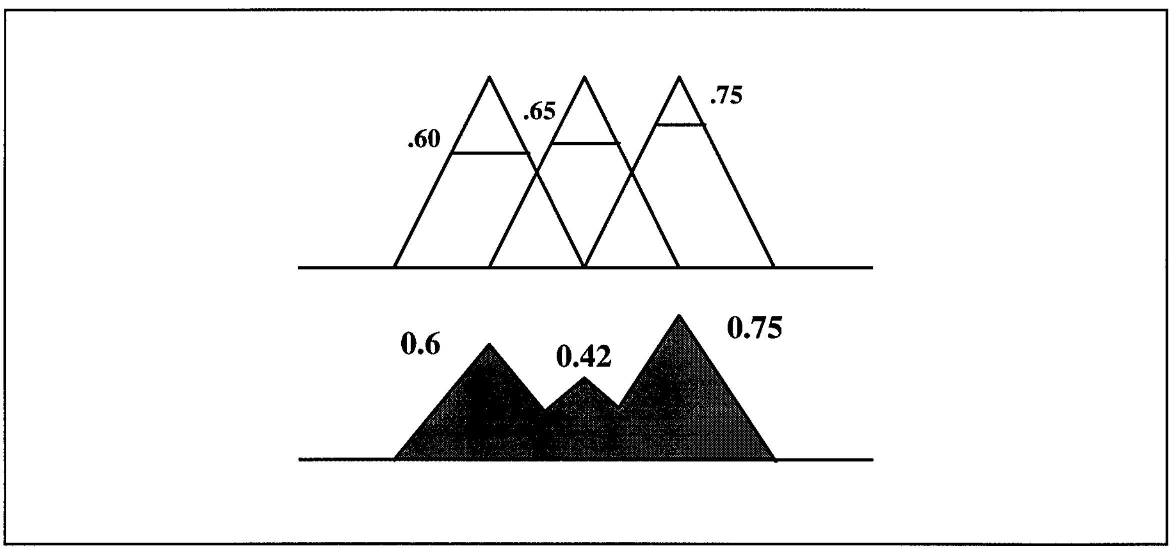

Figure 9 PROD-MAX COMMFFY Compositional Method

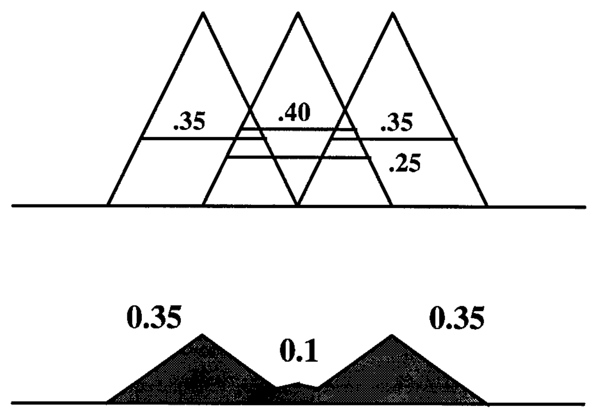

Figure 10 PROD-MIN COMMFFY Compositional Method

For the sample problem, the *MAX-MIN Compositional Method* seems to provide the most intuitively appealing result, in that it looks the most like a standard normal statistical distribution which would result from these data. The MAX-ALL, MAX-MAX, and MIN-MAX result seems to generalize all the membership functions to the maximum level, while the MIN-ALL and MIN-MIN methods seem to generalize all the membership functions to a minimum value; both extremes eliminating the distinctions in the data provided by individual data points. Meanwhile, the product-based methods penalize the generation of the COMMFFY as more data are collected, by decreasing the height of the COMMFFY with every additional observation, due to the multiplication operation.

The aim of the various compositional methods, described above, is to develop a *Composite Fuzzy Membership Function*, representative of the distribution of the observed test data. Therefore, in selecting a compositional method, the result from the method should generate a COMMFFY that resembles the underlying distribution of the data. How can these intuitive descriptions of the fuzzy distributions be quantified? By developing a set of ***Fuzzy-Statistical Similarity Measures***, relating the distribution characteristics of the COMMFFY to standard statistical distributions. The similarity measures, quantifying the relationships between the generated COMMFFYs and a statistical distribution based upon the same data, will be used to eliminate some of the compositional methods and optimize among the remaining ones.

3.2.3 Fuzzy-Statistical Similarity Measures

Various information and similarity measures have been developed to express the information contained in a fuzzy set. *Fuzzy entropy* is a measure which expresses the average difficulty in making a decision on whether or not an element belongs to a fuzzy set [12]. Several researchers have offered different formulations and interpretations of fuzzy entropy, including Kauffman [19], Deluca and Termini [20], Kosko [21], and Pal and Pal [22]. Fuzzy similarity measures have been suggested in the literature that are based upon *fuzzy subsethood* [84], *fuzzy distance measures* [23], or *fuzzy divergence* [12].

Although fuzzy information and similarity measures are abundant, none capture the type of information that is strived for in this work. The *Intelligent Hierarchical Decision Architecture's* goal is to improve the decision-making capability of the high-level decision-maker by aggregating and synthesizing low-level data into valid high-level information. The *COMMFFY Compositional Methods* have been developed to aggregate the low-level data resulting from multiple test observations into a single fuzzy distribution for further processing by the *Intelligent Hierarchical Decision Architecture.* Now, an information or similarity measure is needed which will validate this first step in the *Intelligent Hierarchical Decision Architecture's* methodology. A measure or metric is needed which will describe the information content of the COMMFFY and relate it to the information contained within the raw data set which was used to generate the COMMFFY. None of the currently-available fuzzy information or similarity measures are adequate to perform this task. Therefore, three ***Fuzzy-Statistical Similarity Measures*** are described below to fill the void.

The similarity measures derived for optimizing the choice of COMMFFY compositional methods are described below. Each compares characteristics of the COMMFFY with characteristics of the standard normal statistical distribution which would be derived from the same data set. Two of the comparisons between the fuzzy distribution and the statistical distribution use the *measure of central tendency* for each type of distribution. The measure of central tendency for a standard normal distribution is its mean, while a similar measure in a fuzzy distribution is its defuzzified value.[3] The other measure compares the integrated area under the standard normal distribution with the integrated area under the COMMFFY. Two of these measures are used to eliminate several of the COMMFFY Compositional Methods from further use, while one is used on-line in the *Intelligent Hierarchical Decision Architecture* to determine the optimal choice among the remaining compositional methods.

The Center of Area (COA) Defuzzification Method yields a crisp value that is the domain value for the center of area of the fuzzy region. The ***COA/Mean Similarity Measure*** is calculated by comparing the defuzzified value derived from the COA defuzzification method with the statistical mean, as

$$\frac{\left|y^{COA} - \mu\right|}{\mu} x100\% \qquad \textbf{(3-21)}$$

The Method of Heights (MOH) Defuzzification Method yields a crisp value from the members of the domain having membership function values greater than a given α-cut. The ***MOH/Mean Similarity Measure*** is calculated as

[3] The methods for deriving the defuzzified value from the COMMFFY used in these measures are described in Appendix B.

$$\frac{\left|y^{MOH} - \mu\right|}{\mu} x100\% \qquad (3\text{-}22)$$

Finally, a comparison of the area under a Normal Probability Distribution Function (PDF) derived from the data and the corresponding COMMFFY area is made using the ***COMMFFY/Normal Distribution Similarity Measure***, as

$$\frac{\left|NIA - CIA\right|}{NIA} x100\% \qquad (3\text{-}23)$$

where

NIA = Normal PDF Integrated Area

CIA = COMMFFY Integrated Area

All of the similarity measures described above provide a comparison between the Normal PDF characteristics and the COMMFFY characteristics, and are normalized based upon the Normal PDF values. Each of the similarity measures should be minimized to indicate a good compositional method.

Using these *Fuzzy-Statistical Similarity Measures* on the COMMFFYs generated from the testbed case, demonstrates that there is not one universally optimal method of generating the COMMFFYs. However, the similarity measures illustrate that there are some COMMFFY Compositional Methods that should be eliminated from further use in the *Intelligent Hierarchical Decision Architecture*. The methods that will be eliminated from consideration are the PROD-xxx and the xxx-MIN methods.

The product-based methods are eliminated because of their propensity to penalize the COMMFFY with each additional data point. A compositional method should build more confidence (manifested in some values in the domain having larger degrees of membership) in the COMMFFY as more data are gathered. However, the product-based methods violate this premise. As more data are gathered, the degree of membership of each BMF within the COMMFFY decreases. This trait is illustrated quantitatively by the *COMMFFY/Normal Distribution Similarity Measure* results for the testbed case shown in Section 3.3. There, it can be seen that in all cases, the integrated COMMFFY area is smaller, and in some cases substantially smaller (e.g., PROD-MIN and PROD-ALL result for MOFP #2C) than the corresponding normal curve area. Therefore, since more confidence in the decision-making ability should be generated as more data are gathered, the product-based compositional methods are eliminated from further use.

The xxx-MIN methods are eliminated, because, although for the sample problem illustrated in the previous section, the MAX-MIN method seemed to provide the most appealing result,[4] we see that as we look at the maximum values of the membership functions, the xxx-MIN methods have very low degrees of membership. Using the *MOH/Mean Similarity Measure* with $\alpha = 0.5$ on the testbed case COMMFFYs, this low membership value trait is illustrated. In almost all cases,[5] the MOH defuzzified value was zero because none of the degrees of membership within the COMMFFY exceeded the α-cut level. Again, as with the discussion of the product-based methods, the decision-making should be based on large membership function values. Therefore, the xxx-MIN methods will be eliminated.

With these eliminations, four COMMFFY generation methods remain to be used in the optimization step: MIN-MAX, MIN-ALL, MAX-MAX, and MAX-ALL.

---

[4] This result occurred because one of the data observations was exactly on the maximum membership function value of one of the basic membership functions and the minimum operation yielded a membership function value of unity. If that observation has been on any point other than the apex of the triangular membership function value, we would have seen a COMMFFY similar to the one resulting from the other xxx-MIN methods.

[5] The MOFP #2A COMMFFY has a MOH defuzzified value, and this line of reasoning does not seem to follow for that case. That MOFP's data values are extremely skewed to the right end of the data range. Therefore, in the COMMFFY generation for those data there are domain values with membership function values of unity even when the xxx-MIN methods are used. This does not hold in general for the xxx-MIN methods.

3.2.4 COMMFFY Compositional Method On-Line Optimization

The *COMMFFY/Normal Distribution Similarity Measure* and *MOH/Mean Similarity Measure* were used to eliminate the PROD-xxx and xxx-MIN compositional methods, respectively, from further consideration. With those methods eliminated, the remaining measure, the *COA/Mean Similarity Measure* is used to optimize the COMMFFY generation for a given data set. This optimization is done on-line as the COMMFFY is generated. This on-line optimization process is necessary because, as the testbed case results show in Section 3.3, there is not a clearly optimal method for generating the COMMFFY in all cases. Therefore, the flow of the COMMFFY generation, as indicated in Figure 11, includes the calculation of the *COA/Mean Similarity Measure* and selection of an optimum compositional method based upon its value for the current data set.

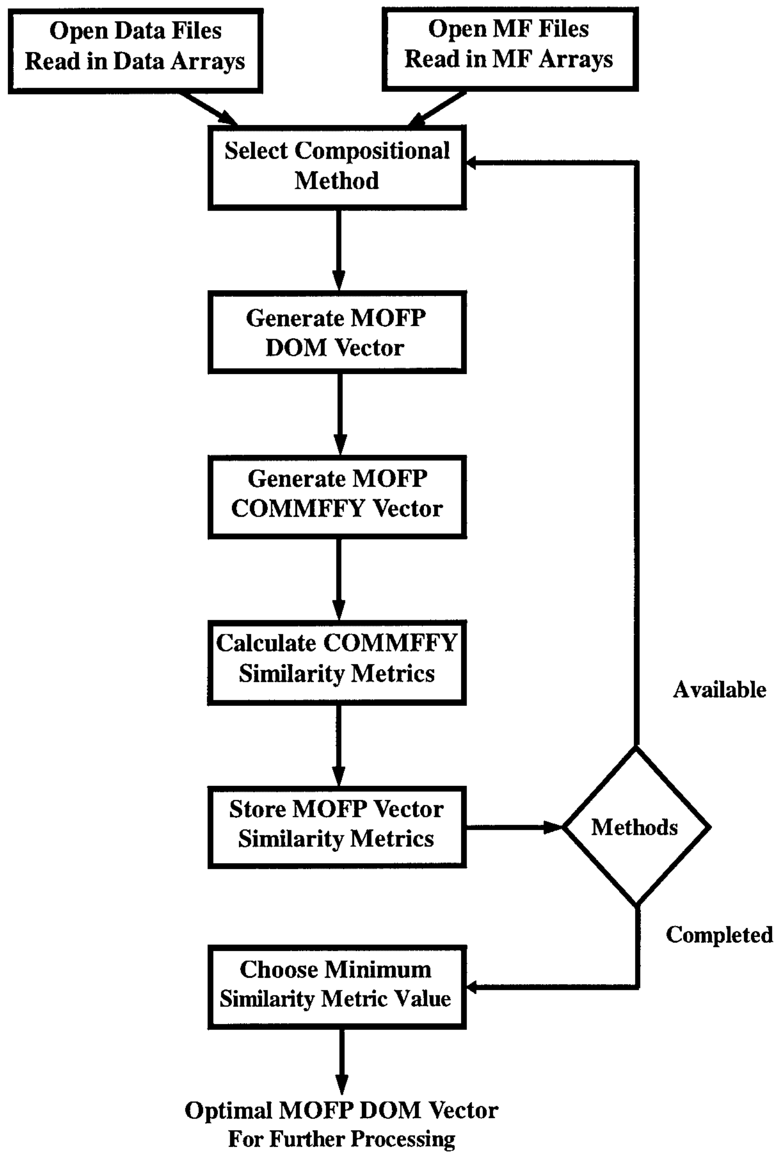

Figure 11 COMMFFY Clustering Method

As illustrated in Figure 11, the *Intelligent Hierarchical Decision Architecture's* Clustering Methodology consists of selecting a COMMFFY generation method, generating the degree of membership vector and the COMMFFY vector, then calculating the *COA/Mean Similarity Measure*. Once a COMMFFY has been generated using all the available methods, the COMMFFY with the minimum *COA/Mean Similarity Measure* value is chosen for use in subsequent processing.

3.3 TESTBED CASE RESULTS

The Clustering Methodology described in this chapter will now be illustrated using one of the functional performance measures from the testbed case. The methods will be illustrated on MOFP #2: Reduction in Guidance, for each of the four threat systems. The complete set of results, showing the resulting COMMFFY for each MOFP/Threat combination is given in Appendix F.

3.3.1 BASIC MEMBERSHIP FUNCTION SELECTION

For this case, as is usually the case for OT&E programs, not have enough relevant data is available to allow the use of the clustering methodology described in Section 3.2.1, so a heuristic approach to defining the *Basic Membership Functions* must be used. In doing this, it is noted that the functional performance measures can be divided into two distinct categories: percentage-based measures with a 50% requirement, and a time-based measurement. Triangular shaped BMFs with a 50% overlap of adjacent BMFs are chosen for their simplicity. With these choices made, the BMFs to be used for the testbed case are shown in Figure 12 and Figure 13 below.

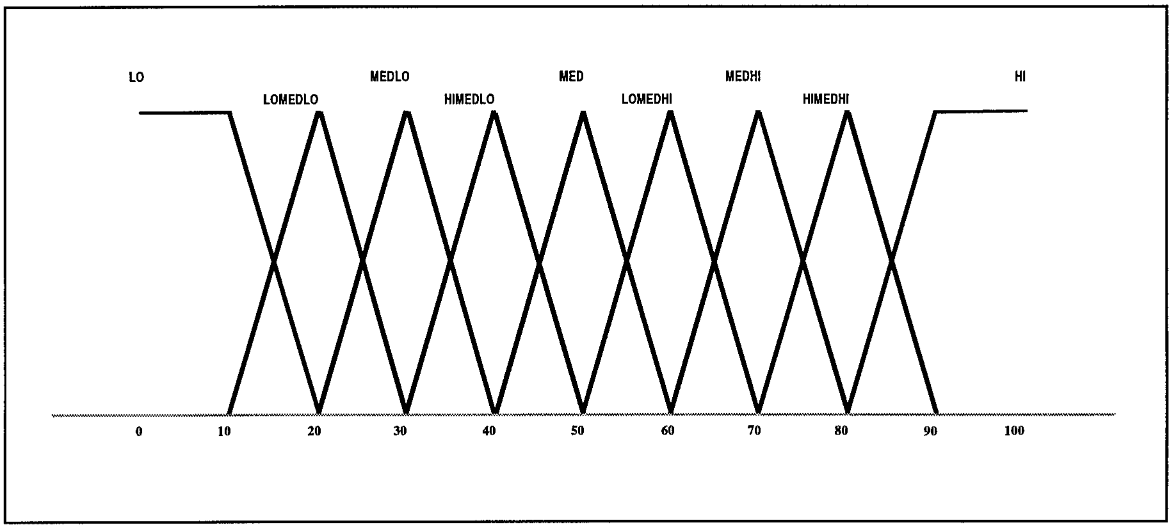

Figure 12 Percentage-Based Basic Membership Functions

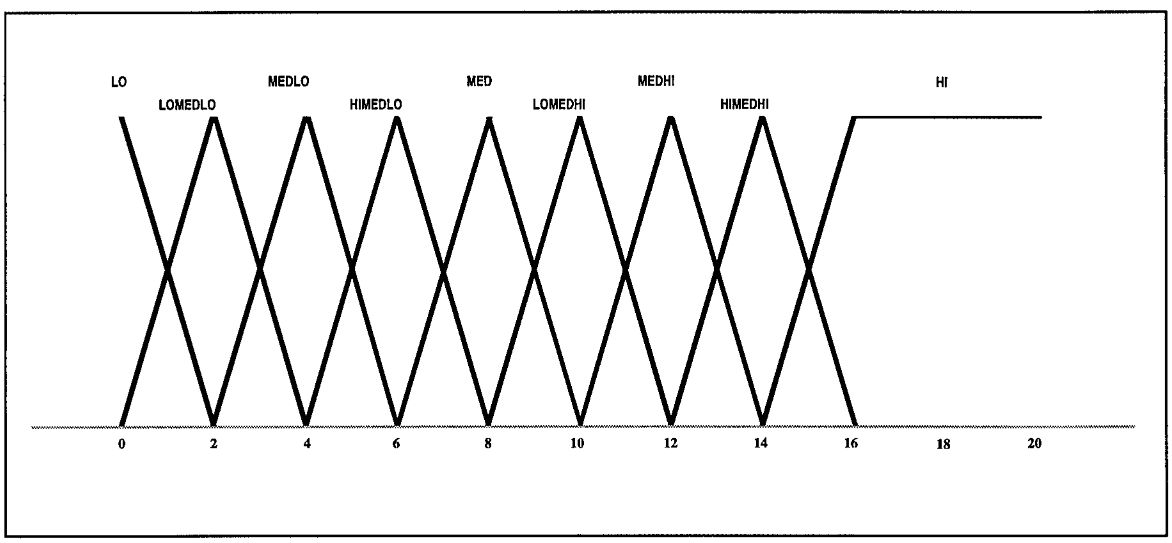

Figure 13 Time-Based Basic Membership Functions

3.3.2 COMMFFY COMPOSITIONAL METHODS RESULTS

Using the BMFs shown in Figure 12, below are illustrated the resulting COMMFFYs for one of the MOFP/threat combinations. Even though the discussion of Section 3.2.3 suggested that the PROD-xxx and xxx-MIN methods have been eliminated from further consideration as compositional methods, their results are included here to further illustrate that discussion.

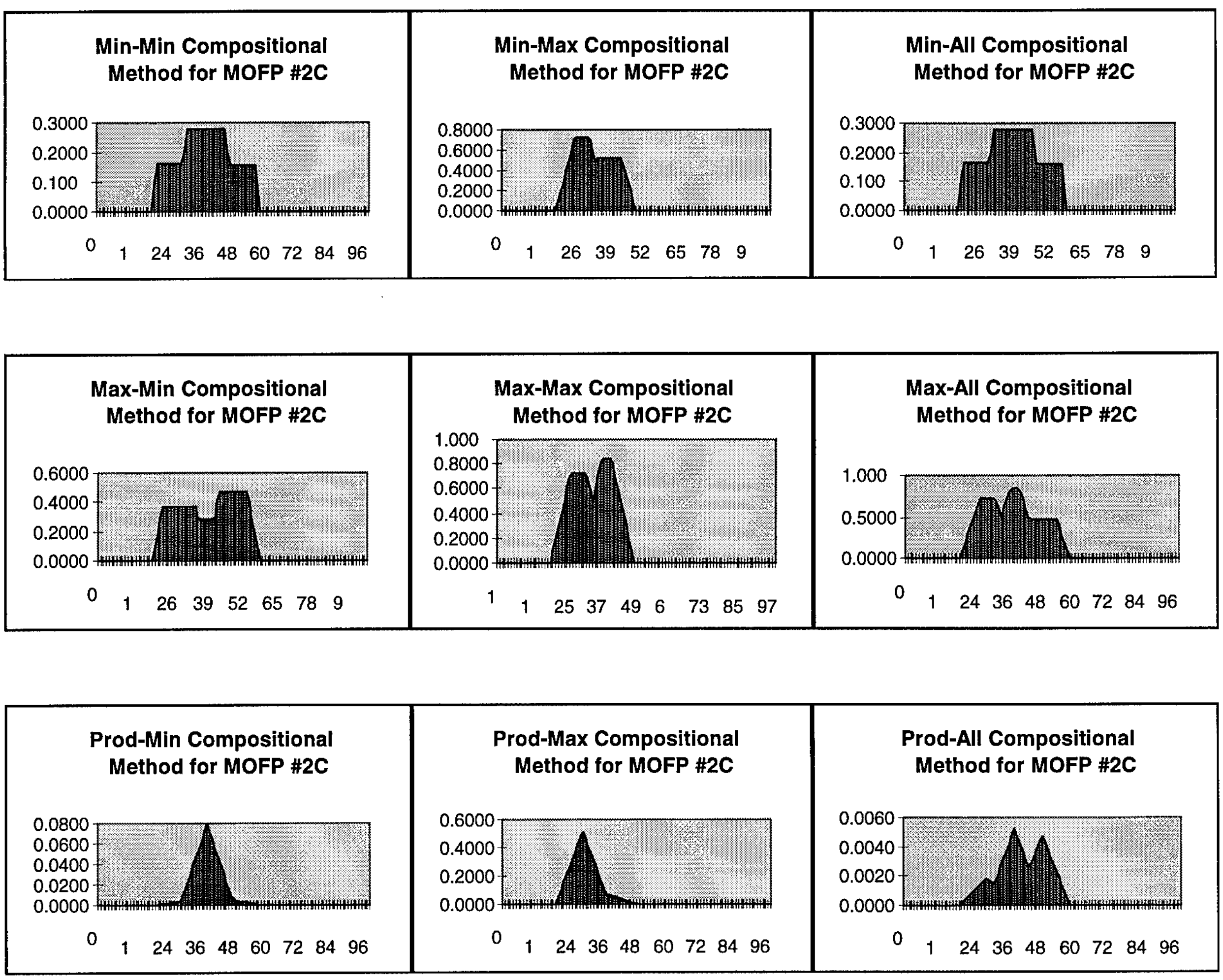

Figure 14 COMMFFYs Generated Using All Nine Compositional Methods for MOFP #2C

3.3.3 Fuzzy-Statistical Similarity Measure Results

It was asserted in Section 3.2.4 that there is not a single *COMMFFY Compositional Method* that is optimal for all cases. To illustrate this assertion, the similarity measures for the COMMFFYs from Section 3.3.2 are shown in Table 2 through Table 5. [6] Included in these tables, above each of the COMMFFY similarity measure results is a listing of the associated statistical characteristics of the data set. The results from the methods that were eliminated are shaded in light gray. The optimal method among the remaining compositional method alternatives, based upon the result of the *COA/Mean Similarity Measure*, is shaded in dark gray and bolded within each table.

[6] The similarity measures are calculated and shown for all of the COMMFFYs generated by the Clustering Method for the testbed case. Only the COMMFFYs for the system's performance against Threat C are shown in Section 3.3.2.

Table 2 MOFP #2A COMMFFY Similarity Measures Results

Mean: 86.8089

Standard Deviation: 9.5034

Data Range: [73.61, 99.10]

Integrated Normal Curve Area: 22.0453

MOFP #2A	Center of Area			COA/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	85.7695	83.9522	85.7695	1.1973	3.2907	1.1973
Max-	85.3143	83.7857	83.7857	1.7217	3.4825	3.4825
Prod-	89.2469	85.8650	90.9851	2.8084	1.0873	4.8107

MOFP #2A	Method of Heights Defuzz Value			MOH/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	93.4285	85.3546	93.4285	7.6254	1.6752	7.6254
Max-	93.4285	85.1071	85.1071	7.6254	1.9603	1.9603
Prod-	93.4285	89.5103	93.4285	7.6254	3.1118	7.6254

MOFP #2A	Integrated COMMFFY Area			COMMFFY/Normal Distribution Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	22.9410	27.0150	22.9410	4.0629	22.5431	4.0629
Max-	23.6230	28.2030	28.2030	7.1566	27.9320	27.9320
Prod-	22.0907	18.8846	16.6498	2.0593	14.3372	24.4746

Table 3 MOFP #2B COMMFFY Similarity Measures Results

Mean: 61.6760

Standard Deviation: 6.9254

Data Range: [51.94, 68.68]

Integrated Normal Curve Area: 17.3784

MOFP #2B	Center of Area			COA/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	64.1881	59.3947	64.1881	4.0730	3.6988	4.0730
Max-	62.4179	60.0765	60.0765	1.2028	**2.5933**	**2.5933**
Prod-	64.9541	58.1887	61.9561	5.3150	5.6542	0.4541

MOFP #2B	Method of Heights Defuzz Value			MOH/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	0.0000	59.2513	0.0000	---	3.9313	---
Max-	0.0000	60.0916	60.0916	---	2.5689	2.5689
Prod-	0.0000	60.0000	0.0000	---	2.7174	---

MOFP #2B	Integrated COMMFFY Area			COMMFFY/Normal Distribution Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	7.8580	20.8840	7.8580	54.7829	20.1721	54.7829
Max-	11.4310	24.2920	24.2920	34.2229	39.7827	39.7827
Prod-	4.3600	10.2380	1.1341	74.9113	41.0877	93.4740

Table 4 MOFP #2C COMMFFY Similarity Measures Results

Mean: 38.6749

Standard Deviation: 4.1911

Data Range: [32.81, 44.70]

Integrated Normal Curve Area: 10.5371

MOFP #2C	Center of Area			COA/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	39.9162	34.5089	39.9162	3.2095	4.2921	3.2095
Max-	40.7790	35.1599	39.0480	5.4404	9.0885	**0.9647**
Prod-	40.1003	30.8735	42.6761	3.6855	20.1717	10.3457

MOFP #2C	Method of Heights Defuzz Value			MOH/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	0.0000	34.3521	0.0000	---	11.1772	---
Max-	0.0000	35.2032	35.2032	---	1.2196	1.2196
Prod-	0.0000	30.0000	0.0000	---	22.4303	---

MOFP #2C	Integrated COMMFFY Area			COMMFFY/Normal Distribution Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	8.0650	14.4600	8.0650	23.4609	37.2294	23.4609
Max-	14.0550	16.4150	21.1150	33.3858	55.7829	100.3872
Prod-	0.8382	5.5371	0.0996	92.0452	47.4513	99.0547

Table 5 MOFP #2D COMMFFY Similarity Measures Results

Mean: 53.2260

Standard Deviation: 21.0762

Data Range: [16.68, 84.02]

Integrated Normal Curve Area: 51.8908

MOFP #2D	Center of Area			COA/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	53.8071	49.4011	51.3137	1.0917	7.1861	3.6566
Max-	54.2361	49.6937	50.4549	1.8977	6.6364	5.2062
Prod-	53.5387	48.0459	49.8857	0.5874	9.7322	6.2756

MOFP #2D	Method of Heights Defuzz Value			MOH/Mean Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	0.0000	49.7809	46.9651	---	6.4725	11.7628
Max-	0.0000	50.1107	50.1107	---	5.8529	5.8529
Prod-	0.0000	44.2310	20.0000	---	16.8996	62.4243

MOFP #2D	Integrated COMMFFY Area			COMMFFY/Normal Distribution Similarity Measure (%)		
	-Min	-Max	-All	-Min	-Max	-All
Min-	23.3590	49.6450	34.2450	54.9843	4.3279	34.0056
Max-	26.6200	50.4040	59.4820	48.6999	2.8652	14.6291
Prod-	17.9290	38.6891	24.4358	65.4485	25.4413	52.9091

It can be seen from these results, that the on-line optimization for these data would select the following compositional methods:

MOFP #2A: MIN-ALL

MOFP #2B: MAX-MAX and MAX-ALL

MOFP #2C: MAX-ALL

MOFP #2D: MIN-ALL

The methodology generated the COMMFFYs using these optimal methods and sent them forward to the next step in the *Intelligent Hierarchical Decision Architecture*. Overall, for the 24 COMMFFYs generated at this stage for the testbed case, the MAX-MAX method was used 8 times, the MAX-ALL was used 6 times, the MIN-MAX was used 7 times, and the MIN-ALL was used 3 times.

The next step is the Fuzzy Associative Memory which uses these optimal MOFP-level COMMFFYs as input in the synthesis from the system's functional performance information level to the task accomplishment information level.

3.4 CONTRIBUTION

The work described in this chapter makes contributions to the areas of both fuzzy set theory and systems analysis theory. First, in the area of fuzzy set theory, clustering techniques currently described in the literature only provide a means for clustering data into like groups and assigning membership function values to define the degree of belonging of each point (or feature vector) within the cluster. The Clustering Methodology described here, that results in the generation of a *Composite Fuzzy Membership Function* based upon the information available on a single variable, which can be related to the underlying distribution of raw data, is an advance to the use of fuzzy set theory in analysis tasks, previously the domain of only statistical analysis tools. Additionally, the similarity measures between the COMMFFY and standard statistical measures provide a quantitative means of relating the probabilistic and possibilistic distributions, not provided in any of the previously-developed fuzzy similarity measures. Finally, the optimization of the COMMFFY generation based upon the fuzzy similarity measures is a solution to the problem of not having a clearly optimal method. It provides a means for adapting the methodology used for generating the COMMFFY on-line based upon the data set being considered.

In the area of systems analysis, current analysis methodologies are limited to the summarization of test observations using standard statistical techniques. Typically, those techniques do not capture the essence of the underlying data completely, and once calculated provide no means for further combination of information for higher level inferencing. The COMMFFY will provide a mechanism for both summarizing the underlying data into a meaningful fuzzy distribution, and provide a mechanism for combining that information with other aspects of the system performance to allow conclusions to be drawn on the system's high-level performance based upon observed test data.

CHAPTER FOUR

FUZZY ASSOCIATIVE MEMORY

4. FUZZY ASSOCIATIVE MEMORY

Fuzzy Associative Memories (FAMs) are a class of fuzzy neural networks that provide a mechanism for mapping fuzzy sets to fuzzy sets [24]. The *Intelligent Hierarchical Decision Architecture* needs a method of mapping the *Composite Fuzzy Membership Functions* generated as a result of test measurements at the functional performance level, to *Composite Fuzzy Membership Functions* at the task accomplishment level. This mapping from functional performance level to task accomplishment level will be accomplished using a modified Fuzzy Associative Memory structure. A FAM has been chosen for the task among the various neuro-fuzzy techniques which could be used at this stage in the hierarchy for several reasons. First, the FAM allows a tracking of the decision-making process through its structure. The interested decision-maker can look at the rule base being used by the FAM, and determine why the decisions are being made as they are -- a very attractive quality for a decision aide. Second, in most cases, there is not an abundance of input/output data available to train an OT&E analysis system. Therefore, using one of the other neuro-fuzzy techniques that relies on vast amount of training data before the system is operable, is impractical for an application where that type of information does not exist. Finally, the FAM can be built from a series of heuristic rules derived from expert opinion, and then can be modified as more information becomes available. This intelligent/iterative building trait of the FAM makes it particularly attractive for an application where there is a requirement to do test planning early in the system's development, when little information on the system's performance characteristics is available. Then as more information about the system is gathered, either through Modeling & Simulation studies or early testing efforts, the information in the rule bank of the FAM can be updated to reflect the new knowledge. Therefore, the FAM has been chosen to perform the MOFP-MOTA transformation in the *Intelligent Hierarchical Decision Architecture*. This chapter discusses the basics of FAMs, describes how they will be used in the *Intelligent Hierarchical Decision Architecture*, and closes with a discussion of how the proposed usage is an extension to current theory in the FAM area.

4.1 FAM BASICS

4.1.1 FAM STRUCTURE

Fuzzy systems can be used to estimate any continuous function to any degree of accuracy without the use of a mathematical model. Using a heuristic rule of the form *"if the antecedent is satisfied, then the consequence follows"* the input-output characteristics can be modeled by associations of fuzzy sets [25]. This forms the basis of the Fuzzy Associative Memory. A fuzzy rule defines the relationship or transformation from one fuzzy set to another, in the form "IF X is *A*, THEN Y is *B*," where *A* and *B* are fuzzy sets. The FAM structure proposed by Kosko, also provides a weighting mechanism such that the output from each rule can be weighted prior to its inclusion in the output function, as shown in Figure 15.

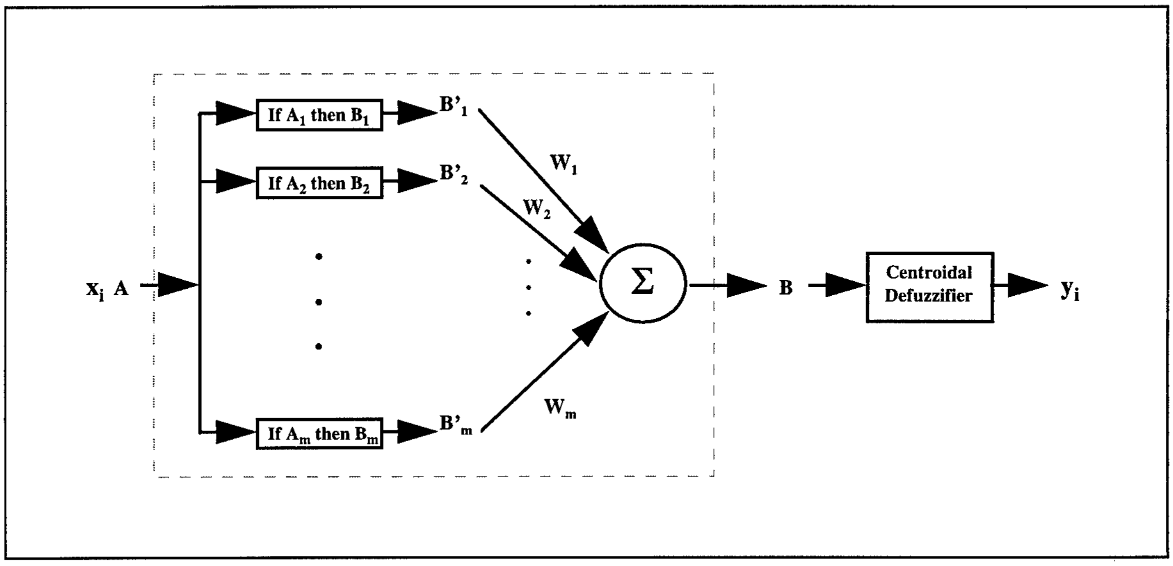

Figure 15 Fuzzy Associative Memory Structure

In this architecture, the input value fires each fuzzy rule to some degree to create a fuzzy output B_i which is weighted and summed to provide an overall output B, which is then subjected to defuzzification to yield a single crisp output, y_j. The weight factors, w_i, control the relative importance of each output result in the overall output, based upon a factor of interest to the system being controlled or analyzed. These weight factors may be static values or may be adaptive, based upon the state of a selected variable in the system.

Although the architecture pictured in Figure 15 suggests a purely parallel operation, the FAM rules can also be formulated into a two- or multi-dimensional FAM bank where each cell is a fuzzy IF-THEN rule. This configuration, without the name FAM, is what is typically seen in the fuzzy controls literature as a Fuzzy Rule Bank, the only difference is the weighting factors that are incorporated into the FAM bank structure that are absent from the standard Fuzzy Rule Bank. This weighting scheme, which some authors have interpreted as a credibility measure [85], allows the output to be a weighted combination of the fuzzy outputs inferred by the individual rules, as

$$F(y)=\sum_{i=1}^{m} w_i F_i(y)=\sum_{i=1}^{m} w_i \tau_i D_i(y) \tag{4-1}$$

where τ_i denotes the firing strength of each fuzzy rule.

4.1.2 FAM BUILDING

Various means have been suggested in the literature to construct FAM Banks, most of them depending on the enumeration of the rules based upon expert opinion or derivation, in controls application, from the control laws of the process. Heuristic rule bases built in this fashion have been used successfully in many applications. But, in addition to building the rule bank from purely qualitative information, how should a FAM Bank be constructed if input/output data from the system/process is available in addition to the expert opinion?

Given desired input-output data pairs, Wang and Mendel [26] suggest a five-step procedure to generate a set of fuzzy rules which are subsequently used in a FAM as a mapping

$$f(x) \rightarrow y \tag{4-2}$$

The steps are:

Step 1: Divide the input and output spaces into fuzzy regions.

Step 2: Determine the degree of membership of each data point within each input/output pair in each fuzzy region defined in *Step 1*, assign each data pair value to the region with the maximum degree, then derive a fuzzy rule for each data pair.

Step 3: Resolve conflicts between rules (i.e., different rules having the same antecedent but different consequences) by defining a degree for each rule by multiplication of the constituent membership functions, as

$$D(rule) = \mu_A(x_1)\, \mu_B(x_2)\, \mu_C(y) \tag{4-3}$$

Then, use the rule with the maximum degree.

Step 4: Create a combined FAM Bank by melding the rules derived from the numerical data and that derived from expert opinion.

Step 5: Define a defuzzification scheme to derive the crisp output value based upon the fuzzy output value.

This procedure provides a mechanism for defining the rules in the FAM Bank based upon rules derived from both input/output data and expert opinion. But what happens when the data samples do not cover the entire range of defined FAM cells or the FAM is multidimensional such that its size makes gathering expert opinion on every cell tedious or impractical? Sudkamp and Hammell [27] introduced the concept of completion, defined as “filling in the holes” in a FAM Bank using two methods: region growing and weighted averaging. They demonstrated that the region growing technique produced better results than the weighted averaging technique in their simulation work. Additionally, they found that the region growing technique maintains consistency in the consequence of the rules until a conflicting neighbor causes a change. This is a more realistic treatment, especially for an application such as evaluation of system performance, because changes in system performance are likely to be gradual, or caused by some external stimulus. Therefore, only the region growing method is discussed here.

Region growing, commonly used in image segmentation, fills the empty cells in a FAM by extending the values in neighboring nonempty cells. Assuming that a numerical value can be assigned to each cell, denoting that value as *val(i,j),* and defining *T* to be the set of nonempty neighbors, the value for the empty cell is obtained as

$$val(i_0, j_0) = \sum_{(i,j)\in T} \frac{val(i,j)}{|T|} \tag{4-4}$$

This calculated value is inserted in the FAM Bank and the region growing process continues until all cells are filled. In the work done by Sudkamp and Hammell, the definition of neighboring cells was narrowly defined, including, in two dimensions, only the cells directly above, below, and to the right and left of the empty cell. An extension to their research could investigate how a broader definition of neighboring cells (i.e., using the cells completely encircling the empty cell) might improve the performance of the resulting FAM Bank.

In developing the FAM Bank for the *Intelligent Hierarchical Decision Architecture*, a variation of the methodology proposed in [26] is used. In the initial phase, the development of the FAM will be fashioned after their *Step 1*, *Step 2*, and *Step 4*. Their conflict resolution procedures are not necessary in the initial FAM building procedures, because only a small amount of information will typically be available to build the FAM. It can be envisioned, however, that once the initial FAM is built and initial modeling and simulation or testing efforts yield more information on the system-under-test's performance, those conflict resolution procedures will be extensively used to modify the initial FAM. In addition, although the region growing procedures suggested in [27] provide an extremely efficient means of filling in the holes in a multi-dimensional FAM, as will be seen in Section 4.1.3, their use will be unnecessary. Due to the

Reduction Theorem proposed by Wang and Vachtsevanos, the problem can be brought down to a reasonable size; and dealing with a sparsely populated, multi-dimensional FAM can be avoided.

4.1.3 Combinatorial Explosion

The term ***combinatorial explosion*** in the discussion of FAM Banks was coined by Wang and Vachtsevanos [24] in their development of fuzzy systems; going from a Single Input, Single Output (SISO) system to a Multiple Input, Multiple Output (MIMO) system. The size of the FAM quickly grows out of control, with both the number of rules and the difficulty in the ability to visualize the FAM structure, growing with each added dimension. This same problem will be encountered with the FAM within the *Intelligent Hierarchical Decision Architecture*. Consider a case with seven functional performance measures, each with five membership function regions, yielding $5^7 = 78{,}125$ FAM rules required to accomplish the MOFP to MOTA transformation! Vachtsevanos and Wang suggested a Reduction Theorem, based upon the compositional rule of inference, to deal with the combinatorial explosion problem.

> ***Reduction Theorem:*** *Let a fuzzy implicative rule be given of the form, IF(A_{i1} and A_{i2}) THEN B_i. If the compositional rule of inference is used for the recall process, then*

$$(A'_{i1}, A'_{i2}) \circ (A_{i1} \text{ and } A_{i2} \rightarrow B_i)$$
$$= [A'_{i1} \circ (A_{i1} \rightarrow B_i)] \wedge [A'_{i2} \circ (A_{i2} \rightarrow B_i)]$$

Using this theorem, a multidimensional problem can be decomposed into individual two dimensional problems, with the result from each piece being combined using the fuzzy intersection operator to form the final result. Not only does this allow the original problem to be handled in much more manageable pieces, but because each input variable is handled with a separate set of relations, the degree of specificity of the fuzzy membership functions may be adjusted as required for each variable, without affecting the others, or further aggravating the combinatorial explosion problem. The Reduction Theorem will be used within the *Intelligent Hierarchical Decision Architecture* to allow the transformation of each individual functional performance measure to a higher-level task accomplishment measure, then once all the information has been transformed to that level, the combination using the minimum (intersection) operator will be performed to yield a single COMMFFY at the task accomplishment level.

4.2 THE IHDA FUZZY ASSOCIATIVE MEMORY

The FAM within the *Intelligent Hierarchical Decision Architecture* (IHDA) will be used to aggregate the information gathered on system performance at the functional performance level to provide information on system performance at the task accomplishment information level. The input to this phase of the hierarchy will be the functional performance level COMMFFYs generated from the gathered test data and the Clustering Methodology described in Chapter 3. The output from this phase will be a COMMFFY of the system performance at the task accomplishment level. This MOTA-level COMMFFY will subsequently be used as input to the FCM phase, which will take into consideration the factors that were not testable or controllable during the testing effort.

The issue with building a FAM to relate low-level, or MOFP-level, system performance to a higher level of system performance is the same suggested by Wang and Vachtsevanos -- combinatorial explosion. In the Jammer-X example, the FAM will relate all the jammer's MOFPs to a single higher level measure. There are six MOFPs and nine membership functions for each MOFP are used, the resulting FAM, including all the interrelationships, would have 9^6 or 531,441 cells! Clearly, a better way than a brute-force FAM which encodes each level of each MOFP in a single rule is needed. The methodology, described below, uses the Reduction Theorem as a basis for bringing the problem down to size.

Rather than discussing the FAM-building methodology in general terms, Section 4.3 will walk through each step, building the FAM for the testbed case to illustrate the discussion. The basic methodology follows the steps suggested in [26], modified by the Reduction Theorem of [24]. The steps in the methodology are:

- Define the input/output fuzzy sets
- Derive the fuzzy rule banks to relate each input variable to each output variable
- Implement the FAM using the Reduction Theorem

4.3 BUILDING THE IHDA FAM

The first task is to build the FAM rule bank. As mentioned at the beginning of this chapter, frequently in the OT&E environment, there is not a lot of information available to relate system performance at the two levels of interest. Therefore, initially the rules will be developed by experts in the field, then as the knowledge of the system under test increases, there may be other sources of information available to tune the FAM rule bank to make it more reliable. For the testbed case, a source of information beyond that provided by expert opinion was available. A one-versus-one (1-v-1) engagement model, named the Enhanced Surface-to-Air Missile Simulation (ESAMS) was used to model the interaction between the target aircraft, carrying the Jammer-X, and the threat system. Using ESAMS and running sensitivity analyses on the input variables, the relationships between each MOFP-level measure and the MOTA-level measure, *Reduction in*

Probability of Kill, were generated. Once these relationships were developed from the ESAMS sensitivity analyses, the FAM rule bank was created.

Once the FAM rule bank has been developed, the next task is to determine how it will be used. In all the current literature on the use of neuro-fuzzy tools, the input is a single value which activates the input fuzzy set to a certain degree, which subsequently activates the output fuzzy set to a certain degree, then finally using an aggregation and defuzzification scheme the crisp result from the given input value results. To more fully understand the standard uses of FAMs, a fuzzy inference system was built for the testbed case using the Fuzzy Logic Toolbox within MATLAB® [28]. From that system, it became evident that a mechanism is in place to consider only one value of each functional performance measure at a time. However, here, instead of dealing with a single input value, an input fuzzy distribution which has been generated from all the possible input values observed during the entire testing phase, is being used. It is not meaningful to separate the gathered data and use it as individual input values to the FAM, because each of the measurements were taken under differing conditions -- taking one from each MOFP to develop an input set has no physical significance to the analysis task. Therefore, a methodology must be developed through which the FAM can handle a COMMFFY as an input and provide a COMMFFY as an output.

The following sections illustrate the construction and use of the *Intelligent Hierarchical Decision Architecture's* FAM with the testbed case.

4.3.1 FUZZY SET DEFINITION

The first step in the FAM-building procedure is to divide the input and output spaces into fuzzy regions. This has already been done for the testbed case, in the definition of the *Basic Membership Functions* used in the Clustering Methodology. In Chapter 3, the MOFPs were divided into two classes: a percentage-based measure and a time-based measure. These percentage-based measures are used for most of the MOFPs, and for the consolidated MOTA in the testbed case. The definitions of the triangular-shaped fuzzy sets were given graphically in Chapter 3. These definitions, now in terms of their apex location, region of support, and linguistic tag are in Table 6, below.

Table 6 Percentage-Based Basic Membership Function Definition

	Percentage-Based Fuzzy Set Characteristics	
Linguistic Tag	Apex Location	Region of Support
LO	Plateau, 0 - 10	0 - 20
LOMEDLO	20	10 - 30
MEDLO	30	20 - 40
HIMEDLO	40	30 - 50
MED	50	40 - 60
LOMEDHI	60	50 - 70
MEDHI	70	60 -80
HIMEDHI	80	70 - 90
HI	Plateau, 90 - 100	80 - 100

In addition to the percentage-based fuzzy set definitions, there is one MOFP within the testbed case which is a measure of time delay. The fuzzy set definitions for that MOFP are given in Table 16. The C-language code that implements the handling of the fuzzy sets has been parameterized such that any fuzzy set definitions can be used simply by defining a file with the membership function limit points. The code is written to read the values contained in the file into an array, then works with the relative array locations rather than the numerical values.

Table 7 Time-Based Basic Membership Function Definition

	Time-Based Fuzzy Set Characteristics	
Linguistic Tag	Apex Location	Region of Support
LO	0	0 - 2
LOMEDLO	2	0 - 4
MEDLO	4	2 - 6
HIMEDLO	6	4 - 8
MED	8	6 - 10
LOMEDHI	10	8 -12
MEDHI	12	10 - 14
HIMEDHI	14	12 - 16
HI	Plateau, ≥ 16	≥ 14

4.3.2 FUZZY RULE DERIVATION

The next step in the FAM-building procedure is to derive the fuzzy rule for each input/output data pair. As described earlier, in the early stages of a test program, there may not be much information available to derive these rules. When this is the case, heuristic rules derived from experts in the field must be used. For the testbed case, an engagement simulation that models interactions between threat systems and aircraft, the Enhanced Surface-to-Air Missile Simulation (ESAMS) was used to develop relationships between each MOFP-level measure and the MOTA-level measure. Sensitivity studies were run and the results were given in graphical form [29] as shown in Figure 16 through Figure 21.

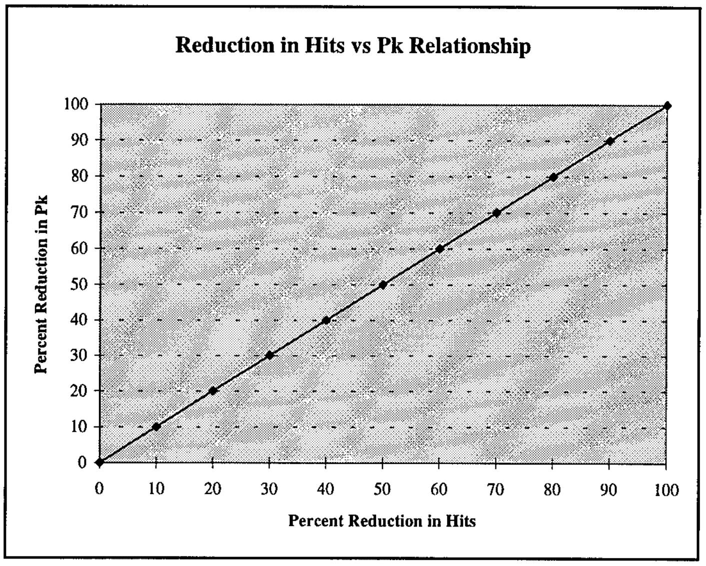

Figure 16 Reduction in Hits vs. Reduction in P_k Relationship

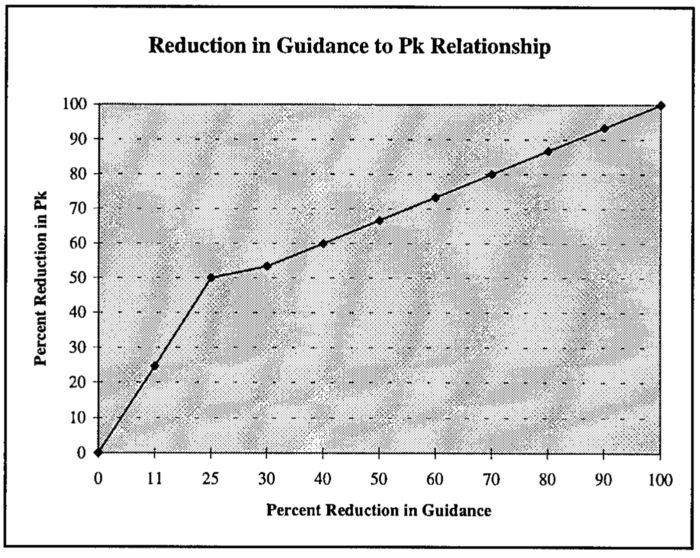

Figure 17 Reduction in Guidance vs. Reduction in P_k Relationship

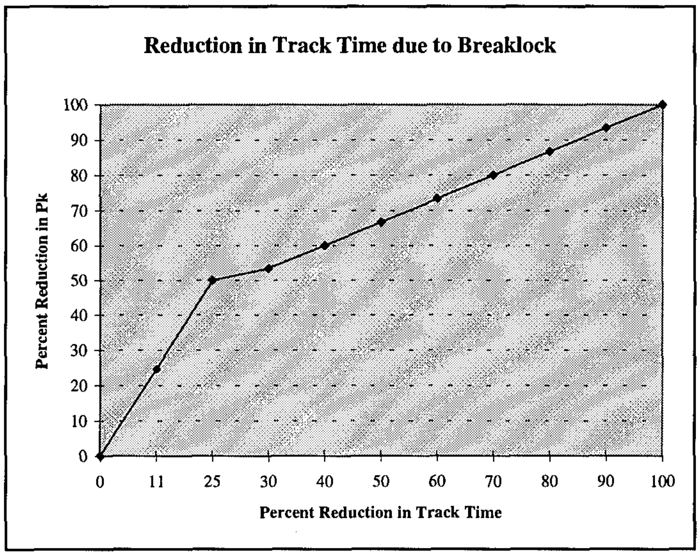

Figure 18 Reduction in Track Time due to Breaklock vs. Reduction in P_k Relationship

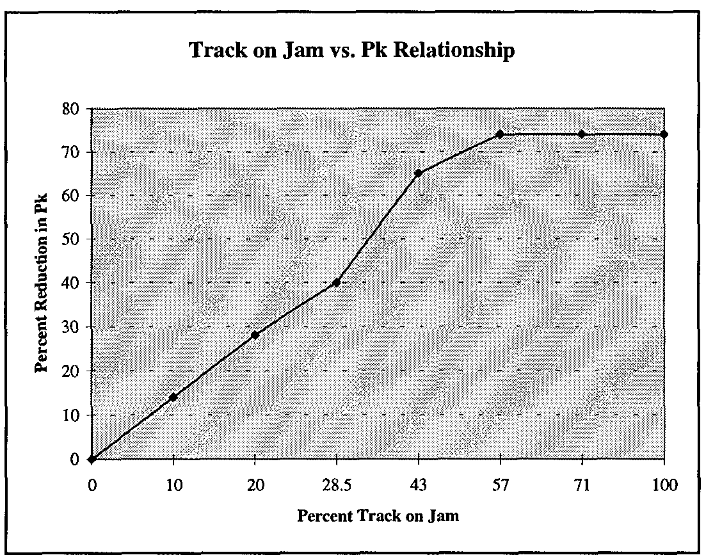

Figure 19 Track on Jam vs. Reduction in P_k Relationship

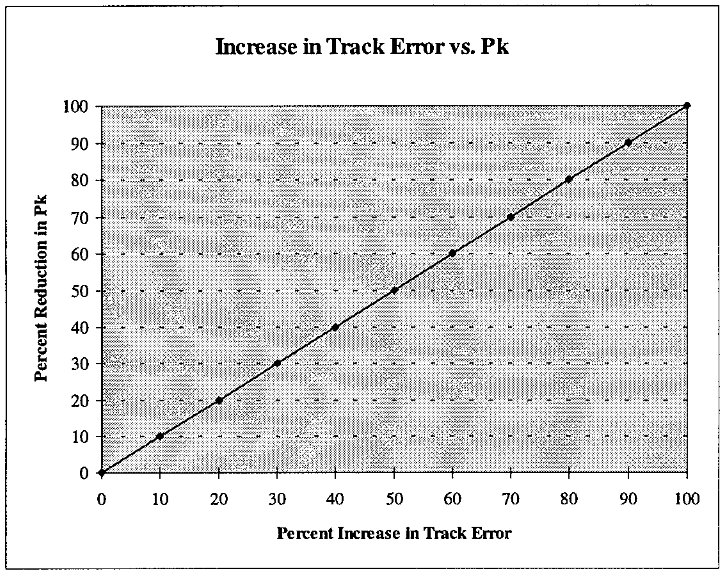

Figure 20 Increase in Track Error vs. Reduction in P_k Relationship

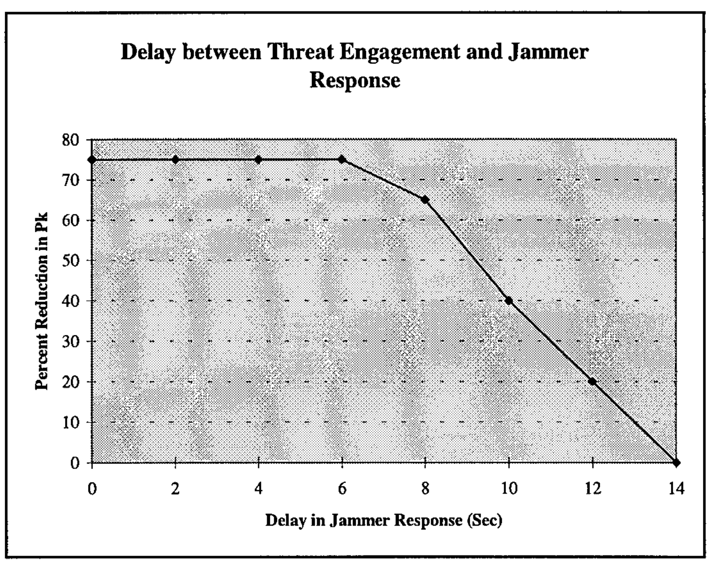

Figure 21 Delay between Threat Engagement and Jammer Response vs. Reduction in P_k Relationship

Based upon the fuzzy set definitions given in Table 6 and Table 7 and the relationships given in Figure 16 through Figure 21, the FAM Rule Bank is developed. The *Reduction Theorem* will be used to combine the information across the various MOFPs once it is transformed to the MOTA-level; therefore, the rules are written to relate each individual MOFP to the MOTA measure. The rules are shown below.

4.3.2.1 REDUCTION IN HITS RULE BANK

If *Reduction in Hits* is LO, *Reduction in* P_k is LO

If *Reduction in Hits* is LOMEDLO, *Reduction in* P_k is LOMEDLO

If *Reduction in Hits* is MEDLO, *Reduction in* P_k is MEDLO

If *Reduction in Hits* is HIMEDLO, *Reduction in* P_k is HIMEDLO

If *Reduction in Hits* is MED, *Reduction in* P_k is MED

If *Reduction in Hits* is LOMEDHI, *Reduction in* P_k is LOMEDHI

If *Reduction in Hits* is MEDHI, *Reduction in* P_k is MEDHI

If *Reduction in Hits* is HIMEDHI, *Reduction in* P_k is HIMEDHI

If *Reduction in Hits* is HI, *Reduction in* P_k is HI

4.3.2.2 REDUCTION IN GUIDANCE RULE BANK

If *Reduction in Guidance* is LO, *Reduction in* P_k is LOMEDLO

If *Reduction in Guidance* is LOMEDLO, *Reduction in* P_k is MED

If *Reduction in Guidance* is MEDLO, *Reduction in* P_k is MED

If *Reduction in Guidance* is HIMEDLO, *Reduction in* P_k is LOMEDHI

If *Reduction in Guidance* is MED, *Reduction in* P_k is MEDHI

If *Reduction in Guidance* is LOMEDHI, *Reduction in* P_k is MEDHI

If *Reduction in Guidance* is MEDHI, *Reduction in* P_k is HIMEDHI

If *Reduction in Guidance* is HIMEDHI, *Reduction in* P_k is HI

If *Reduction in Guidance* is HI, *Reduction in* P_k is HI

4.3.2.3 REDUCTION IN TRACK TIME RULE BANK

If *Reduction in Track Time* is LO, *Reduction in* P_k is LOMEDLO

If *Reduction in Track Time* is LOMEDLO, *Reduction in* P_k is MED

If *Reduction in Track Time* is MEDLO, *Reduction in* P_k is MED

If *Reduction in Track Time* is HIMEDLO, *Reduction in* P_k is LOMEDHI

If *Reduction in Track Time* is MED, *Reduction in* P_k is MEDHI

If *Reduction in Track Time* is LOMEDHI, *Reduction in* P_k is MEDHI

If *Reduction in Track Time* is MEDHI, *Reduction in* P_k is HIMEDHI

If *Reduction in Track Time* is HIMEDHI, *Reduction in* P_k is HI

If *Reduction in Track Time* is HI, *Reduction in* P_k is HI

4.3.2.4 TRACK ON JAM RULE BANK

If *Track on Jam* is LO, *Reduction in* P_k is LO

If *Track on Jam* is LOMEDLO, *Reduction in* P_k is MEDLO

If *Track on Jam* is MEDLO, *Reduction in* P_k is HIMEDLO

If *Track on Jam* is HIMEDLO, *Reduction in* P_k is LOMEDHI

If *Track on Jam* is MED, *Reduction in* P_k is MEDHI

If *Track on Jam* is LOMEDHI, *Reduction in* P_k is HIMEDHI

If *Track on Jam* is MEDHI, *Reduction in* P_k is HIMEDHI

If *Track on Jam* is HIMEDHI, *Reduction in* P_k is HIMEDHI

If *Track on Jam* is HI, *Reduction in* P_k is HIMEDHI

4.3.2.5 INCREASE IN TRACK ERROR RULE BANK

If *Increase in Track Error* is LO, *Reduction in* P_k is LO

If *Increase in Track Error* is LOMEDLO, *Reduction in* P_k is LO

If *Increase in Track Error* is MEDLO, *Reduction in* P_k is LO

If *Increase in Track Error* is HIMEDLO, *Reduction in* P_k is LO

If *Increase in Track Error* is MED, *Reduction in* P_k is LO

If *Increase in Track Error* is LOMEDHI, *Reduction in* P_k is LO

If *Increase in Track Error* is MEDHI, *Reduction in* P_k is LO

If *Increase in Track Error* is HIMEDHI, *Reduction in* P_k is LOMEDLO

If *Increase in Track Error* is HI, *Reduction in* P_k is LOMEDLO

4.3.2.6 DELAY TIME RULE BANK

If *Delay Time* is LO, *Reduction in* P_k is MEDHI

If *Delay Time* is LOMEDLO, *Reduction in* P_k is MEDHI

If *Delay Time* is MEDLO, *Reduction in* P_k is MEDHI

If *Delay Time* is HIMEDLO, *Reduction in* P_k is MEDHI

If *Delay Time* is MED, *Reduction in* P_k is LOMEDHI

If *Delay Time* is LOMEDHI, *Reduction in* P_k is HIMEDLO

If *Delay Time* is MEDHI, *Reduction in* P_k is LOMEDLO

If *Delay Time* is HIMEDHI, *Reduction in* P_k is LO

If *Delay Time* is HI, *Reduction in* P_k is LO

4.3.3 FAM Implementation

Due to the lack of a diverse group of information sources, that would have provided conflicting information in the development of the FAM rules, the task of resolving conflicts is not required. However, the conflict resolution method might come into play in the refinement of the FAM rule bank as more information is gathered on the system-under-test. Additionally, the task of melding the rules derived from numerical data with those derived from expert opinion is limited to examining the rules listed in the sections above and verifying that they make intuitive sense. Thus, the next step after construction of the FAM, is FAM implementation. C code has been developed to perform the actions of transforming the COMMFFY derived from the previous steps into a COMMFFY at the MOTA level. The flow-chart for that implementation code is shown in Figure 22.

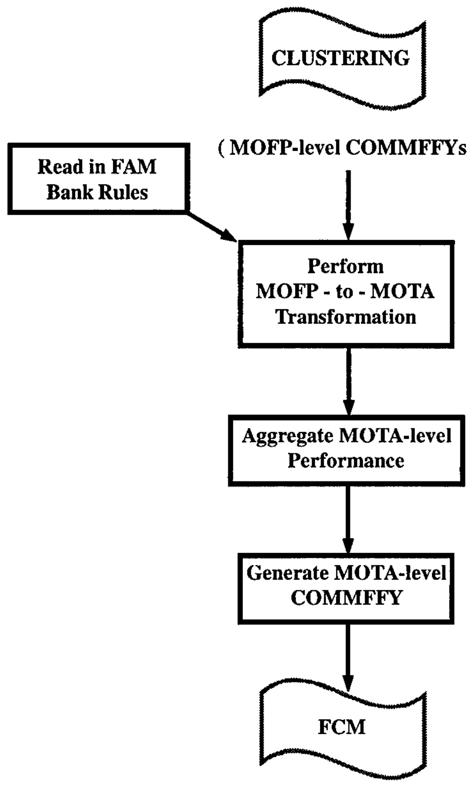

Figure 22 FAM Implementation Code Flow Chart

Figure 22 illustrates that the FAM is used as follows. Beginning with the MOFP-level COMMFFYs generated by Clustering Methodology, the FAM Rule Banks are read into system memory, giving the transformation from the MOFP-level BMFs to the MOTA-level BMFs. The transformation from MOFP-level to MOTA-level is then performed using these FAM Rule Banks. Once the MOFP-to-MOTA transformation has occurred, using the Reduction Theorem, the performance at the MOTA-level for each MOFP is combined using a fuzzy intersection operation. The end product of the aggregation is system performance at the MOTA-level for each logical division of the system performance (e.g., in the testbed case, the logical division is system performance against individual threat systems). These aggregated functions are still in BMF format, so the final step is to generate the MOTA-level COMMFFY from these degrees of membership of each BMF. These MOTA-level COMMFFYs are the output of this phase of the *Intelligent Hierarchical Decision Architecture*. They will be used as input to the FCM in the next phase.

4.4 TESTBED CASE RESULTS

The FAM described in Sections 4.2 and 4.3 has been successfully implemented in a C-language program, based upon the flow chart shown in Figure 22. This code was executed using the testbed case data and the resulting MOTA COMMFFYs are shown in Figure 23 through Figure 26. The input to the FAM was a COMMFFY for each MOFP and for each threat. The FAM served to aggregate the information across the various MOFPs and provide an output at the MOTA-level for each threat system.

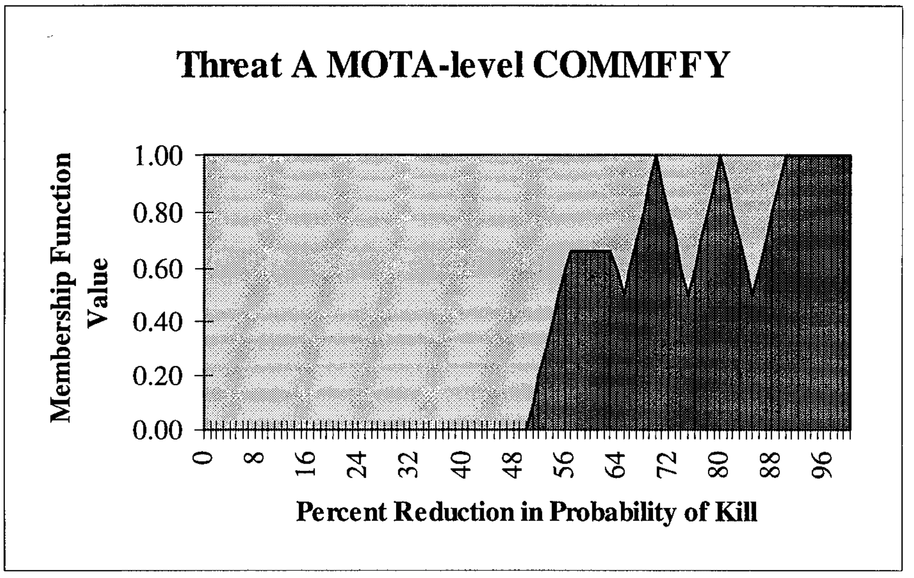

Figure 23 Threat A MOTA-level COMMFFY

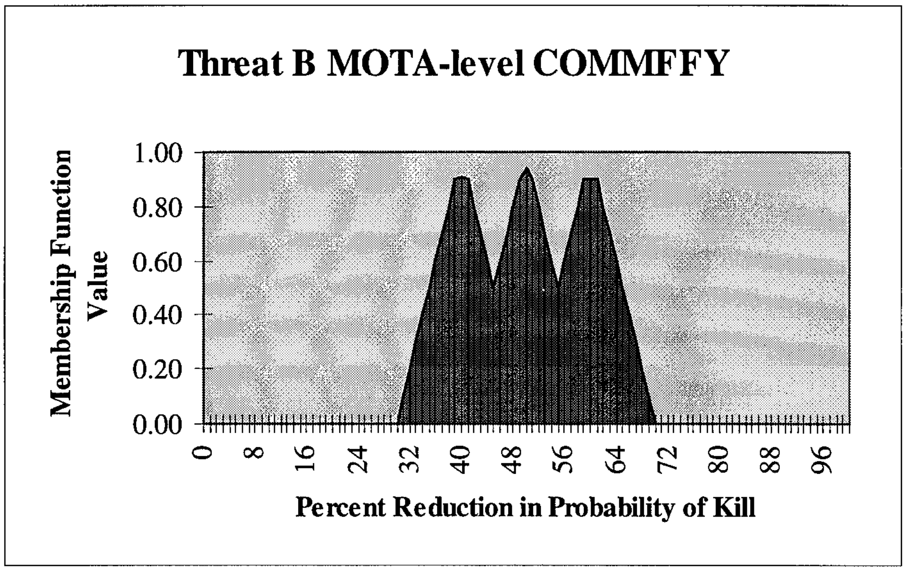

Figure 24 Threat B MOTA-level COMMFFY

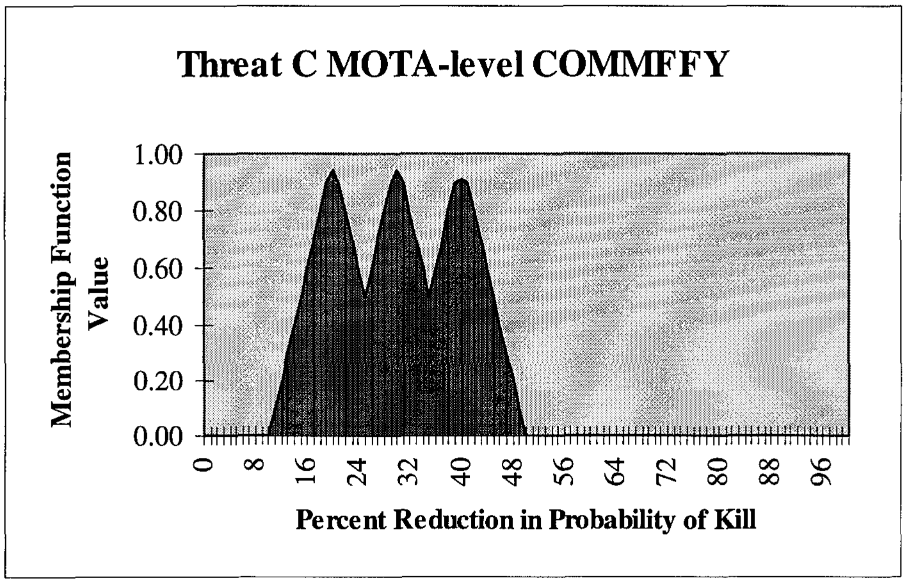

Figure 25 Threat C MOTA-level COMMFFY

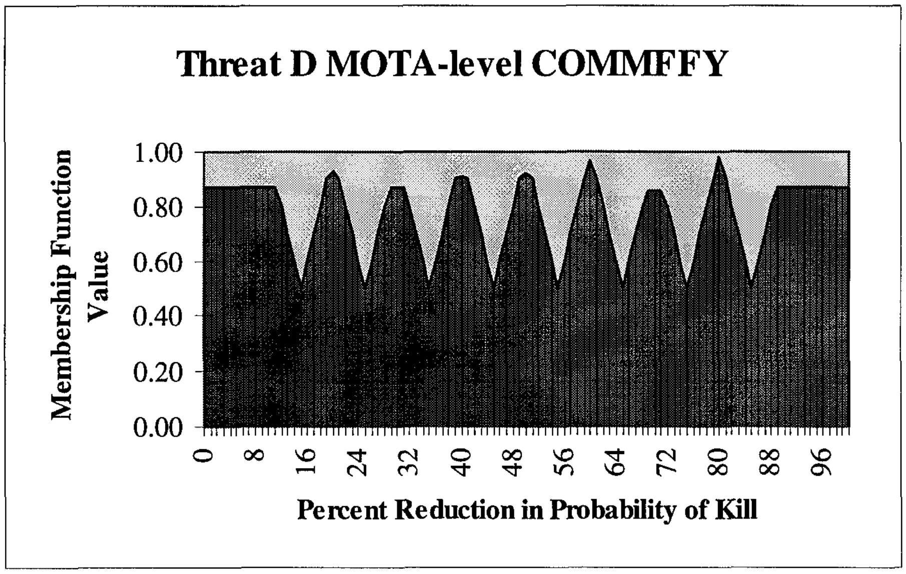

Figure 26 Threat D MOTA-Level COMMFFY

From these COMMFFYs, it can be seen that the original trend in the data has carried through the analysis and aggregation process to this point. The Threat A COMMFFY is substantially above the 50% requirement. The Threat B COMMFFY is concentrated around the 50% requirement. The Threat C COMMFFY is concentrated below the 50% requirement. Finally, the Threat D COMMFFY shows a wide dispersion across the domain. This gives a satisfying intuitive feel that the analysis hierarchy is manipulating the data as would have been expected. These MOTA-level COMMFFYs will now serve as the input to the next phase of the hierarchy -- the *Fuzzy Cognitive Map* which will adjust the COMMFFY to take into consideration factors that could not be tested or controlled during the testing effort.

4.5 CONTRIBUTION

The development of this section of the decision hierarchy has provided a means for working with a Fuzzy Associative Memory whose input and output are functions, here in the form of COMMFFYs, rather than individual data points. This is particularly useful for those applications where data can only be gathered on one aspect of a system's performance at a time, such as is usually the case during testing efforts. During a test, one experimental design can be set up to collect data on a given aspect of the system's performance, then another experimental design is set up to collect data on another aspect, and so on. Taking one reading from each of those experimental design set-ups and combining them into an artificially-derived data set that would be required to use the currently-available fuzzy tools, is a misuse of the collected data, and could lead to misleading results. This artificial data-set would be required to analyze data using the tools provided in such popular fuzzy logic packages as MATLAB's Fuzzy Logic Toolbox. This research effort provides an improvement to the current fuzzy logic analysis toolbox -- the ability to work with input and output functions rather than individual data points.

CHAPTER FIVE

FUZZY COGNITIVE MAP

---

5. FUZZY COGNITIVE MAPS

Political scientist, Robert Axelrod introduced cognitive maps in the mid-1970's as a method of representing political and social scientific knowledge [30]. The maps he proposed were signed digraphs with nodes representing variable concepts and edges representing causal connections. Since that time, Bart Kosko has extended the theories put forth by Axelrod to develop Fuzzy Cognitive Maps (FCMs), allowing the causal relations between concepts to take on fuzzy relations. Other researchers have taken Kosko's work and extended it to look at such diverse topics as: allowing non-linear relations between concepts, determining weighting schemes for the provided information, building hierarchical FCMs to consider relations at various conceptual levels, and using FCMs for failure mode analyses. Fuzzy Cognitive Maps will be used within the *Intelligent Hierarchical Decision Architecture* to modify the system behavior, measured at the task accomplishment level, considering factors that cannot be tested or controlled during the testing phase. The FCM theories put forth by Kosko and others, will be extended in this research to allow FCMs to be used as a modifier to system performance measures. The current state-of-the-art in FCM use is to use them as a binary predictor or conceptual presence indicator. Appendix C provides an overview of the basic theories of FCMs developed by Kosko and other researchers. This chapter begins with a brief overview of the current uses and advances in the FCM arena. After that introduction, the chapter outlines the FCM's use in the *Intelligent Hierarchical Decision Architecture*, provides an illustration of the *Intelligent Hierarchical Decision Architecture's* FCM Methodology, then finally, closes with a discussion of how this use will advance the state of knowledge in the FCM arena.

5.1 DRAWING INFERENCES USING FCMS

Drawing inferences based upon the information contained within an FCM is a simple matter of repeated matrix multiplication and thresholding operations. The result is a "walk" through the states of the FCM until a limit state or cycle is reached; indicating the final stable state(s) of the system resulting from the original input condition. Based upon the limit cycle existence, the FCM has proven to be an ideal tool to answer "what if" questions. Once the FCM has been constructed, it can be used to draw inferences on what concepts contained in the FCM will result if one of the concepts is activated. For example, a common example is the FCM built to illustrate the interrelationships between various aspects of South African politics. This FCM was developed by Kosko based upon a syndicated article by Walter Williams and is shown in Appendix C. This research will extend the use illustrated there, to show that FCMs can also be used to adjust input membership functions as a result of the influence of the factors included in the FCM.

5.2 CURRENT FCM USES AND ADVANCES

Bart Kosko introduced the concept of Fuzzy Cognitive Maps in his seminal 1986 work [93], based on the original work done by Axelrod [30] in the area of Cognitive Maps to model social and political science phenomena. Kosko developed the fuzzy causal algebra as a means for quantifying the edge values in terms of fuzzy sets. Since then, Kosko's work has been extended in a number of ways. This section contains a brief review of the state of the art of FCM construction and use, outlined in the current literature in the field. It is included to give the reader an appreciation of where the theory stands and how the current research will extend the theoretical state of the art in the FCM arena.

Taber and his colleagues provided a mechanism for quantifying the credibility weighting scheme discussed in Equation C-4 of Appendix C. They assumed that a knowledge pool contains a healthy measure of concurrence around the central concepts [31]. Based upon this assumption, they developed a quality measure for the knowledge base developed by each expert based upon the Hamming distance between inferences invoked by each FCM to the same input conditions. This quality measure was used to weight the expert's opinion based upon his closeness to others'. To test the weighting mechanism, they included FCMs generated by "synthetic experts" into the process. The FCMs from the synthetic experts were FCMs from the real experts contaminated with noise to degrees varying from 0-100% [32]. Their methodology was able to separate the actual experts from the synthetic experts. Additionally, they assigned weighting factors illustrating a roll-off effect to minimize the risk associated with using information contributed by sources demonstrating a significant deviation from the standard response.

Zhang and Chen [92] introduced a trivalent logic mechanism, allowing negative, positive, and neutral causal relations and developed what they termed a NPN (negative-positive- neutral) calculus. This contribution eliminated the need, suggested originally by Kosko, of replacing each concept quantity with two concepts: the quantity and its associated dis-quantity (thus doubling the size of the FCM). They developed a logic framework to deal with the trivalent values and extended standard fuzzy logic inferencing to handle the new logic system. Using these mechanisms, they showed a method of combining expert opinion that takes into consideration the effects of different paths to a result rather than simply adding augmented matrices together, which has the potential of "canceling out" opposing opinions.

The limitation of the FCMs proposed to this point has been that they only allow representations of relationships that are linear, time-invariant, and non-conditional [33].

These limitations were addressed by Masafumi Hagiwara in his work on Extended Fuzzy Cognitive Maps. There, he introduced E-FCMs including weights with nonlinear membership functions, conditional weights, and time delay weights. Using the FCM originally introduced by Zhang and Chen, dealing with public health issues, the relations were modified to include non-linear functions, time delays, and conditional weights. Hagiwara claimed to show that the resulting inference behavior of the E-FCM was radically different (although the results illustrated in the referenced article do not seem as dramatic as the author claims) from that using the standard FCM.

FCMs discussed thus far have been developed based upon information supplied by experts in the form of expert opinions. The mechanism for building FCMs based upon data had not been discussed or considered. Researchers at Florida Institute of Technology have developed a mechanism through which FCMs can be built from user-supplied data in place of, or as a supplement to expert opinion [34]. This is done through converting numerical vectors containing the data sets into fuzzy sets through an α-level cut and an interpolation procedure to project the data elements proportionally into the [0, 1] interval. To determine the positive or negative causality values, a similarity relation between the vectors is calculated. Finally, the direction of the causality is determined by expert opinion. By following this procedure an FCM can be developed based upon measured data, rather than relying solely on expert opinion. The introduction of this work suggests a means of collecting information from both experts and numerical data sources, then using the expert weighting/grading scheme suggested by Taber and his associates to determine relative credibility of the various sources.

Finally, Australian researchers at the University of Melbourne have begun to explore the use of FCMs in a multi-layered approach [35]. Each layer of the FCM would be responsible for making inferences at a specific information level, then providing the information as input to the next higher inference level. They suggest a goal level which describes desired outcomes linked by common criteria, a context level providing a second level of generalization, and finally the lowest FCM level providing a specialization level.

This work parallels, to a certain extent the work being done here, in that various levels of decision-making are segregated into separate tasks which feed into each other. However, the work done by the Australian researcher uses FCMs for each level, whereas, the *Intelligent Hierarchical Decision Architecture* uses other tools including Fuzzy Associative Memories, Fuzzy Clustering, and Dempster-Shafer Theory to accomplish the task.

5.3 THE IHDA FUZZY COGNITIVE MAP

The FCM within the *Intelligent Hierarchical Decision Architecture* will be used to modify the measured system performance at the task accomplishment level to account for factors that are not controllable or testable in the OT&E testing effort. For example, consider the Jammer-X system. During the testing effort, the effects of using the system are measured and quantified by measurements taken at the functional performance level and these measures are subsequently aggregated using a FAM to yield a COMMFFY at the task accomplishment level. There are various factors, including such things as *tactics*, the *number of other aircraft* in the engagement scenario, the *mix of threats* involved in the scenario, *weather conditions*, etc. Which cannot be measured or controlled during testing, but are known to have an impact on the system performance. An FCM will be developed and used to adjust the testing-based system performance measurement to account for these other factors.

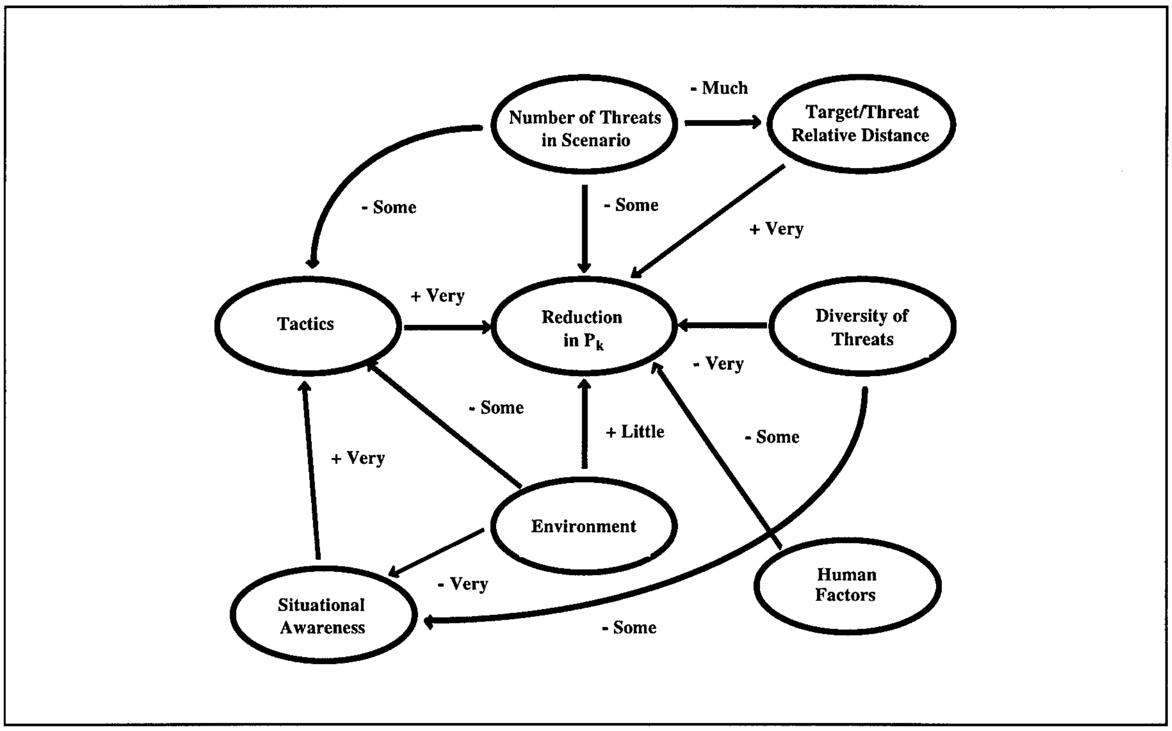

Figure 27 Electronic Combat System Global FCM

The adjustment to the MOTA-level COMMFFY will be accomplished by first, defining the factors that are potential contributors to task-level system performance that were not tested during OT&E. Once the factors are identified, they will be used to construct a FCM. The FCM will provide information on the causal relationships between the included factors.

For the testbed case, the information for the individual FCMs was gathered from surveys conducted on HQ AFOTEC personnel with previous operational experience with EC systems [36]. Each individual provided a FCM including the concepts and relationships he felt were important to system performance. The FCMs were combined using the knowledge combination function suggested by Pelaez, described in Appendix C. The resulting FCM is shown in Figure 27. With the global FCM developed, it will be used to infer the change to the MOTA-level COMMFFY resulting from the FCM factors, using the methodology described below.

5.3.1 THE IHDA FCM PERFORMANCE ADJUSTMENT METHODOLOGY

The performance-adjustment methodology depends on a melding of the contributions of many of the researchers described in Section 5.2 as well as extensions of their methodologies to solve the unique application issues posed by using an FCM to adjust system performance measures. The methodology is enumerated below, with an illustration provided in the Section 5.4.

1. Collect individual FCMs from system experts and meld into a single, global FCM using the knowledge combination operator proposed by Pelaez, described in Appendix C [97].

2. Using the global FCM, ignore the linguistic tags associated with each link and treat the FCM as a trivalent map. Using this global, trivalent map, define its associated adjacency matrix.

3. From the adjacency matrix definition and a vector representing the activation of each concept, use the matrix multiplication/thresholding operation with the +1/0/-1 clamping values to determine the limit cycle or limit state associated with each concept. The limit cycle or limit state for each concept defines the active concepts that are subsequently used to limit the number of variables considered in defining paths from the *policy variable* to the *value variable*[7] [93].

[7] These terms were introduced by Axelrod in his use of Cognitive Maps as a political science tool. The policy variable is the active concept being considered, while the value variable is the central concept whose value is being examined for changes due to the application of the active concept. In the testbed case, the

4. Using the active concepts defined from the adjacency matrix multiplication/thresholding operation, define all possible paths from each *policy variable* to the *value variable* using the active concepts as the allowable links.

5. Once all the possible paths, *l*, are defined, use the pre-defined linguistic tag relative strengths to define the *indirect effect* using the minimum operation, as

$$I_l(C_i, C_j) = \min\{e(C_p, C_{p+1}) : (p, p+1) \in (i, k_1^l, k_2^l, \ldots, k_{n_1}^l, j)\} \qquad \textbf{(5-1)}$$

6. Considering all the possible paths from a given *policy variable* to the *value variable*, calculate the *total effect* using the maximum operation, as

$$T(C_i, C_j) = \max_{1 \le l \le m} I_l(C_i, C_j) \qquad \textbf{(5-2)}$$

7. List the *total effects* in rank-order from the most negative impact on the system performance measure to the most beneficial impact on the system performance measure.

8. Calculate the adjustment to the MOTA-level COMMFFY generated by the most negative impact (worst case) and the most positive impact (best case) using the resulting total effect linguistic tag as a modifying hedge. This adjustment will be performed based upon the use of linguistic hedges [13] which serve to modify a fuzzy set. The adjustment will be based upon a parameterized exponential factor applied to the MOTA-level COMMFFY membership function values, as shown in (5-3) for a positive hedge adjustment and (5-4) for a negative hedge adjustment.

---

policy variable will be each concept in the FCM that are examined in turn, to examine the result in Reduction in P_k, the value variable.

$$\mu_{+Adj} = \mu^{\frac{1}{k}} \quad (5\text{-}3)$$

$$\mu_{-Adj} = \mu^{k} \quad (5\text{-}4)$$

The values of k are chosen to provide an adequate adjustment to the COMMFFY.

The worst case and best case performance results are carried forward to the Aggregation Methodology step to establish a system performance bound at the task-level based upon both measured test performance and other factors that will affect system performance that could not be tested.

5.4 TESTBED CASE RESULTS

Step 1: Define the global FCM.

A survey, including a Fuzzy Cognitive Map Background Primer, has been distributed to Electronic Combat system experts within the Air Force, Department of Defense, and commercial contractor communities. Results, in the form of FCMs for the testbed case, were received and the global FCM shown in Figure 27 resulted.

Step 2: Define the global FCM's adjacency matrix.

From Figure 27, define the following concept numbers to simplify the FCM to adjacency matrix conversion:

C_1: Number of Threats in Scenario

C_2: Target/Threat Relative Distance

C_3: Tactics

C_4: Diversity of Threats

C_5: Situational Awareness

C_6: Environment

C_7: Human Factors

C_8: Reduction in P_k

Then, the adjacency matrix becomes:

$$F = \begin{bmatrix} 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 3: Determine limit state associated with each concept.

This step involves consideration of each concept, C_1 through C_7, in turn as a *policy variable*, while C_8 (Reduction in P_k) remains the *value variable* throughout. To accomplish this, the multiplication/thresholding operation is performed using the +1/0/-1 clamping values, with each of the concepts C_1 - C_7 as a starting point. The operation is illustrated below with C_1, the remainder follow in the same manner.

$$C_1 = [\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ ]$$

$$C_1F = [\ 0\ -1\ -1\ 0\ 0\ 0\ -1\ ]$$

$$\rightarrow [\ 1\ -1\ -1\ 0\ 0\ 0\ -1\ ] = C_{1a}$$

$$C_{1a}F = [0\ -1\ -1\ 0\ 0\ 0\ -3\ ]$$

$$\rightarrow [\ 1\ -1\ -1\ 0\ 0\ 0\ -1\ ] = C_{1LS}$$

The remaining concepts are used similarly in the multiplication/thresholding operation to yield the following limit states.

$$C_{2LS} = [\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ ]$$

$$C_{3LS} = [\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ ]$$

$$C_{4LS} = [\ 0\ 0\ -1\ 1\ -1\ 0\ 0\ -1\ ]$$

$$C_{5LS} = [\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ ]$$

$$C_{6LS} = [\ 0\ 0\ -1\ 0\ -1\ 1\ 0\ 0\ ]$$

$$C_{7LS} = [\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ -1\ ]$$

Step 4: Define possible paths from each policy variable to the value variable.

Considering the allowable nodes, defined by the limit states, the possible paths can be defined. Again using C_1 as an example, the allowable nodes are C_1, C_2, C_3, and C_8. Using only these nodes and their connecting edges, the following allowable paths from C_1 (*policy variable*) to C_8 (*value variable*) are defined:

$$C_1 \rightarrow C_8$$

$$C_1 \rightarrow C_2 \rightarrow C_8$$

$$C_1 \rightarrow C_3 \rightarrow C_8$$

The remainder of the possible paths are defined similarly, with the following results.

$$C_2 \rightarrow C_8$$

$$C_3 \rightarrow C_8$$

$$C_4 \rightarrow C_8$$
$$C_4 \rightarrow C_5 \rightarrow C_3 \rightarrow C_8$$

$$C_5 \rightarrow C_3 \rightarrow C_8$$

$$C_6 \rightarrow C_8$$
$$C_6 \rightarrow C_3 \rightarrow C_8$$
$$C_6 \rightarrow C_5 \rightarrow C_3 \rightarrow C_8$$

$$C_7 \rightarrow C_8$$

Step 5: Define indirect effects.

Using the possible paths defined in Step 4, the *indirect effect* is calculated. First, the relative strengths of the linguistic labels used to describe the relationships are defined. For the testbed case, assume that $\{little \leq some \leq much \leq very\}$. The minimum operation is carried out in accordance with these relative strengths. The sign of the label is brought along to describe the direction of the relationship; however, it is not considered in the minimization operation. The result for the paths defined in Step 4 are shown below.

$$C_1 \rightarrow C_8 \Rightarrow \min\{\textit{-some}\} = \textit{-some}$$

$$C_1 \rightarrow C_2 \rightarrow C_8 \Rightarrow \min\{\textit{-much}, \textit{+very}\} = \textit{-much}$$

$$C_1 \rightarrow C_3 \rightarrow C_8 \Rightarrow \min\{\textit{-some}, \textit{+very}\} = \textit{-some}$$

$$C_2 \rightarrow C_8 \Rightarrow \min\{\textit{+very}\} = \textit{+very}$$

$$C_3 \rightarrow C_8 \Rightarrow \min\{\textit{+very}\} = \textit{+very}$$

$$C_4 \rightarrow C_8 \Rightarrow \min\{\textit{-very}\} = \textit{-very}$$

$$C_4 \rightarrow C_5 \rightarrow C_3 \rightarrow C_8 \Rightarrow \min\{\textit{-some}, \textit{+very}, \textit{+very}\} = \textit{-some}$$

$$C_5 \rightarrow C_3 \rightarrow C_8 \Rightarrow \min\{\textit{+very}, \textit{+very}\} = \textit{+very}$$

$$C_6 \rightarrow C_8 \Rightarrow \min\{\textit{+little}\} = \textit{+little}$$

$$C_6 \rightarrow C_3 \rightarrow C_8 \Rightarrow \min\{\textit{-some}, \textit{+very}\} = \textit{-some}$$

$$C_6 \rightarrow C_5 \rightarrow C_3 \rightarrow C_8 \Rightarrow \min\{\textit{-very}, \textit{+very}, \textit{+very}\} = \textit{-very}$$

$$C_7 \rightarrow C_8 \Rightarrow \min\{\textit{-some}\} = \textit{-some}$$

Step 6: Define total effects.

The *total effect*, described intuitively as the strongest of the weakest links, is now defined by taking the maximum of the path link strengths within each concept. The results are shown below.

$$T(C_1, C_8) = \max\{\textit{-some}, \textit{-much}\} = \textit{-much}$$

$$T(C_2, C_8) = \max\{+very\} = +very$$

$$T(C_3, C_8) = \max\{+very\} = +very$$

$$T(C_4, C_8) = \max\{-very, -some\} = -very$$

$$T(C_5, C_8) = \max\{+very\} = +very$$

$$T(C_6, C_8) = \max\{+little, -some, -very\} = -very$$

$$T(C_7, C_8) = \max\{-some\} = -some$$

Step 7: Rank order the effects.

The effects shown in Step 6 are now rank-ordered for their importance in affecting the value variable. Using the same linguistic importance ranking as shown in Step 5, but now including the sign of the effect, the following results.

Most Negative Impact: C_4 and C_6; *-very*

C_7; *-some*

C_1; *-much*

Most Positive Impact: C_2, C_3, and C_5; *+very*

Step 8: Adjust the MOTA-level COMMFFY.

The final step is to use the most negatively and most positively impacting effects to adjust the MOTA-level COMMFFY generated from the previous phases of the *Intelligent Hierarchical Decision Architecture*. Using the linguistic tags as a modifying hedge, the parameter, *k*, to be used for the COMMFFY adjustment is defined. Based upon the common use of linguistic modifiers in the fuzzy literature [82], a value of $k=2$ is chosen to be associated with the label *very*. Table 8 gives the definition for the values of *k* chosen for the other linguistic labels in this testbed case.

Table 8 Linguistic Tag/Hedge Parameter Relationships

Linguistic Tag	Value of *k*
very	2.0
much	1.5
some	1.25
little	1.1

Using these values for the exponential parameter, the adjustment on a sample MOTA-level COMMFFY is illustrated in Figure 28 through Figure 31. The best-case and worst-case adjustments to all the testbed case task-accomplishment level COMMFFYs (using only the *+very* and *-very* adjustments) are given in Appendix F.

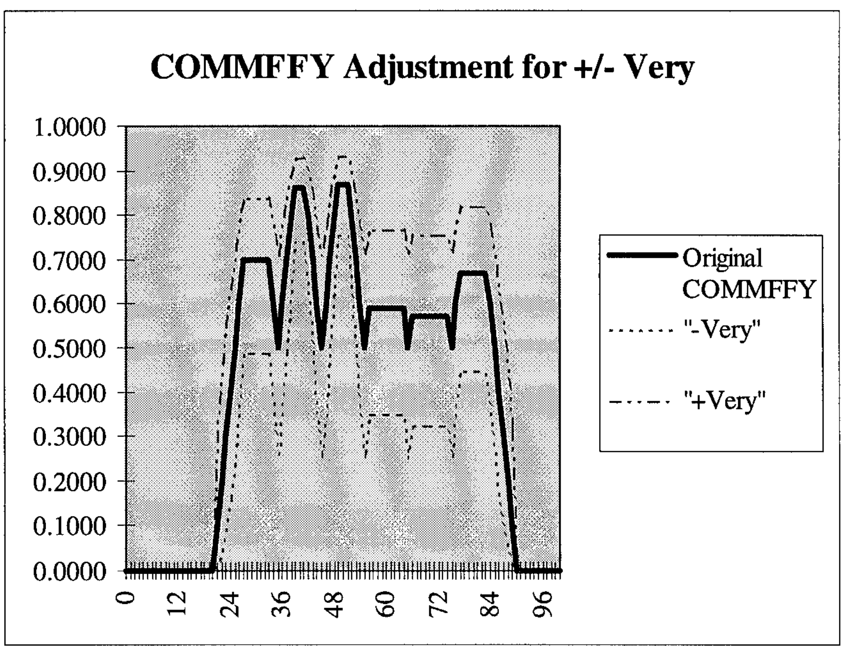

Figure 28 COMMFFY Adjustment as a Result of Linguistic Hedge "Very" ***(k=2.0)***

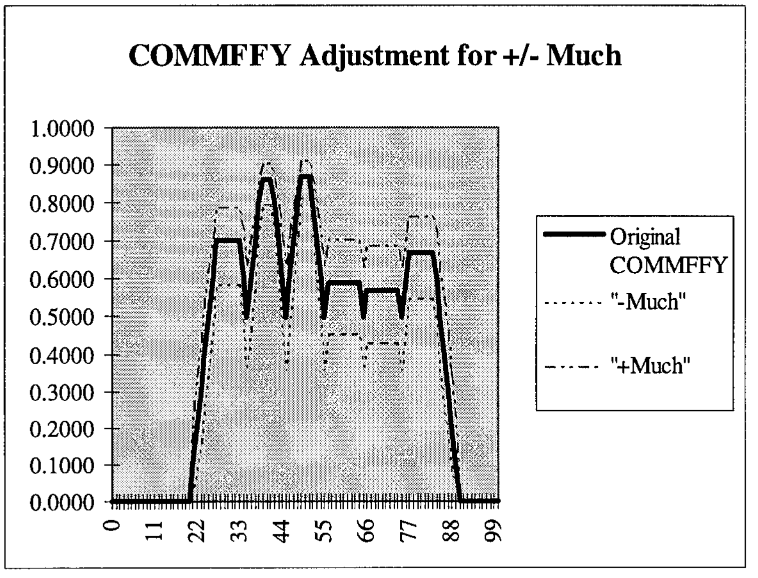

Figure 29 COMMFFY Adjustment as a Result of Linguistic Hedge "Much" ***(k=1.5)***

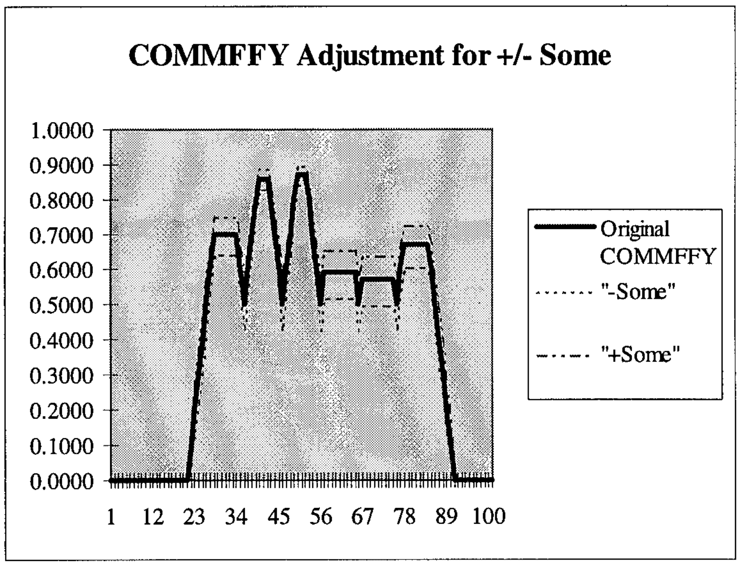

Figure 30 COMMFFY Adjustment as a Result of Linguistic Hedge "Some" ***(k=1.25)***

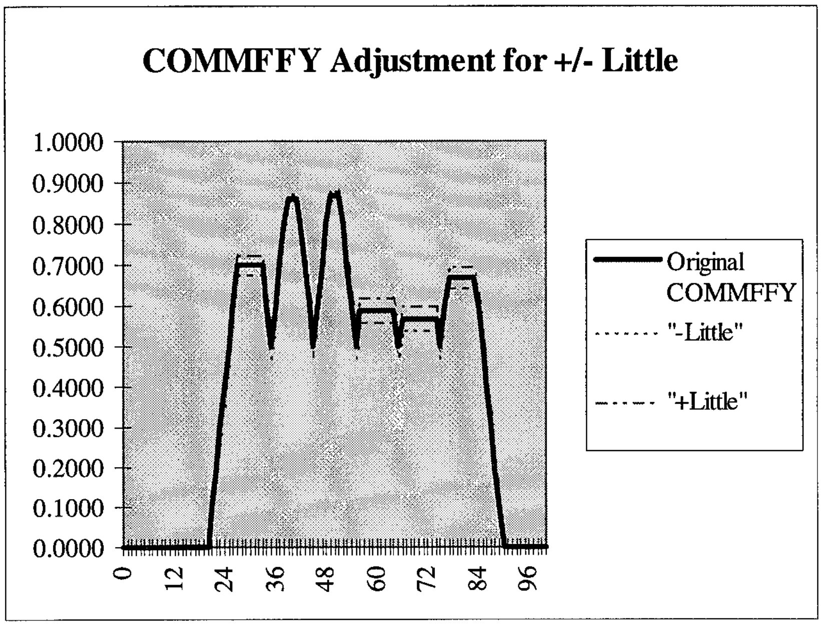

Figure 31 COMMFFY Adjustment as a Result of Linguistic Hedge "Little" (*k=1.1*)

Visually, these adjustments are very appealing. Although the aggregation method does not involve a defuzzification step, one can be used to determine if the adjustment procedure is working correctly by defuzzifying the adjusted COMMFFY and looking at the resulting values. Table 19 shows that when the *Center of Area Defuzzification Method* is used, the crisp value moves in the expected direction, based upon the change in the COMMFFY, however, the change is very slight. If a different defuzzification method is chosen, such as the *Method of Heights Defuzzification Method*, with an α-cut value of 0.5, a more satisfying result is seen (i.e., the crisp value shows more movement relative to the original value). Depending on the amount of adjustment desired in the final, crisp result of the methodology, the exponential parameter and the defuzzification method can be adjusted accordingly.

Table 9 Adjusted Defuzzification Values

	COA Defuzzification Method			MOH Defuzzification Method		
	"+" Adj	No Adj	"-" Adj	"+" Adj	No Adj	"-" Adj
Very	54.5716	54.0968	53.0683	52.9821	46.8112	11.8162
Much	54.4175	54.0968	53.5922	52.1435	46.8112	26.5651
Some	54.2911	54.0968	53.8475	51.1641	46.8112	38.8920
Little	54.1857	54.0968	53.9979	51.3308	46.8112	47.0258

5.5 CONTRIBUTION

All of the current uses of FCMs, including the FMEA FCM contributed by Bowles and Pelaez, are limited to activating a concept and seeing which concepts result. Although the FMEA work used the linguistic tags to describe the extent of the result, it stops there. This work extends the use of the FCM-drawn inference to modify test-derived results. Thus, taking the FCM a step further than previous work. Additionally, using the result of the multiplication/thresholding operation as a mechanism to limit the scope of subsequent steps in the methodology, allows larger and more complex FCMs to be considered than would have been practical without this step.

This work also represents an advance in the state of the art in the test and evaluation arena. Current analysis methods are limited to summarization techniques based upon statistical concepts and offer no mechanism for extending results beyond those observed in the laboratory or test range. However, the aim of Operational Test and Evaluation is to simulate an operational environment, which necessarily included factors that are not quantifiable or controllable. Therefore, a mechanism is desperately needed which will allow information that is commonly known to affect the system performance to be brought into the decision-making process. The methodology outlined here will provide that mechanism.

CHAPTER SIX

AGGREGATION METHODOLOGY

6. AGGREGATION METHODOLOGY

The final stage of processing in the *Intelligent Hierarchical Decision Architecture* is that of aggregating the system performance across all the logical divisions of the system performance to form an overall assessment of system performance. In the testbed case, this involves taking the system performance demonstrated against each individual threat system and aggregating it across all the threat systems to develop an assessment of the jamming system's overall performance. There are several options available for performing this aggregation. After a brief review of the current literature's treatise on aggregation operators, a discussion of the Dempster-Shafer Theory of Evidential Reasoning is turned to. It will be shown that the Dempster-Shafer theory, originally developed by Dempster in the early 1960's [37] and extended by Shafer in the mid-1970's [38], will provide the ideal mechanism for combining the system performance information into its final aggregated form -- providing a probabilistic bound on the performance at the task-accomplishment level. This system performance bound can then be provided to the decision-maker, as a truly meaningful piece of information to aid in the decision-making process.

6.1 CURRENT AGGREGATION METHODS

Aggregation operators, used to combine two or more fuzzy sets into a single set are represented by a function h defined by [13]

$$h\colon [\,0,1]^n \rightarrow [0,1], \;\; n \geq 2 \tag{6-1}$$

where n is the number of aggregation sources. The following properties are given which must be satisfied in order for an operator to be considered an aggregation operator.

Definition: A function, h, is called an aggregation operator if it satisfies the following properties.

(AP1) $h(0, 0, ..., 0) = 0$ and $h(1, 1, ..., 1) = 1$, that is, h is bounded;

(AP2) For any pair $(a_i | i \in \mathbb{N}_n)$ and $(b_i | i \in \mathbb{N}_n)$, where $a_i \in [0,1]$ and $b_i \in [0,1]$, if $a_i \geq b_i$ for all $i \in \mathbb{N}_n$, then $h(a_i | i \in \mathbb{N}_n) \geq h(b_i | i \in \mathbb{N}_n)$, that is, h is *monotonic nondecreasing* in all its arguments;

(AP3) h is a *continuous function*

(AP4) h is a *symmetric* function in all its arguments, that is, $h(a_i | i \in \mathbb{N}_n) = h(a_{p(i)} | i \in \mathbb{N}_n)$, for any permutation p on $\mathbb{N}_n$.

where $\mathbb{N}_n$ denotes the set of all integers from 1 through the value of the subscript; that is, $\mathbb{N}_n = \{1, 2, ..., n\}$.

The current literature on decision-making aggregation methods focus on two classes of methodologies: aggregation of sensor information and preference ordering among alternatives. Both categories share similar traits, including the decision of which aggregation operator is most appropriate for their task. The discussion that follows highlights one work from each class to illustrate the work that has been accomplished in this area. Following that discussion, Section 6.1.1 discusses other options that might be available for the aggregation task. Section 6.2 and Appendix D include discussions of the Dempster-Shafer theory and an examination of its applicability to needs of this work. Finally, in Section 6.3, the *Intelligent Hierarchical Decision Architecture's Aggregation Methodology* is discussed.

Loskiewicz-Buczak and Uhrig [39] use information fusion based upon fuzzy sets operations to analyze vibration signatures for fault detection applications. In their work, they conclude that human decisions and evaluations almost always show some degree of compensation. They suggest that the generalized mean operator very closely matches the human decision making process. The generalized mean is defined by

$$g(x_1, x_2, \ldots, x_n; p, w_1, w_2, \ldots, w_n) = \left(\sum_{i=1}^{n} w_i x_i^p\right)^{1/p} \qquad \textbf{(6-2)}$$

The rate of compensation of this operator is controlled by the parameter, p, and the relative importance of the factors is given by the weights, w_i.

They find the generalized mean an attractive alternative for two of its properties [40]

- min(a,b) ≤ mean(a,b) ≤ max(a,b)
- mean increases with an increase in p; thus, by varying the value of p from $-\infty$ and $+\infty$, one can obtain all values between min and max.

In their sensor fusion work, they suggest using the generalized mean aggregation operator and performing an adaptive, on-line fusion methodology. First, they begin with a large value of p and decrease the value of p as the number of sensors increase, thus, becoming more selective as more information is introduced into the system. As each additional sensor's information is considered, it is fused with the existing information, then a confidence factor is calculated. If the information from the current sensor does not increase the confidence factor, it is not included in the decision process. A final decision on fault identification is made by the Method of Heights Defuzzification Scheme.

The other class of applications of decision-making aggregation methods is illustrated by the work of Kaymak and van Nauta Lemke [41] to develop a method of preference ordering among alternatives. After evaluating the various t-norms (i.e., fuzzy conjunctive operators, Boolean logic AND (e.g., minimum)) and t-conorms (i.e., fuzzy disjunctive operators, Boolean logic OR (e.g., maximum)) available as aggregation operators, they determine that the compensatory operators, which are a combination of a t-norm and a t-conorm are the most appropriate for modeling the human information aggregation method used to preference order among given alternatives.

It is evident from these two brief descriptions that most of the research to date in this area has focused on determining an appropriate operator to combine the information at hand. Additionally, neither of these application areas completely address the OT&E aggregation task. Information gathered on certain aspects of the system performance cannot simply be disregarded, as was suggested in the sensor fusion work, for all of it is relevant to the decision being made. This work is not trying to rank order any alternatives associated with the system performance, so the information on preference ranking, although insightful, does not help with this task. Finally, in applying these aggregation schemes to the testbed case data, it became evident that the information content was watered down to the point where it would be difficult to make a decision. In viewing the results derived from these aggregation schemes on the testbed case data, Major General George Harrison's[8] reaction was "Basically, you are not telling me anything [42]." Figure 32 and Figure 33 are the results from performing the aggregation of the COMMFFYs

[8] Major General George B. Harrison is the current (1993 - current) commander of the Air Force Operational Test and Evaluation Center -- the Air Force's operational test and evaluation organization.

from the testbed case using the methods described above. The aggregation is carried out using the minimum, maximum, and generalized mean operators on the best-case and worst-case COMMFFYs separately. The figures illustrate the inability of these aggregation methods to distinguish between conflicting or confirming information -- they treat everything equally, resulting in information of dubious worth.

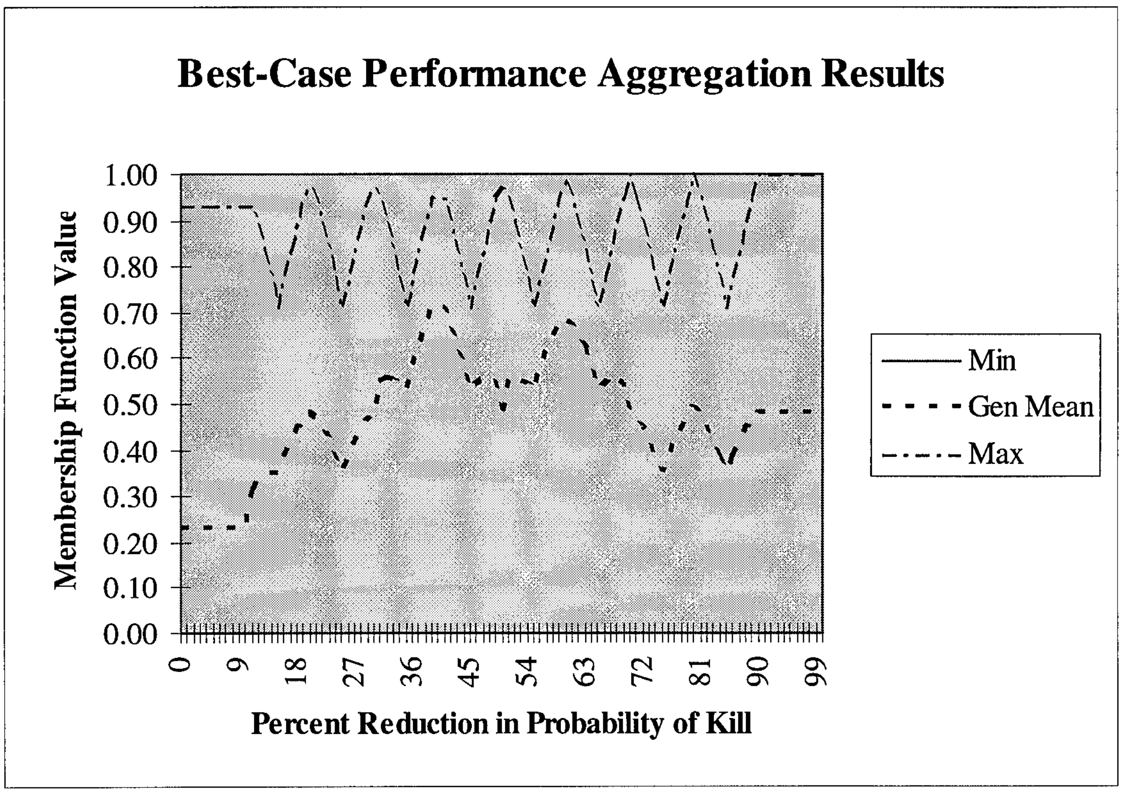

Figure 32 Best-Case Performance Aggregation Results

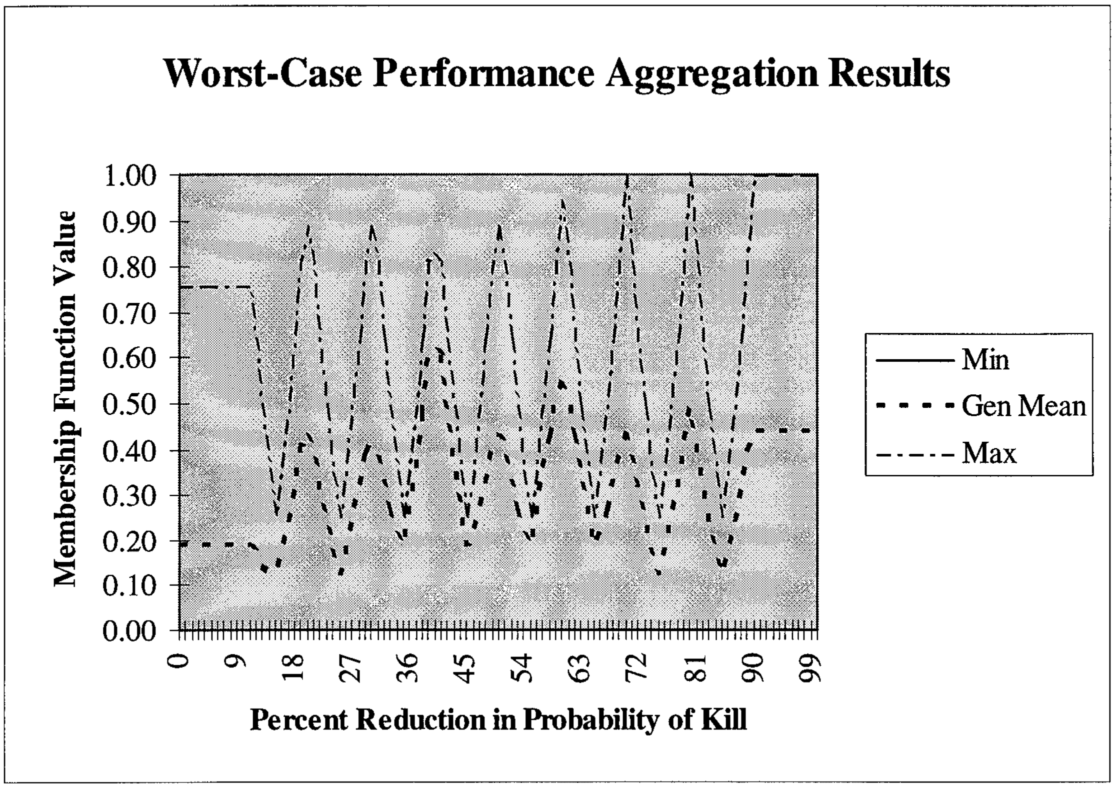

Figure 33 Worst-Case Performance Aggregation Results

Because the raw test data in the testbed case indicated such diverse performance against the individual threat systems, the aggregation methods shown here are very unsatisfying. The minimum operation yields a result of zero across the entire domain. Thus, even through the Jammer-X system performed quite well against one of the threats and acceptably well against the remainder, this aggregation scheme indicates the intersection of the performance across the threats has no merit. On the other hand, the maximum operation using the fuzzy union of the performance yielded an overall performance function that is overly optimistic. Finally, the generalized mean provides middle of the road membership function values, but again because of the diversity of the original data, ranges over the majority of the domain, resulting in very little useful information.

Are there other aggregation alternatives?

6.1.1 OTHER AGGREGATION OPTIONS

A *neural network* consisting of nodes used to form a weighted sum of the input values, which then passes through a nonlinearity, could be used to aggregate the individual performance COMMFFYs to an overall performance COMMFFY [43]. However, the use of a neural network for this application presents a number of problems. First, the neural network's "black-box" operation mode is not desirable for this application. One of the goals in the development of this analysis methodology is to develop a method where all the decisions can be justified, and the reasoning process can be understood intuitively. Using a neural network at this stage would destroy the intuitive appeal of the *Intelligent Hierarchical Decision Architecture* developed to this point. Second, and more importantly, the transformation from individual system performance to overall system performance is not well understood, and input/output data representative of it, which could be used as training data for the neural network, is not readily available. Without training data available to train the network for its desired performance, it would be impractical to pursue this option. Therefore, the neural network, although a tool that could be applied to accomplishing this portion of the task, will not be considered further.

A *Fuzzy Associative Memory* could be developed to relate various aspects of system performance to overall system performance for the final aggregation step, similar to the FAM used in the second stage of the hierarchy to aggregate MOFP-level performance to MOTA-level performance. The FAM structure provides an appealing method of tracking the decisions being made through the decision-making process. However, a problem similar to that highlighted in considering the neural network for this application, is the lack of information relating performance at the two levels in question. Where would the information used to develop the FAM come from? For the testbed case, mission-level models exist which allow the user to define lethality values for each threat, then fly a mission through the defined scenario. These models could be used to define the individual-performance to aggregated-performance relationships for an application like the testbed case. However, in general, these types of models do not exist and the type of information that would be needed to develop a FAM for the aggregation task is not available. Therefore, as with the neural network option, this option is abandoned due to a lack of information available to implement it.

Another approach for aggregating the information would be to defuzzify each individual MOTA-level COMMFFY, then perform a *weighted averaging of the crisp values* to come up with a final result. In using this approach, although the defuzzification scheme would consider the information contained within each individual COMMFFY, performing the defuzzification operation too early in the process diminishes the rich information content resident in the fuzzy distribution. Additionally, for the testbed case, previous attempts at developing a linear weighting scheme for Electronic Combat system performance have met with criticism from the Electronic Combat community [44].

Thus, it can be seen that neither neural networks, Fuzzy Associative Memories, nor a defuzzified value linear-weighting scheme are appropriate for this work. The work done in the sensor fusion and preference ordering areas, also do not quite fit the bill. The Dempster-Shafer (D-S) Theory of Evidential Reasoning, introduced by Dempster in the 1960's and extended by Shafer, provides a generalization to the Bayesian reasoning framework. The D-S method will prove to be an excellent method of aggregating the information across the logical divisions of the task level system performance and providing a probabilistic bound on the system performance for the decision-maker.

6.2 DEMPSTER-SHAFER THEORY

The Dempster-Shafer Theory is a generalization of the Bayesian probabilistic reasoning approach. It allows the hypotheses being considered to be subsets of an overall universe, in addition to the singleton hypotheses allowed by the Bayesian approach [45]. The D-S approach represents knowledge as a mapping of knowledge sources in the observation space, or bodies of evidence, to hypotheses in the conclusion space [46]. The D-S theory will serve as the basis for the *Intelligent Hierarchical Decision Architecture's Aggregation Methodology.* Appendix D contains an overview of the basic definitions associated with the Dempster-Shafer Theory. The Appendix also provides an example of Dempster's Rule of Combination. The discussion here now focuses on an illustration of how the Dempster-Shafer basic probability assignment can be derived from fuzzy sets, then on a discussion of the use of Dempster's Rule of Combination to perform the final aggregation of the *Intelligent Hierarchical Decision Architecture's* processing.

6.2.1 DERIVING BPA VALUES FROM FUZZY SETS

Several researchers have suggested a link between the uncertainty associated with fuzzy sets and the knowledge combination functions of the Dempster-Shafer theory [47, 48] in an effort to provide a seamless method of combining uncertain information. In those works, the consideration of the fuzzy set as a possibility distribution suggested by Zadeh in [49] is used to define the Dempster-Shafer basic probability assignment for the fuzzy set. The bpa is subsequently used to define the belief and plausibility functions, and as the basis for the combination in extensions of the Dempster's Rule of Combination.

If the elements within the fuzzy set are nested, that is, arranged such that they form subsets

$A_1 \supset A_2 \supset \ldots \supset A_n$, then the bpa is calculated from the possibility distribution as [47]

$$m(A_i) = \pi_i - \pi_{i-1} \tag{6-3}$$

where $\pi_i = \inf_{x \in A_i} poss(x)$. Using the α-cut level to decompose the fuzzy set, a natural nesting results and the π_i becomes the alpha cut level, therefore, the bpa is formed from

$$m(A_i) = \alpha_i - \alpha_{i-1} \tag{6-4}$$

Once the bpa is defined, the belief and plausibility functions are defined as [48]

$$Bel(B) = \sum_A m(A) \sum_{\alpha_i} [\alpha_i - \alpha_{i-1}] \times \inf_{x \in A_{\alpha_i}} \mu_B(x) \tag{6-5}$$

$$Pl(B) = \sum_A m(A) \sum_{\alpha_i} [\alpha_i - \alpha_{i-1}] \times \sup_{x \in A_{\alpha_i}} \mu_B(x) \tag{6-6}$$

The *Rule of Combination* is extended for use with fuzzy sets in a two step operation, consisting of combining the bpa's using a cross-product operation followed by a normalization step that brings the maximum value of the fuzzy set to unity. First the fuzzy bpa's are combined using the following [48]

$$m'_{12}(C) = m_1 \otimes m_2(C) = \sum_{A \cap B = C} m_1(A) m_2(B) \tag{6-7}$$

where C is an unnormalized intersection of the fuzzy sets A and B and m' is the unnormalized, combined bpa. Then the normalization process is applied as [48]

$$N[m'](D) = \frac{\sum_{\overline{C}=D} \max_{x_i} \mu_C(x_i) m'(C)}{1 - \sum_{C \subset T} (1 - \max_{x_i} \mu_C(x_i)) m'(C)} \tag{6-8}$$

where $\overline{C}$ is the normalized set associated with C that is characterized by the following membership function

$$\mu_{\overline{C}}(x) = \frac{\mu_C(x)}{\max_x \mu_C(x)} \tag{6-9}$$

and T is the hypothesis space, formed by a set of mutually exclusive and exhaustive hypotheses of the form IF X is s_i THEN Y is A_i where s_i is a piece of evidence and A_i is a fuzzy subset in the hypothesis space T.

6.3 THE IHDA AGGREGATION METHODOLOGY

The Dempster-Shafer Theory of Evidential Reasoning will serve as the basis for the final stage of the *Intelligent Hierarchical Decision Architecture* to aggregate the information across all the logical divisions of the system-under-test's performance. In the case of the testbed case, the aggregation is across all the threat systems against which the Jammer-X was tested. The final aggregated solution gives a system performance bound on the *Reduction in* P_k capability of the Jammer-X system for all the threat systems that the system is likely to encounter in its operational use.

The *Intelligent Hierarchical Decision Architecture's Aggregation Methodology* will use the work described above as a guide. The aggregation will be carried out in two parallel operations. All the best-case COMMFFYs resulting from the best-case FCM adjustment of the previous step will be aggregated together to form a best-case probabilistic bound on the system performance. Likewise, all the worst-case

COMMFFYs resulting the from the worst-case FCM adjustment will be aggregated together to form a worst-case probabilistic bound on the system performance. The *Intelligent Hierarchical Decision Architecture's Aggregation Methodology* entails the following steps.

Step 1. In order to make the processing in this stage manageable and the results intuitively meaningful, the aggregation is carried out at the *Basic Membership Function* level. Therefore, the first step entails condensing the information contained within the COMMFFYs generated from the first three stages of the *Intelligent Hierarchical Decision Architecture* back to the *Basic Membership Function* level. This is accomplished by looking at the universe of discourse associated with each BMF and selecting the maximum value within the region to represent the membership function value for that BMF.

Step 2. Once the BMF values for each logical division are defined, nested sets are formed by taking α-cuts at each distinct membership function value.

Step 3. With the α-cut level sets defined, the basic probability assignment associated with each α-cut level set is determined using (6-4).

Step 4. The combination and normalization are carried out with these basic probability assignment values using the intersection tableau method illustrated in Appendix D. The intersection tableau method incorporates the calculations given in (6-7) and (6-8) in a more illustrative manner, therefore, it has been chosen for use here.

Step 5. Finally, the belief and plausibility functions are calculated using the equations given in Appendix D. These function values are used to form the belief interval, the final information that is supplied to the decision-maker to aid in his decision-making process. The belief interval takes the form [101]

$$[Bel(A),\ Pl(A)] \qquad \textbf{(6-10)}$$

giving a lower and upper bound on the probability of the hypothesis. The Degree of Certainty is also calculated and provided to the decision-maker, as an indication of the believability associated with each choice.

6.4 TESTBED CASE RESULTS

The Aggregation Methodology described in Section 6.3 is now illustrated using the testbed case. Each step in the methodology is illustrated for a portion of the testbed case, with complete results shown in Appendix F.

Step 1. In this step the information contained in the adjusted, task-accomplishment level COMMFFY is transformed back to a representative value at the *Basic Membership Function* level. This is accomplished by taking the maximum value of the COMMFFY within the domain represented by each BMF. The results of this operation for the best-case adjustment are shown in Table 10.

Table 10 BMF Membership Function Values for Best-Case Adjustment COMMFFYs

BMF Tag	BMF #	Threat A	Threat B	Threat C	Threat D
LO	0	0.00	0.00	0.00	0.00
LOMEDLO	1	0.00	0.00	0.71	0.71
MEDLO	2	0.00	0.00	0.94	0.84
HIMEDLO	3	0.00	0.71	0.98	0.93
MED	4	0.00	0.93	0.71	0.93
LOMEDHI	5	0.00	0.93	0.00	0.77
MEDHI	6	0.71	0.71	0.00	0.75
HIMEDHI	7	0.96	0.00	0.00	0.82
HI	8	0.99	0.00	0.00	0.71

Step 2. The next step is to form α-cut level sets for each distinct membership function value. This step is illustrated using the BMF membership function values for Threat D shown in Table 10, the complete results are given in Appendix F.

$D_{0.71} = \{1,2,3,4,5,6,7,8\}$

$D_{0.75} = \{2,3,4,5,6,7\}$

$D_{0.77} = \{2,3,4,5,7\}$

$D_{0.82} = \{2,3,4,7\}$

$D_{0.84} = \{2,3,4\}$

$D_{0.93} = \{3,4\}$

Step 3. With the α-cut level sets defined, the basic probability assignment value associated with each α-cut level set is determined as shown below for the sets given in Step 2 above.

$m(D_{0.71}) = 0.71$

$m(\, D_{0.75}\,) = 0.04$

$m(\, D_{0.77}\,) = 0.02$

$m(\, D_{0.82}\,) = 0.05$

$m(\, D_{0.84}\,) = 0.02$

$m(\, D_{0.93}\,) = 0.09$

$m(\ \Theta\ ) = 0.07$

Note that the remainder of the evidence not committed to any of the α-cut level sets is assigned to the universe of discourse.

Step 4. The combination and normalization are carried out with these basic probability assignment values using the intersection tableau method illustrated in Appendix D. The method is illustrated below in combining the evidence associated with Threat A and Threat B, with complete results given in Appendix F.

Table 11 Intersection Tableau for Testbed Case Threat A and Threat B

	$m(A_{0.71}) = 0.71$ $A_{0.71} = \{6,7,8\}$	$m(A_{0.96}) = 0.25$ $A_{0.96} = \{7,8\}$	$m(A_{0.99}) = 0.03$ $A_{0.99} = \{7,8\}$	$m(\Theta) = 0.01$
$m(B_{0.71}) = 0.71$ $B_{0.71} = \{3,4,5,6\}$	$m\{6\} = 0.5041$	$m\{\varnothing\} = 0.1775$	$m\{\varnothing\} = 0.0213$	$m\{3,4,5,6\} = 0.0071$
$m(B_{0.93}) = 0.22$ $B_{0.93} = \{4,5\}$	$m\{\varnothing\} = 0.1562$	$m\{\varnothing\} = 0.0550$	$m\{\varnothing\} = 0.0066$	$m\{4,5\} = 0.0022$
$m(\Theta) = 0.07$	$m\{6,7,8\} = 0.0497$	$m\{7,8\} = 0.0175$	$m\{8\} = 0.0021$	$m(\Theta) = 0.0007$

From this tableau, the values of the bpa for all the subsets shown here are calculated and adjusted by the value of the bpa assigned to the empty set, by normalizing by 1-K, where K is the sum of the bpa's assigned to the empty set.

$K = 0.1775 + 0.0213 + 0.1562 + 0.0550 + 0.0066 = 0.4166;\ 1\text{-}K = 0.5834$

$m\{6\} = 0.5041 / 0.5834 = 0.8641$

$m\{8\} = 0.0021 / 0.5834 = 0.0036$

$m\{4,5\} = 0.0038$

$m\{7,8\} = 0.0300$

$m\{6,7,8\} = 0.0852$

$m\{3,4,5,6\} = 0.0122$

$m\{\Theta\} = 0.0012$

Then these values are carried forward to the next tableau, which combines them with the evidence from Threat C. Then the combined A,B, and C values are combined with the evidence from Threat D. The complete results are shown in Appendix F.

Step 5. Finally, the belief and plausibility functions and the Degree of Certainty are calculated. The function values are used to form the belief interval, the final information that is supplied to the decision-maker to aid in his decision-making process. The results for the best-case bpa, belief function, plausibility function, and the Degree of Certainty are given in Table 12. The bpa values came from the final tableau calculations where the combined Threats A, B, and C information was combined with Threat D information (see Appendix F). The belief and plausibility function values are calculated, for example for Bel{2,3} and Pl{2,3} as

$$\text{Bel}\{2,3\} = m\{2\} + m\{3\} + m\{2,3\} = 0 + 0.1037 + 0.0078 = 0.1114$$

$$\begin{aligned}\text{Pl}\{2,3\} &= m\{3\} + m\{2,3\} + m\{3,4\} + m\{2,3,4\} + m\{3,4,5\} + m\{1,2,3,4\} + m\{3,4,5,6\} \\ &\quad + m\{2,3,4,7\} + m\{2,3,4,5,7\} + m\{2,3,4,5,6,7\} + m\{1,2,3,4,5,6,7,8\} + m\{\Theta\} \\ &= 0.4127\end{aligned}$$

Table 12 Basic Probability Assignment, Belief and Plausibility Function, and Degree of Certainty Values for Best-Case Aggregated System Performance

Hypothesis	Basic Probability Assignment	Belief	Plausibility	Degree of Certainty
{1}	0.0000	0.0000	0.0000	-1.0000
{3}	0.0000	0.0000	0.0000	-1.0000
{4}	0.0000	0.0000	0.0000	-1.0000
{5}	0.9621	0.9621	0.9925	0.9242
{7}	0.0007	0.0007	0.0379	-0.9985
{1,2}	0.0000	0.0000	0.0000	-1.0000
{1,3}	0.0000	0.0000	0.0000	-1.0000
{4,5}	0.0000	0.9621	0.9925	-0.0379
{5,7}	0.0015	0.9644	1.0000	-0.0341
{1,2,3}	0.0000	0.0000	0.0000	-1.0000
{3,4,5}	0.0000	0.9621	0.9925	-0.0379
{6,7,8}	0.0068	0.0075	0.0379	-0.9857
{1,4,5,7}	0.0000	0.9644	1.0000	-0.0356
{5,6,7,8}	0.0288	0.9712	1.0000	0.0000
{1,3,4,5,7}	0.0000	0.9644	1.0000	-0.0356
{0,1,2,3,4,5,6,7,8}	0.0000	1.0000	1.0000	0.0000
{Θ}	0.0000	1.0000	1.0000	0.0000

These probability bounds would be reported to the decision-maker as the final outcome of the OT&E, adjusted for factors that could not be tested or controlled during the testing. The table should be interpreted as follows. The hypotheses listed in the left column are the possible *Basic Membership Functions* that could answer the question "Which Basic Membership Function most likely characterizes the aggregated and adjusted best-case system performance of Jammer-X?". The Degree of Certainty value points to the hypothesis that has the most confirming evidence and the least non-confirming evidence. As mentioned in the definition of this measure, it ranges [-1, +1] with -1 indicating total disbelief in the hypothesis and +1 indicating total belief in the hypothesis. From the values listed in Table 12, it can be seen that most of the evidence is pointing to a conclusion of BMF #5, the triangular shaped Basic Membership Function centered at 60% Reduction in Probability of Kill. The belief interval associated with this hypothesis is [0.9621, 0.9925], indicating a very strong conclusion.

6.5 CONTRIBUTION

The *Aggregation Methodology* described in this chapter serves as the final stage in the *Intelligent Hierarchical Decision Architecture*, which in its entirety provides advances for both the Systems Analysis and Test and Evaluation arenas. The *Intelligent Hierarchical Decision Architecture* provides a means to take low-level test data and aggregate and synthesize it into information that is truly meaningful to the decision-maker. The Aggregation Methodology, separate from the entire *Intelligent Hierarchical Decision Architecture*, provides new methodologies for both the Systems Analysis and T&E worlds.

In Systems Analysis, previous methods used to give a system performance bound have been limited to statistical methods which typically depend more on sample size than on information content for the size of the bound. Using the Dempster-Shafer Theory, the need to use statistical measures to establish the system performance bound is eliminated, building a bound whose size is based solely on the gathered information. Additionally, the D-S theory allows the transformation of the information from the possibilistic realm to the probabilistic, where decision-makers are more comfortable. This transformation is made in the definition of the basic probability assignments using α-cuts of the fuzzy set information, and the subsequent manipulation of the bpa's to form the final probabilistic bounds.

In the area of Test and Evaluation, only ad-hoc methods have been developed to date to allow aggregation of information to provide overall system performance evaluations. For example, in the F-15 TEWS Operational Assessment, a color scheme was used through which system performance against each threat was given a green, yellow, or red rating. Then, the overall system performance was determined by the test team by looking at the spectrum of colors and making a judgment on overall system performance [73]. In the OT&E of the Milstar Satellite Communication system, an aggregation methodology was developed through which any failure of what was termed a "critical" measure would cause the failure of the entire system. If other than critical measures failed, a group of test team members got together to decide the relevance to system performance [72]. In other OT&E programs, no attempt has been made to aggregate system performance past the individual measured performance level -- the individual results are reported and the aggregation method is left up to the decision-maker. Clearly, a more systematic method, as suggested here, is needed to standardize the analysis of OT&E data.

CHAPTER SEVEN

CONCLUSION

---

7. CONCLUSION

In this dissertation, the *Intelligent Hierarchical Decision Architecture* has been developed. It provides an intelligent method of analyzing Operational Test and Evaluation data with the goal of providing meaningful information to decision-makers based upon the low-level data gathered during the testing effort. This final chapter concludes the discussion by comparing the methodology with currently available statistical/probabilistic and M&S methods and providing an *information-content measure*, based upon fuzzy entropy concepts that will be used to show that with the progression through the processing of the *Intelligent Hierarchical Decision Architecture* the amount of entropy (a measure of the inability to make a decision) in the system decreases. Finally, this chapter closes with a review of the total work and a discussion of the overall contributions that this work has made to the fuzzy logic theory, system analysis, and Operational Test and Evaluation arenas. Also included is a discussion of future research that should be conducted in order to further the work accomplished here.

7.1 COMPARISON WITH CURRENT METHODS

7.1.1 FUNCTIONAL PERFORMANCE LEVEL DATA MANIPULATIONS

The analysis method proposed in this work provides a means of drawing conclusions at the operational task level based upon system performance characteristics demonstrated at the functional performance level. How does this method compare to those currently being used, or those that could be proposed based completely upon statistical or probabilistic techniques?

The current analysis methods of the operational testing community, as discussed in Chapter One, are limited to standard statistical methods. Using these methods, the data gathered at the functional performance level can be analyzed, and conclusions drawn based upon the pass/fail criteria at that information level. The most appropriate statistical tool for the analysis of the data provided in the testbed case would be Statistical Hypothesis Testing, yielding a statement on pass/fail against the requirement and a measure of the degree of plausibility associated with that statement, in the form of a p-value. The p-value gives a probability associated with the extremity of the current observation. That is, if the experiment were repeated 100 times, only *p* of the trials would result in a value as extreme as the current result. Therefore, a small value of *p* points to the implausibility of the hypothesis. A rough guide to interpreting the p-value is given as [58]:

- not significant if $p > 0.10$
- mildly significant if $p \leq 0.10$ but $p > 0.05$
- significant if $p \leq 0.05$ but $p > 0.01$
- highly significant if $p \leq 0.01$

where significance means that there is evidence indicating a rejection of the null hypothesis.

The calculations associated with the analysis of the testbed case data are illustrated below for two of the Measures of Functional Performance, one measure where the requirement is given in "*greater than or equal to*" form and the other where the requirement is given in "*less than or equal to*" form. The final result of the analysis that would be presented to the decision-maker, would be a collection of this information, usually in the form shown in Table 13.

Illustrating the calculations associated with MOFP #1, Threat D, (see Appendix F for a listing of the raw data), the hypothesis being tested is:

H_0: Reduction in Hits (RIH) $\geq$ 50%

H_1: Reduction in Hits (RIH) < 50%

Using a Student-t distribution, due to the small sample size, the statistical test would reject H_0 if $t \leq -c$ where $t = \dfrac{\overline{X} - \mu}{s/\sqrt{n}}$ and c, the critical value, is calculated as $\Pr(t \geq c) = \alpha$, where alpha is the desired risk of making a Type I error.

Using $\alpha = 0.10$ and the t-distribution, then $c = 1.383$. With $n = 10$, $\overline{X} = 55.65$, $s = 15.98$, the result is $t = 1.1181$. Thus, the outcome is to fail to reject H_0 (i.e., accept H_0) because t > -1.383. The associated p-value, the quantitative measure of the plausibility of the hypothesis, is $p = 0.87$, a very strong indication that the null hypothesis should be accepted.

To illustrate the calculations associated with the one-sided hypothesis when the requirement is such that the value should fall below it, examine the analysis of MOFP #6, Threat D, where

H_0: Response Time $\leq$ 10 seconds

H_1: Response Time > 10 seconds

The only change is the statistical test, where now the test rejects H_0 if $t \geq c$.

Using a one-sided test with $\alpha = 0.10$, again, $c = 1.383$. With $n = 10$, $\overline{X} = 8.38$, $s = 3.65$, the result is $t = -1.40$. Thus, this result causes a failure to reject H_0 (i.e., accept H_0) because $t < 1.383$. The associated p-value, is $p = 0.42$, again, indicating a very strong conclusion to accept the null hypothesis.

The information provided to the OT&E decision-maker, has typically taken the form of a summary chart illustrating the pass/fail determinations at the functional performance level determined from the hypothesis testing. The information provided in the chart typically is given in the form of red, yellow, or green ratings -- indicating a relative degree of acceptance of the hypothesis in question. Various criteria have been established for the color-coding of the information provided within the chart (i.e., what constitutes a red, yellow, or green rating) but consistently, the information provided to the decision-maker stops at the level of providing pass/fail determinations against each *Measure of Functional Performance* and logical system division that was examined during the testing effort. A sample chart, giving pass/fail determinations based solely upon the results of the hypothesis testing is illustrated in Table 13. This table also illustrates the associated p-values so the decision-maker can determine the confidence associated with the decision-making.

Table 13 MOFP/Threat Pass/Fail Determinations with p-value

	MOFP #1	MOFP #2	MOFP #3	MOFP #4	MOFP #5	MOFP #6
Threat A	Pass (p=1)	Pass (p=1)	Pass (p=1)	Pass (p=1)	Pass (p=1)	Pass (p=1)
Threat B	Pass (p=1)	Pass (p=1)	Pass (p=1)	Pass (p=1)	Pass (p=1)	Pass (p=1)
Threat C	Fail (p=0)	Fail (p=0)	Fail (p=0)	Fail (p=0)	Fail (p=0)	Fail (p=0)
Threat D	Pass (p=0.87)	Pass (p=0.68)	Pass (p=0.53)	Pass (p=0.50)	Pass (p=0.46)	Pass (p=0.42)

The other method, frequently used within the OT&E community to present the functional performance level summary information to the decision-maker is to construct a color-coded decision chart that examines the test data's mean surrounded by an associated confidence interval. If the requirement falls completely to the "correct" side of the confidence interval the system is given a green rating, if the requirement falls completely to the "wrong" side of the confidence interval the system is given a red rating, and if the requirement is contained within the confidence interval a yellow rating is used. Using this method with the testbed case, the information given to the decision-maker as a result of the testing effort would be as illustrated in Table 14. The decision-maker would then make his decision based upon the predominant color in the chart.

Table 14 Color-Coded Decision Chart For Testbed Case Results

	MOFP #1	MOFP #2	MOFP #3	MOFP #4	MOFP #5	MOFP #6
Threat A	Green	Green	Green	Green	Green	Green
Threat B	Green	Green	Green	Green	Green	Green
Threat C	Red	Red	Red	Red	Red	Red
Threat D	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow

Other statistical methods could have been applied to the data at this point in the analysis, providing alternate means of drawing conclusions as to the significance of the testing outcomes. These other methods might include such approaches as nonparametric statistics, where conclusions can be drawn without making assumptions on the underlying statistical distribution of the data [50]; sequential methods where the statistical hypotheses are examined after each observation, or group of observations, and a determination is made to accept or reject the hypotheses or reserve judgment until more data are collected [5]; or analysis of variance techniques where conclusions are drawn as to the significance of the outcome based upon a comparison to the variance in the process.[9] However, these methods are limited to drawing conclusions at the level where data are available, they do not provide a mechanism for extrapolating to higher information levels where data cannot be gathered.

7.1.2 Task Accomplishment Level Information Generation

The current analysis methods used within the operational testing community end with the statistical inference methods illustrated in Section 7.1.1, requiring the decision-maker to draw conclusions from the pass/fail information provided. Are there other approaches that could be applied to allow conclusions to be drawn at the higher information content levels?

Probabilistic-based methods, such as Bayesian Inference or Dempster-Shafer Theory might offer an analysis mechanism. Using the Bayesian approach, the conditional probabilities that relate each of the functional performance measures to the task accomplishment measure could be defined, then the methods of combining the evidence suggested by the Bayesian framework could be used to draw conclusions.

[9] The data collected from the testbed case does not lend itself to an analysis of variance approach, where comparisons of various alternatives yield statements on the significance of one alternative over the other. The original test design for the Jammer-X system would have to have considered the ANOVA techniques and gathered different data to facilitate that type of analysis.

To illustrate this methodology on the testbed case, first, assume independence among the functional performance measures, then, define hypotheses and divisions of the evidence space. Then a matrix is derived that relates each level of the measures of functional performance with each hypothesis. An excerpt of the matrix, using the independence assumption, for the testbed case might look like [51]:

H_1: P_k performance is well above the requirement

H_2: P_k performance is slightly above the requirement

H_3: P_k performance is slightly below the requirement

H_4: P_k performance is well below the requirement

Each piece of evidence (measure of functional performance) would also be divided into these four hypothetical cases, such that:

e_1^1: MOFP #1 performance is well above the requirement

e_1^2: MOFP #1 performance is slightly above the requirement

:

:

e_4^6: MOFP #6 performance is well below the requirement

Using this notation, the matrix of conditional probabilities is formed as:

	e_1^1	e_1^2	...	e_4^6
H_1	$P(e_1^1 \mid H_1)$	$P(e_1^2 \mid H_1)$	...	$P(e_4^6 \mid H_1)$
H_2	$P(e_1^1 \mid H_2)$	$P(e_1^2 \mid H_2)$	...	$P(e_4^6 \mid H_2)$
H_3	$P(e_1^1 \mid H_3)$	$P(e_1^2 \mid H_3)$	...	$P(e_4^6 \mid H_3)$
H_4	$P(e_1^1 \mid H_4)$	$P(e_1^2 \mid H_4)$	...	$P(e_4^6 \mid H_4)$

Then, assuming conditional independence with respect to each hypothesis, the overall belief in the i-th hypothesis is calculated from

$$P(H_i|e^1,\dots e^N)=\frac{P(H_i)}{[P(e^1,\dots e^N)]}\prod_{k=1}^{N}P(e^k|H_i) \tag{7-1}$$

The issues with adopting this approach are listed below.

- First, although a method of defining the transformational relationships between the functional performance level and the task accomplishment level used in the Fuzzy Associative Memory was determined, the definition of these relationships within the Bayesian framework requires a far more rigorous approach -- one for which the amount of information available within the OT&E context will seldom yield.

- Second, the combination of information within the Bayesian framework assumes independence between the factors -- an assumption that is severely violated for the case of the OT&E data where a series of interrelated measurements on a single system are taken.

- Third, if the multi-valued approach to knowledge combination is adopted, a matrix consisting of all the combinations of possible outcomes and possible test conditions would have to be developed. Just considering four values for each functional performance measure, and four outcomes for the task accomplishment level (i.e., a much courser division of the hypothesis space than was incorporated in the fuzzy-based approach), the matrix of probabilities would contain $4^6 = 4096$ elements. This combinatorial explosion problem within the fuzzy framework was dealt with using the reduction theorem, however, in the probabilistic case, unless the (unrealistic) independence assumption is made, no such shortcut exists.

- Fourth, most of the information necessary to carry out this calculation cannot be derived from the information gathered during an OT&E. For example, the $P(e_j^k | H_i)$ contained within the matrix, would have to be defined using some sort of a heuristic approach since data cannot be gathered to provide these probabilistic assessments; the $P(H_i)$ value would have to be arrived at heuristically, or an assumption made that all are equally likely, since testing cannot be carried out at that information level, and the $P(e^1,\ldots e^6)$ could only be given a rough value based upon the limited sample gathered the OT&E, with no method to determine if that sample is an adequate representation of the population.

So, even though a probabilistic method is available to provide information at the operational task level, the information needed to implement this method is seldom available in the OT&E context. Therefore, abandoning the application of the probabilistic-based methods, statistical model building techniques, such as regression analysis might be suggested as an alternative to provide information at the operational task level. Regression analysis provides a mechanism for building a model, which is subsequently used to predict the value of a variable based upon the values of the predictor variables contained in the model. This seems to offer the most promise as a means of predicting future performance from the information gathered during the testing. The problem with this approach is that an assumption has been made that measurements of the system's performance at the task accomplishment level cannot be made, thus, data necessary to build the model is not available

Another technique that could be suggested to perform the transformation from the functional performance level to the task accomplishment level, might be the use of Monte-Carlo simulation. If the assumption is made that the distribution of the functional performance levels can be characterized by the test data, and further, that a functional transform from the functional performance level to the task accomplishment level can be defined; then a Monte-Carlo simulation technique to draw inferences on the task accomplishment level distribution could be used. This is done by repeatedly sampling from the functional performance level distribution and determining the outcome due to that measurement at the task accomplishment level. If this sampling is accomplished repeatedly, a distribution of the measure at the task accomplishment level results. This simulation could be accomplished with the information that could be derived from the OT&E effort if the information to determine the functional transform from the functional performance level to the task accomplishment level can be quantified and if the data gathered during the OT&E are enough to characterize the underlying statistical distributions.

For the testbed case, an assumption is made that the data gathered during the testing are enough to characterize a normal distribution. A random sample is drawn from those distributions and the transformations used to derive the Fuzzy Associative Map rules (i.e., the curves derived from the ESAMS sensitivity runs) are used to transform the drawn sample at the functional performance level to a representative sample at the operational task level. This sampling and transformation process is repeated 1000 times for each functional performance measure and the results are binned and normalized to form a probabilistic distribution for each threat system. The results are shown in Figure 34 through Figure 37 below.

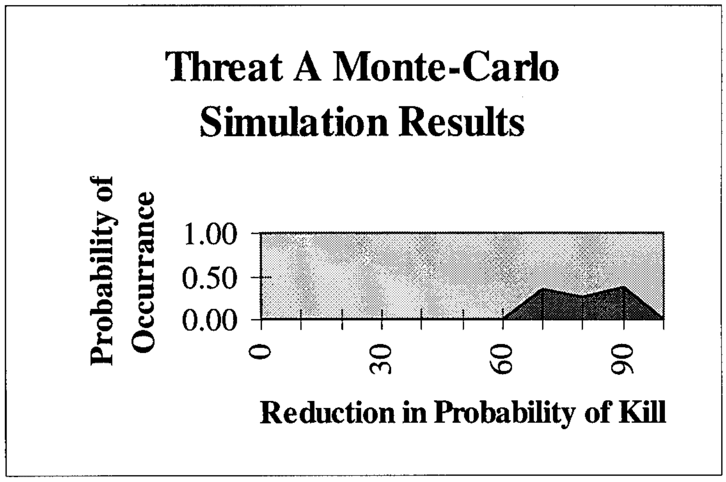

Figure 34 Percent Reduction in Probability of Kill Results for Threat A Derived From Monte-Carlo Simulation

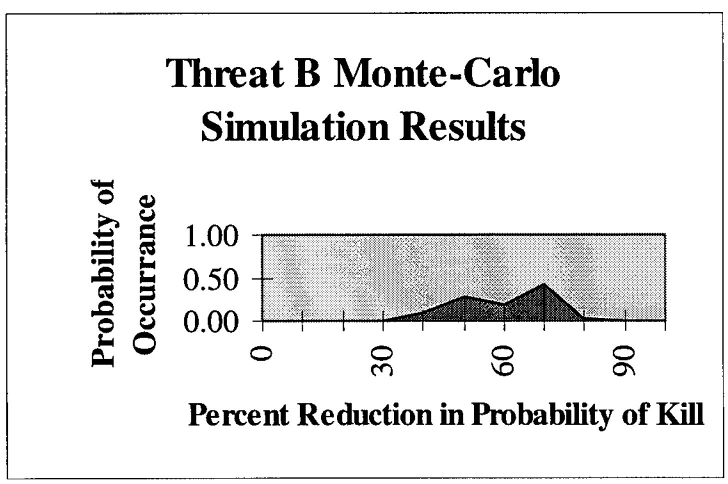

Figure 35 Percent Reduction in Probability of Kill Results for Threat B Derived From Monte-Carlo Simulation

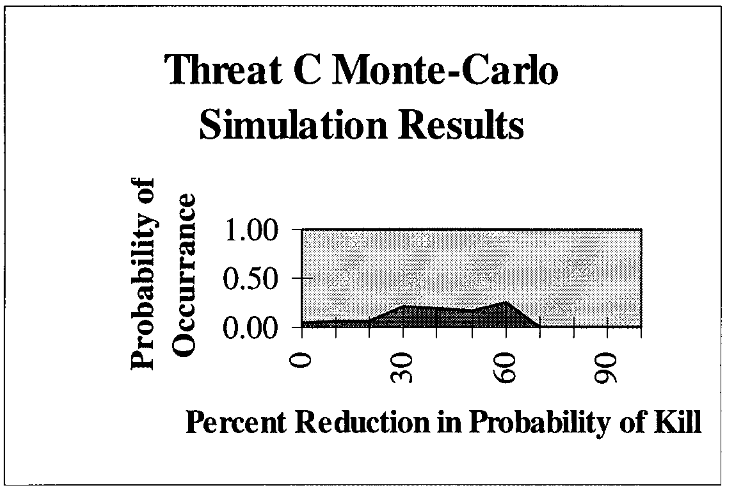

Figure 36 Percent Reduction in Probability of Kill Results for Threat C Derived From Monte-Carlo Simulation

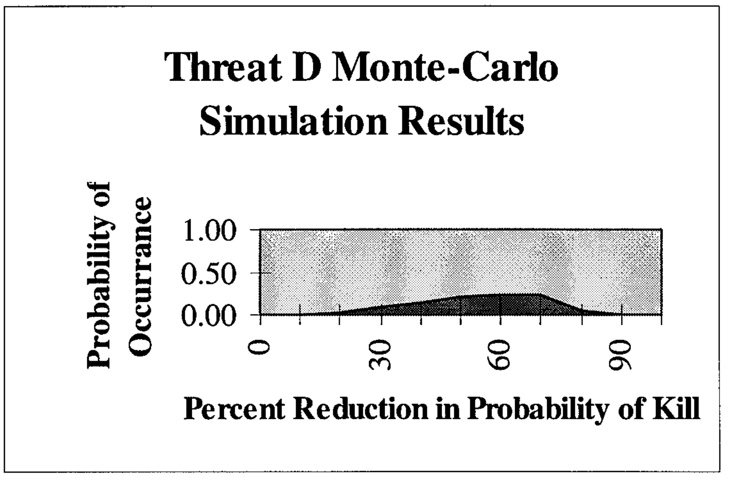

Figure 37 Percent Reduction in Probability of Kill Results for Threat D Derived From Monte-Carlo Simulation

Figure 34 through Figure 37 illustrate that the trends in the original data, that were also evident in the task-accomplishment level COMMFFYs, are also noticeable in these probabilistic distributions. Therefore, this simulation method could be used as a substitute for the methods proposed in the *Intelligent Hierarchical Decision Architecture's* first two stages if the test data are such that statistical distributions can be characterized from them, and if transformations from the functional performance level to the task accomplishment level can be defined in a functional form.

7.1.3 Final Information Aggregation

Once the probabilistic distributions generated through the Monte-Carlo simulation have been developed, they can be combined using the Dempster-Shafer Theory, forming a single bound on the system performance for the decision-maker, similar to the aggregation scheme used within the *Intelligent Hierarchical Decision Architecture.*

The issue encountered with this approach, especially for a situation such as the testbed case where the information being combined is very diverse, is the possibility of no overlapping regions in the hypothesis space. When there is no overlap the Dempster's Rule combination techniques will not yield a solution. This can be illustrated with the testbed case if the hypotheses are chosen as 10% intervals of the hypothesis space. The result from the combination of this information is given in Table 15 through Table 17.

Table 15 Dempster's Rule Combination of Threat A and Threat B (10% Hypotheses)

	$m\{70\text{-}79\}=$ 0.3500	$m\{80\text{-}89\}=$ 0.2600	$m\{90\text{-}99\}=$ 0.3800	$m\{100\}=$ 0.01000	$m\{\Theta\}=$ 0.0000
$m\{40\text{-}49\} = 0.0900$	0.0315	0.0234	0.0342	0.0009	0.0000
$m\{50\text{-}59\} = 0.2700$	0.0945	0.0702	0.1026	0.0027	0.0000
$m\{60\text{-}69\} = 0.1900$	0.0665	0.0494	0.0722	0.0019	0.0000
$m\{70\text{-}79\} = 0.4200$	0.1470	0.1092	0.1596	0.0042	0.0000
$m\{80\text{-}89\} = 0.0300$	0.0105	0.0078	0.0114	0.0003	0.0000
$m\{\Theta\} = 0.0000$	0.0000	0.0000	0.0000	0.0000	0.0000

With this combination, K= 0.8452 and 1-K= 0.1548, and the resulting combined basic probability assignments are $m_{AB}\{70\text{-}79\} = 0.9496$ and $m_{AB}\{80\text{-}89\} = 0.0504$. Then combining this information with Threat D, the result is shown in Table 16.

Table 16 Dempster's Rule Combination of Threat A/B with Threat D (10% Hypotheses)

	$m\{70\text{-}79\}=$ 0.9496	$m\{80\text{-}89\}=$ 0.0504	$m\{\Theta\}=$ 0.0000
$m\{20\text{-}29\}=0.0300$	0.0285	0.0015	0.0000
$m\{30\text{-}39\} = 0.0800$	0.0760	0.0040	0.0000
$m\{40\text{-}49\} = 0.1400$	0.1329	0.0071	0.0000
$m\{50\text{-}59\} = 0.220$	0.2089	0.0111	0.0000
$m\{60\text{-}69\} = 0.2400$	0.2279	0.0121	0.0000
$m\{70\text{-}79\} = 0.2400$	0.2279	0.0121	0.0000
$m\{80\text{-}89\} = 0.0500$	0.0475	0.0025	0.0000
$m\{\Theta\} = 0.0000$	0.0000	0.0000	0.0000

With the combination of Threat A/B with Threat D, K= 0.7696 and 1-K= 0.2304, and the resulting combined basic probability assignments are $m_{ABD}\{70\text{-}79\} = 0.9891$ and $m_{ABD}\{80\text{-}89\} = 0.0109$. Then, finally, combining this information with Threat C, the result is shown in Table 29. In that table it can be seen that there is no overlap in the hypotheses, therefore, no conclusions can be drawn on the overall system's performance.

Table 17 Dempster's Rule Combination of Threat A/B/D with Threat C (10% Hypotheses)

	$m\{70\text{-}79\}=$ 0.9496	$m\{80\text{-}89\}=$ 0.0504	$m\{\Theta\}=$ 0.0000
$m\{0\text{-}9\} = 0.0500$	0.0475	0.0025	0.0000
$m\{10\text{-}19\} = 0.0600$	0.0570	0.0030	0.0000
$m\{20\text{-}29\} = 0.0600$	0.0570	0.0030	0.0000
$m\{30\text{-}39\} = 0.220$	0.2089	0.0111	0.0000
$m\{40\text{-}49\} = 0.2000$	0.1899	0.0101	0.0000
$m\{50\text{-}59\} = 0.1700$	0.1614	0.0086	0.0000
$m\{60\text{-}69\} = 0.2500$	0.2374	0.0126	0.0000
$m\{\Theta\} = 0.0000$	0.0000	0.0000	0.0000

Rather than just abandoning the approach, by expanding the size of the hypothesis region until there is overlap, the method will allow conclusions to be drawn. In the testbed case, if the possible solution set is broken into 15% intervals, the combination across the threats with yield a solution, as illustrated in Table 18 through Table 20.

Table 18 Dempster's Rule Combination of Threat A and Threat B (15% Hypotheses)

	$m\{30\text{-}45\}=$ 0.0200	$m\{45\text{-}60\}=$ 0.3500	$m\{60\text{-}75\}=$ 0.5000	$m\{75\text{-}90\}=$ 0.1300	$m\{\Theta\}=$ 0.0000
$m\{60\text{-}75\} =$ 0.3300	0.0066	0.1155	0.1650	0.0429	0.0000
$m\{75\text{-}90\} =$ 0.2900	0.0058	0.1015	0.1450	0.0377	0.0000
$m\{90\text{-}100\} =$ 0.3800	0.0076	0.1330	0.1900	0.0494	0.0000
$m\{\Theta\}=$ 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

The result of this first combination is K= 0.7973, 1-K=0.2027, giving combined basic probability assignments of $m_{AB}\{60\text{-}75\} = 0.8140$ and $m_{AB}\{75\text{-}90\} = 0.1860$.

Table 19 Dempster's Rule Combination of Threat A/B with Threat C (15% Hypothesis)

	$m\{60\text{-}75\}=$ 0.8140	$m\{75\text{-}90\}=$ 0.1860	$m\{\Theta\}=$ 0.0000
$m\{0\text{-}15\} = 0.0800$	0.0651	0.0149	0.0000
$m\{15\text{-}30\} = 0.0800$	0.0651	0.0149	0.0000
$m\{30\text{-}45\} = 0.3400$	0.2768	0.0632	0.0000
$m\{45\text{-}60\} = 0.2500$	0.2035	0.0465	0.0000
$m\{60\text{-}75\} = 0.2500$	0.2035	0.0465	0.0000
$m\{\Theta\} = 0.0000$	0.0000	0.0000	0.0000

The resulting combinations from Table 19 yields $m_{ABC}\{60\text{-}75\} = 1.00$. Finally combining this result with Threat D, as in Table 20, the result is, again, $m_{ABCD}\{60\text{-}75\} = 1.00$. The final aggregated result drawn from the *Intelligent Hierarchical Decision Architecture*, indicated that the overall system performance at the operational task level (percent reduction in probability of kill) was the Basic Membership Function centered at 60%, while the conclusion drawn here is that the final aggregated performance is somewhere in the range between 60%-75%.

Table 20 Dempster's Rule Combination of Threat A/B/C with Threat D (15% Hypotheses)

	$m\{60\text{-}75\}=$ 1.0000	$m\{\Theta\}=$ 0.0000
$m\{15\text{-}30\} = 0.0300$	0.0300	0.0000
$m\{30\text{-}45\} = 0.1500$	0.1500	0.0000
$m\{45\text{-}60\} = 0.2900$	0.2900	0.0000
$m\{60\text{-}75\} = 0.4100$	0.4100	0.0000
$m\{75\text{-}90\} = 0.1200$	0.1200	0.0000
$m\{\Theta\} = 0.0000$	0.0000	0.0000

A side-by-side comparison of the final results from the statistical/probabilistic method and the *Intelligent Hierarchical Decision Architecture* method is shown in Table 21.

Table 21 Statistical Methods Final Result Comparison

Method	**Standard Method (10% Hypotheses)**	**Standard Method (15% Hypotheses)**	**Intelligent Hierarchical Decision Architecture**
Final Conclusion	No solution	60-75% {Meets Reqmt.}	BMF Centered at 60% {Meets Reqmt.}

Table 21 illustrates that when the Dempster-Shafer method is used with 10% hypotheses, no solution results. When the Dempster-Shafer method is used with the 15% hypotheses the same acquisition decision as derived from the *Intelligent Hierarchical Decision Architecture* results. However, the solution region is much broader than that derived from the IHDA method.

The difference in the results from the two methods can be attributed to two factors. (1) The *Intelligent Hierarchical Decision Architecture* considered the effect of other than the tested factors, thus providing the lower, and likely more realistic, estimate of the system performance. (2) The *Intelligent Hierarchical Decision Architecture's* conclusion is a more precise value (i.e., most likely 60%, then falling off in both directions in accordance with the shape of the *Basic Membership Function* vs. somewhere in the equally likely range of 60%-75%) because the gradual transitions considered through the use of fuzzy set theory allowed more overlapping information to be considered in the final aggregation step.

7.1.4 Statistical Methods Comparison Summary

The statistical/probabilistic methods described in Sections 7.1.1 and 7.1.2 can be used to achieve most of the tasks accomplished by the *Intelligent Hierarchical Decision Architecture*. However, two factors that were incorporated in the *Intelligent Hierarchical Decision Architecture* cannot be handled with a purely statistical/probabilistic approach. These two factors are: the incorporation of qualitative information at the functional performance level and the consideration of the outcome due to factors that could not be controlled or included in the testing effort. For example, in some cases, the information gathered during an OT&E could consist partially, or wholly, of subjective information gathered from surveys of the operational user. The development of a *Composite Fuzzy Membership Function* can be easily accomplished using that type of information, while statistical analysis of these data would be difficult, if not impossible. Second, the incorporation of the untestable/uncontrollable factors that were brought into the analysis through the use of the Fuzzy Cognitive Map, cannot be considered using a purely statistical/probabilistic approach. The realism brought into the analysis process due to the inclusion of these factors is a valuable means of helping the decision-maker make more informed decisions. A side-by-side comparison of the tasks accomplished by the *Intelligent Hierarchical Decision Architecture*, current OT&E practices, and other available statistical/probabilistic techniques is given in Table 22.

Table 22 Intelligent Hierarchical Decision Architecture / Statistical Methods Comparison

Function	Intelligent Hierarchical Decision Architecture Method	Current OT&E Analysis Methods	Statistical / Probabilistic Method
Manipulate raw data	Clustering Method	Hypothesis Testing, Analysis of Variance, Nonparametric Statistical Tests, etc.	*Hypothesis Testing*[10], Analysis of Variance, Nonparametric Statistical Tests, etc.
Transform from functional performance info level to task accomplishment info level	Fuzzy Associative Memory	N/A	Bayesian Inference, Dempster-Shafer Theory, *Monte-Carlo Simulation*
Consider factors not tested/controlled	Fuzzy Cognitive Map	N/A	N/A
Aggregate across logical divisions of system performance	Aggregation Method	N/A	*Dempster-Shafer Theory*

Though it can be seen that statistical/probabilistic approaches can be used to accomplish some of the same analysis tasks that the *Intelligent Hierarchical Decision Architecture* does, they fall short in a number of ways, namely:

- Current OT&E analysis methods use statistical techniques to drawn conclusions at the information level where data are gathered, providing the information to the decision-maker, and requiring that he aggregate the information on his own.

[10] The methods shown in the table in italics have been used in this section with the testbed case data. The final result is essentially the same as that derived from the Intelligent Hierarchical Analysis Structure's methodology.

- Probabilistic-based methods, such as Bayesian Inference or the Dempster-Shafer Theory could be used to combine information and generate information at the operational task level, however, the information needed to calculate these probabilities, in general, does not exist in the OT&E context.

- The methods used within the probabilistic framework to deal with the combinatorial explosion problem require that the measures of functional performance be independent -- an assumption that is strongly violated with the OT&E data.

- The Monte-Carlo simulation technique requires (1) that enough test data be gathered to characterize the statistical distributions from which the random samples are drawn, and (2) a functional transformation from the technical performance level to the task accomplishment level, to allow distributions to be generated at the task level based upon the random draws from the functional performance level distributions.

- When the system performance demonstrated against the logical divisions is diverse, the aggregation method provided by Dempster's Rule of Combination can fail to yield information for the decision-maker.

- No statistical/probabilistic methods exist that would allow the consideration of qualitative data or factors that were not included or controlled during the testing effort.

The advantage to the statistical-based methods appears in the first stage, where well-established techniques can be used to manipulate the raw test data and draw conclusions based upon the data. The statistical methods also provide well-understood means of defining the uncertainty associated with the conclusions, in the form of p-values or confidence intervals. However, from that point forward, the advantage falls to the

Intelligent Hierarchical Decision Architecture's methodology where the ability to easily draw conclusions at higher information levels, deal with qualitative data, draw conclusions when transformations between information levels can only be described heuristically, and bring added realism into the decision-making process make it well suited for the analysis tasks of the operational testing community.

7.1.5 Mission-Level Model Comparison

Current methods used within the OT&E community to provide high-level answers to decision-makers involve the use of mission-level models. These models are used to simulate the engagements between aircraft and threat systems in a representative operational scenario, such as the one shown in Figure 38.

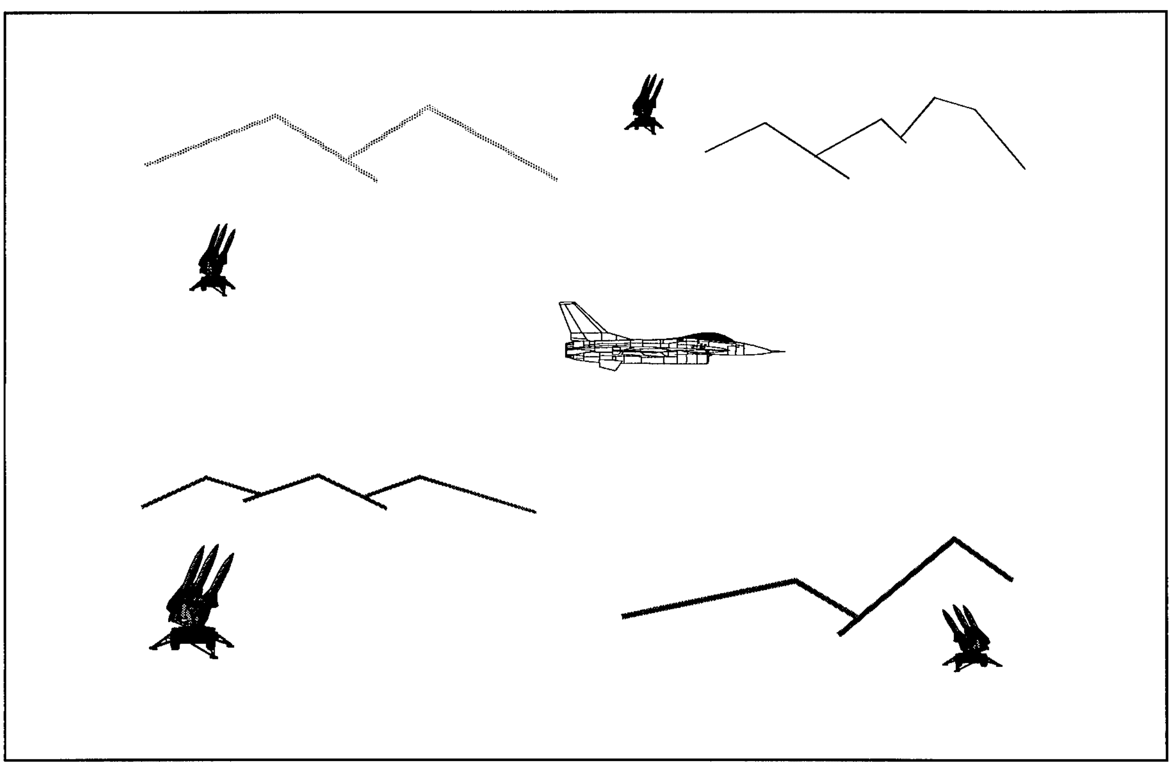

Figure 38 Representative Electronic Combat Operational Scenario

From these simulated engagements, the aircraft's probability of survival can be calculated. By running the engagement, first without the aircraft's jamming system operating, then a second time with the jamming system operating, the model can give a change in the survival rate attributable to the jammer.

The General Effectiveness Methodology (GEM) Model, developed by the Georgia Tech Research Institute, is a mission-level model, widely used throughout the Operational Testing and Electronic Combat communities, to assess mission-level survivability effects due to electronic combat systems. The GEM model will be used here to illustrate that when the GEM is run deterministically (i.e., only the mean value produced by the model is used in making the acquisition decision) that the GEM and the IHDA provide the same solutions. However, when the uncertainties associated with the GEM are included, it will be shown that the GEM provides inconclusive information to the decision-maker, while the IHDA continues to provide a conclusive result.

The General Effectiveness Methodology Model is a data-driven model which simulates the encounters between an aircraft and various surface-to-air missile systems in a representative operational scenario [52]. As the aircraft flies through the scenario, the GEM determines its probability of being killed, and removes the aircraft from the scenario when its probability of kill exceeds a given threshold. When one aircraft is removed from the scenario, a replacement aircraft assumes the original aircraft's position in the scenario. At the conclusion of the scenario the number of aircraft killed is used to calculate an overall mission survivability rate. By running the GEM through the same scenario, first without the jammer functioning (Dry) and then a second time with the jammer functioning (Wet), the model will give a percent change in survivability based upon the jammer's effectiveness.

The GEM will be used in a comparison study is to show that the *Intelligent Hierarchical Decision Architecture* will produce the same results as a "standard" methodology or model when the standard model is run in its deterministic mode. Yet, when "noise" or uncertainty are considered with the standard method, the *Intelligent*

Hierarchical Decision Architecture's method will be able to provide a definitive solution, where the standard method's results are overwhelmed by the uncertainty.

In order to illustrate this phenomenon, the GEM model will first be used to show that the *Intelligent Hierarchical Decision Architecture* provides the same results as when the GEM model is run deterministically. That is, when the decision-maker uses just the mean value generated by the GEM and compares it to the evaluation criterion, the acquisition decision that would be made would be the same as that made with the results from the IHDA methodology. Then, the errors produced in the running of the GEM will be considered. When these uncertainties are taken into consideration, the results from the GEM are found to be inconclusive, while those produced by the *Intelligent Hierarchical Decision Architecture* are conclusive.

7.1.5.1 COMPARISON APPROACH

The GEM model was run using a scenario that included five of each type of the threat systems considered in the IHDA testbed case. The GEM model uses Dry P_k values to generate a number of aircraft kills in the no-jamming case and the Wet P_k values to generate the number of aircraft kills in the jamming case. The two columns of Table 23 labeled Dry P_k and Wet P_k are the input data used to stimulate the GEM model for the control case. In order to facilitate the comparison between the two methods, the *Reduction in P_k* values generated from the second stage of the *Intelligent Hierarchical Decision Architecture*, for each threat type, were used to calculate the Wet P_k (with Jammer-X operating) values based upon a common Dry P_k (without Jammer-X operating) value. The column in Table 23 labeled *% Reduction in P_k* is included to show that the two methods were evaluated from the same input conditions.

Table 23 GEM Control Case Input Values

Threat System	Dry P_k Value	Wet P_k Value	% Reduction in P_k
A-1	0.45	0.180	60%
A-2	0.45	0.135	70%
A-3	0.45	0.090	80%
A-4	0.45	0.045	90%
A-5	0.45	0.000	100%
B-1	0.45	0.315	30%
B-2	0.45	0.270	40%
B-3	0.45	0.225	50%
B-4	0.45	0.180	60%
B-5	0.45	0.135	70%
C-1	0.45	0.405	10%
C-2	0.45	0.360	20%
C-3	0.45	0.3115	30%
C-4	0.45	0.270	40%
C-5	0.45	0.225	50%
D-1	0.45	0.360	20%
D-2	0.45	0.270	40%
D-3	0.45	0.180	60%
D-4	0.45	0.090	80%
D-5	0.45	0.000	100%

After the control case was run, which included all twenty of the threats, subsequent GEM runs were performed where each threat type was removed from the scenario. The removal of the individual threat systems was meant to introduce a bias in the analysis, to see how the two evaluation systems would respond. The evaluation of the system performance with the individual threat systems removed was then accomplished using the IHDA methodology, such that the results generated by the two methods could be compared under identical input conditions.

7.1.5.2 COMPARISON RESULTS

The results from the GEM runs, and the associated results derived using the IHDA for both the control case and the biased conditions, are shown in Table 24. The values given in Table 24 are the *Change in Probability of Survival* (ΔP_s) values with the change from no jamming (Dry) to jamming (Wet) conditions. Assuming a 50% ΔP_s requirement, the YES/NO in each block represents whether or not the Jammer-X system should be acquired based upon the mean value produced using GEM or the location of the Basic Membership Function produced by the *Intelligent Hierarchical Decision Architecture.*

Table 24 GEM Mean Value vs. IHDA Results Decision Comparison

	GEM	**IHDA**
All Threats (*CONTROL*)	60% {YES}	BMF @ 60% {YES}
Without Threat A	43% {NO}	BMF @ 40% {NO}
Without Threat B	57% {YES}	BMFs @ 60-90% {YES}
Without Threat C	71% {YES}	BMF @ 60% {YES}
Without Threat D	57% {YES}	BMF @ 60% {YES}

Table 24 illustrates that if the mean produced by the GEM runs is used in comparison to the evaluation criterion to make the acquisition decision, then the IHDA and GEM solutions "track" each other. That is, both methods would provide the same recommendation to the decision-maker for the five cases evaluated here. However, when the errors associated with the results produced by the GEM are considered, the tracking of the two methods quickly disappears. When the uncertainties associated with the solutions from both methods are considered, the GEM leaves the decision-maker without any information upon which to base his decisions, while the IHDA provides a conclusive result in all the cases illustrated here.

The results from the analyses, including the error bounds generated by the GEM and the distributions of the BMFs generated by the IHDA, are used to generate a color-coded decision chart, illustrated in Table 25. When the GEM error bounds are considered, it can be seen that GEM provides inconclusive results for all the cases, while the IHDA provides conclusive results in all cases. These results were generated using the mean and the error bound to determine the green, yellow, and red region, similar to the method used in current OT&E hypothesis testing efforts (i.e., If the mean and all the error bound is on the "correct" side of the requirement the system gets a green rating. If the mean and the error bound straddle the requirement a yellow rating is given. If the mean and the error bound are completely on the "wrong" side of the requirement a red rating is given.). Table 25 shows that, in this example, the GEM always produces a situation (i.e., a yellow rating) where no decision can be made, whereas, the IHDA provides conclusive results in all cases. The mean and its associated error bound have been included in the table to illustrate the magnitude of the uncertainty associated with the GEM results.

Table 25 Color-Coded GEM / IHDA Comparison

	GEM	**IHDA**
All Threats	YELLOW *60% +/- 26%*	GREEN
Without Threat A	YELLOW *43% +/- 29%*	RED
Without Threat B	YELLOW *57% +/- 26%*	GREEN
Without Threat C	YELLOW *71% +/- 26%*	GREEN
Without Threat D	YELLOW *57% +/- 29%*	GREEN

7.1.6 MISSION-LEVEL MODEL COMPARISON CONCLUSION

This brief example has illustrated that the *Intelligent Hierarchical Decision Architecture* provides the same acquisition recommendation to the decision-maker as the General Effectiveness Methodology Model, when the uncertainties associated with the GEM are not considered. However, when the uncertainties generated by the GEM are included in determining the acquisition recommendation, the GEM provides inconclusive results while the IHDA can continue to provide a useful result. This has illustrated the limitations of the decision-making capabilities of an analysis based upon the use of mission-level models. In addition to the limitations illustrated here, other factors prevent mission-level models from being an adequate solution to the issue that the development of the *Intelligent Hierarchical Decision Architecture* was designed to address. These factors are briefly described below.

- Sensitivity studies using the GEM model were conducted in the process of developing the scenario used in this illustration. In those studies, it was determined that the GEM model is very sensitive to the Dry P_k values chosen for use as the input. The uncertainties associated with generating an absolute P_k value from a testing effort have proven to be very large. Thus, the OT&E and Electronic Combat communities have strived to develop means of analysis that can use relative P_k values (i.e., use a measure of the change in P_k from dry to wet conditions rather than having to measure an absolute P_k value). However, the GEM model requires absolute P_k values as input. This sensitivity to a parameter that is difficult to generate with any certainty limits the model's usefulness as an analysis tool.

- The GEM model does not use the information gathered during the testing effort to generate its results. Rather, it uses digital models of the aircraft, jammer, and threat systems which have been developed to simulate the engagements. These models are not always completely accurate representations of the true hardware due to limitations on the information available to build the models. Therefore, although these mission-level models are included in the analyst's toolbox, they are seldom an adequate substitute for an analysis based upon the data gathered during the testing effort.

- The GEM cannot consider factors that were not included in the model's development. For example, it cannot consider the effect of threats that were not included in the scenario, if no digital model for those threats is available. GEM also cannot be used to examine such factors as operator training, tactics, environment, etc. or the other non-testable factors that were considered through the use of the FCM in the *Intelligent Hierarchical Decision Architecture*. Thus, although mission-level models such as GEM allow a realistic threat scenario to be simulated, analysis methods such as that provided by the *Intelligent Hierarchical Decision Architecture* provide a more realistic assessment of the system-under-test's task-level accomplishment capabilities.

- The running of mission-level models, such as GEM, depends upon having a detailed knowledge of the aircraft dynamics, vulnerabilities, etc. If that information is not available, the model can be run, as it was here, using only the P_k - to - P_s algorithms. However, running the model from that point assumes a knowledge of the P_k values, which is information that is not directly available from the OT&E testing effort. Therefore, the analyst is still faced with determining a method to take the low-level information that is gathered on the test range and aggregating it to a level where it can be used as input to the mission-level model.

7.2 INFORMATION CONTENT MEASURE

In 1949, Claude E. Shannon, working for Bell Telephone Laboratories, devised a measure for indicating the amount of information, choice, and uncertainty in a digital transmission. His measure, based upon the probability of each bit of information having a certain probability of occurrence, p_i, took the form of entropy from statistical mechanics, as [53]

$$H = -K\sum_{i=1}^{n} p_i \log(p_i) \tag{7-2}$$

Within fuzzy set theory, an important measure has become fuzzy entropy, which is a measure of the amount of difficulty or ambiguity associated with making a decision based upon the information contained within a fuzzy set. Most of the work in developing fuzzy entropy measures has begun with the Shannon entropy measure as its departure point. An excellent overview of the work done in developing the myriad of fuzzy entropy measures is provided in [54].

One of the fuzzy entropy measures, introduced in 1972 by DeLuca and Termini, is based upon the Shannon entropy, and is given as [20]

$$H = -K\sum_{i=1}^{n} \mu_i \log(\mu_i) + (1-\mu_i)\log(1-\mu_i) \tag{7-3}$$

where K is a positive constant used as a normalization factor. This measure satisfies the four properties described in Chapter 2, as necessary to be considered an entropy measure.

This measure will be used to assess the amount of information contained in the COMMFFYs of the first three stages of the *Intelligent Hierarchical Decision Architecture*, where fuzzy sets are being used. For the probabilistic information resulting from the final stage -- the aggregation using Dempster's Rule of Combination -- a different measure is needed. There, to measure the entropy contained in the information given to the decision-maker, a formulation that is capable of measuring the entropy provided by the Dempster-Shafer approach is needed. Stephanou and Lu provide the ideal entropy measures based upon the information provided from Dempster's Rule of Combination in [55]. There, they define the *belief entropy*, a measure of the degree of confusion in one's knowledge about the exact fraction of belief that should be committed to each focal element of Θ, as

$$H_b = \sum_{i=1}^{n} m(q_i)\log(q_i) \tag{7-4}$$

The *core entropy*, a measure of the degree of confusion in one's knowledge of which possible subset(s) of Θ the true value of the variable might be in, as

$$H_q = \sum_{i=1}^{n} h(q_i)\log h(q_i) \tag{7-5}$$

where $h(q_i) = \frac{k(q_i)}{\sum_{i=1}^{n} k(q_i)}$ and $k(q_i)$ is the cardinality of the subset minus 1.

The *partial ignorance*, a measure of one's inability to confine the true value of x within a small subset of the frame of discernment, as

$$I = \sum_{i=1}^{n} b(q_i) q_i^{\#} \tag{7-6}$$

where $q_i^{\#} = \frac{k(q_i)}{k(\Theta)}$.

Finally, the generalized entropy, which is the measure used to assess the information content of the final *Intelligent Hierarchical Decision Architecture* result, is given by

$$H = H_b + H_q + \beta I \tag{7-7}$$

where β is a scaling factor.

Since entropy is the measure of the inability to make a decision based upon the available information, the amount of entropy should decrease with a progression through the stages of the *Intelligent Hierarchical Decision Architecture.*

7.2.1 TESTBED CASE ENTROPY RESULTS

Entropy is used as a measure to determine if the information content is moving in the correct direction with a progression through the *Intelligent Hierarchical Decision Architecture.* Using (7-3) to determine the entropy of the first three stages of the *Intelligent Hierarchical Decision Architecture* and (7-7) for the final stage, the entropy result for the *Intelligent Hierarchical Decision Architecture* Best-Case and Worst-Case information are given in Figure 39.

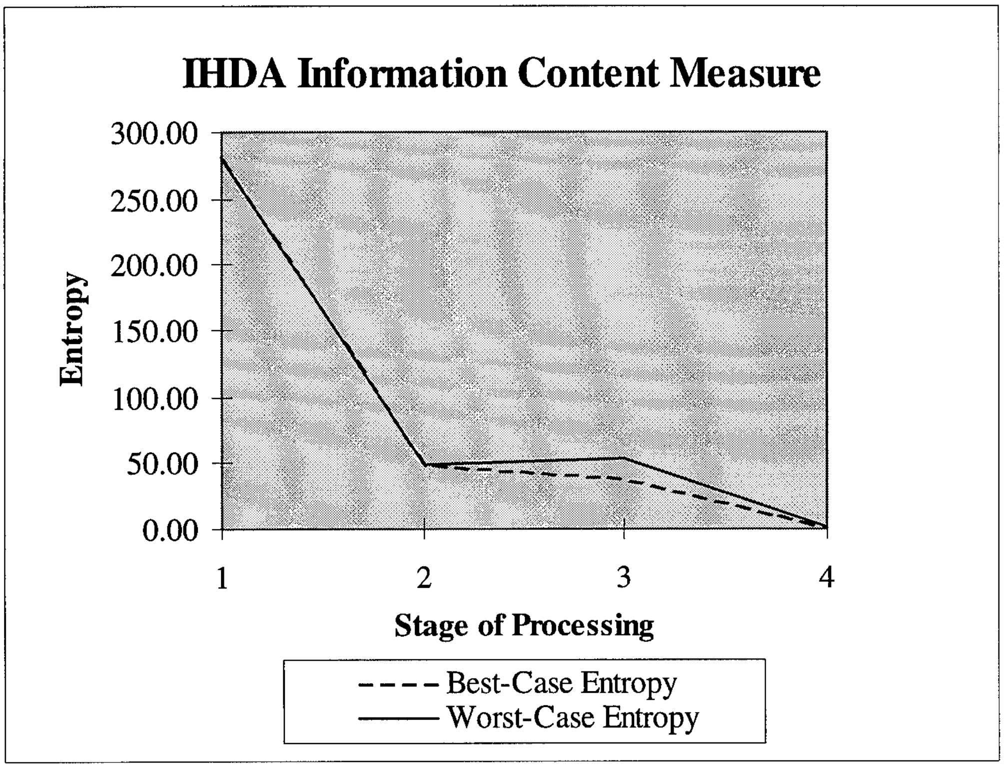

Figure 39 Intelligent Hierarchical Decision Architecture Information Content (Entropy) Measure

From Figure 39, it can be seen that the amount of entropy in the decision-making process is decreasing with the progression through the stages of the *Intelligent Hierarchical Decision Architecture*. The normalization factor, K, in (7-3) was held at unity through the stages. This choice of the factor further emphasizes the concentration and compression of information that is occurring with the progression through the *Intelligent Hierarchical Decision Architecture*.

Also, one can note from the figure that the amount of entropy associated with the best-case adjustment is, as expected, lower than the entropy associated with the worst-case adjustment. This is due to the concentration effect of the exponential adjustment used in the best-case. This illustrates property EP3, discussed in Chapter 2. Finally, it should be noted that the entropy associated with making a decision with the Dempster-

Shafer probabilistic bound information is minuscule in comparison to that of the entropy associated with decision-making with the original raw test data -- the desired result!!

7.3 SUMMARY OF RESULTS

The *Intelligent Hierarchical Decision Architecture* has been developed to address a challenge faced by analysts in the information age. That challenge is to provide information to decision-makers on a system-under-test's ability to accomplish its designated tasks, when testing can only provide information at a functional-performance level. As stated in Chapter One, although this research was spawned by the issues faced by the OT&E community, its results are equally applicable to any analyst who is faced with the task of aggregating low-level data into information that is meaningful for high-level decision-making.

Chapter 1 described the Operational Test and Evaluation process and highlighted the problems with current analysis methods. The problem being solved by this work was introduced, and an overview was presented in the introductory chapter.

Chapter 2 provided an overview of the relevant fuzzy set and fuzzy logic concepts, whose understanding was necessary for the remaining work.

Chapter 3 discussed the ***Clustering Methodology***. In that methodology, first, one of two methods was chosen to define the *Basic Membership Functions (BMFs)*, the fuzzy sets used to define regions of the universe of discourse: either a clustering or heuristic method, depending on the amount of data available to perform this step. Once the BMFs are defined, the *Composite Fuzzy Membership Function (COMMFFY) Compositional Methods* are used to derive a COMMFFY (i.e., a fuzzy distribution) from the observed functional-performance level test data. In the process of determining the COMMFFY that will pass out of this phase of the *Intelligent Hierarchical Decision Architecture* into the next, several *Fuzzy-Statistical Similarity Measures* and an *On-Line Optimization*

Method to optimize the choice among the available compositional methods to be used for a given data set were defined. Thus, the input to the Clustering Methodology was the raw test data, at the functional-performance level and the output was an optimized *Composite Fuzzy Membership Function (COMMFFY)* representing a fuzzy distribution of those data.

Chapter 4 discussed the second phase of the *Intelligent Hierarchical Decision Architecture*, the ***Fuzzy Associative Memory***. The Fuzzy Associative Memory transforms the *Composite Fuzzy Membership Function* at the functional-performance level, developed in the first phase, to a *Composite Fuzzy Membership Function* at the task-accomplishment level. This transformation from one information level to the next, is the first step toward synthesizing the low-level information into high-level information that is more meaningful to the decision-maker. The Fuzzy Associative Memory is a fuzzy rule bank containing the transformation relationships between each input and output variable. Using the result of the Reduction Theorem, the combinatorial explosion problem of dealing with all the interactions among the input variables was avoided. The Fuzzy Associative Memory, in general, can be developed from any information available for the application, including input/output data or expert opinion. The Fuzzy Associative Memory for the testbed case, was developed based upon information derived from a Modeling & Simulation sensitivity study. The study used an engagement-level model to relate the functional-performance level measures to the task-accomplishment level measure. Thus, the FAM served to aggregate the performance across all the functional-performance measures and provide a measure of system performance at the task-accomplishment level. The input to the Fuzzy Associative Memory were the functional-performance level COMMFFYs derived from the Clustering Methodology, and the output was a COMMFFY at the task-accomplishment level.

Chapter 5 described the ***Fuzzy Cognitive Map***, used within the *Intelligent Hierarchical Decision Architecture* to adjust the task accomplishment-level COMMFFY derived from the first two phases. This adjustment took into consideration the factors that could not be controlled or included during the testing phase, but which would have ultimately affected the system performance measure. A Fuzzy Cognitive Map is an expert-created drawing of cause-and-effect relationships. Basic theories on Fuzzy Cognitive Maps suggested by Kosko and other researchers, have been extended in this work to allow them to be used to adjust the system performance derived in the first two phases of the methodology. Using the FCM-based methodology described in Chapter 5, the input to this phase is the task-level COMMFFY derived from the test measurements in the first two phases of the *Intelligent Hierarchical Decision Architecture*, and the output is a task-level COMMFFY adjusted to account for the factors that cannot be included or controlled in the testing effort.

The first three phases of the *Intelligent Hierarchical Decision Architecture*, described thus far, have taken the low-level functional performance measurements gathered in the laboratory or test range and aggregated/synthesized them into a fuzzy distribution at the task-accomplishment level. In addition to simply aggregating the information included in the testing effort, realism has been added to the analysis process by accounting for the effect of factors known to affect the outcome, if those factors could have been controlled or included in the testing, using the Fuzzy Cognitive Map. In the first three stages, the *Intelligent Hierarchical Decision Architecture* has dealt with each individual logical division of the system performance separately. For example, in the testbed case, the Jammer-X performance against each individual threat system it might encounter during operational use, is analyzed. The final stage of the *Intelligent Hierarchical Decision Architecture* aggregates across these logical divisions to form a single, overall system performance bound. In Chapter 6, the ***Aggregation Methodology*** is described, which makes use of the Dempster-Shafer Theory of Evidential Reasoning, and in particular the Dempster's Rule of Combination to combine the adjusted task-level performance information across all the logical divisions of the system performance to provide the final probabilistic bound on the system performance that is provided to the decision-maker.

The illustration of the methodology applied to a Testbed Case are shown throughout the document, with the full results given in Appendix F. There the reader can follow all the steps through the *Intelligent Hierarchical Decision Architecture*, beginning with the raw test data and ending with the probabilistic bound that is provided to the decision-maker.

Finally, here in Chapter 7, a comparison with current analysis methods and proposed methods based solely upon statistical and probabilistic methods was made. With this comparison, it was illustrated that the current methods provide limitations where the *Intelligent Hierarchical Decision Architecture* does not. The comparison with current mission-level models also illustrated the shortfalls of those methods and the strength of the *Intelligent Hierarchical Decision Architecture* in the same situation. Also in this chapter, an information content measure was introduced, based upon the concepts of fuzzy entropy. With this measure, it was illustrated how the inability to make a decision decreases as a progression is made through the stages of the *Intelligent Hierarchical Decision Architecture*. The comparison with current methods and the entropy measure were used to illustrate the merits of the approach. The following section outlines the contributions that this work makes to fuzzy set theory, system analysis, and test and evaluation.

7.4 CONTRIBUTION

Throughout this dissertation, a description of how each segment of the *Intelligent Hierarchical Decision Architecture* contributes to both the areas of fuzzy set theory and systems analysis, from a theoretical perspective, has been provided. In addition to its theoretical contributions, this work serves as the first step in solving the decision-making dilemma faced by decision-makers in the information age. It provides a methodology through which functional-performance level data can be aggregated, synthesized, and adjusted to provide high-level, realistic information to the decision-maker at a level where it is truly meaningful as a decision-making aide.

The current analysis methodologies used by the OT&E community were described in Chapter 1. These methods provide a mechanism for summarizing system performance data observed on the test range or in the laboratory and making statistical statements on the relevance of the measurements. No satisfactory method has been developed to aggregate and synthesize the gathered data into relevant information until this work. A comparison with proposed statistical/probabilistic methods was conducted in this chapter, where the benefits of the *Intelligent Hierarchical Decision Architecture* were highlighted. In addition to those benefits, in a broader perspective, the *Intelligent Hierarchical Decision Architecture* provides

- The *Intelligent Hierarchical Decision Architecture* provides information to assist the decision-maker at a level where it is meaningful to the decision being made. Current methods provide a statistically-based pass/fail determination at the functional-performance level and leave the job of assimilating the information to a relevant information level up to the decision-maker.

- Using the iterative process inherent in the development of the *Intelligent Hierarchical Decision Architecture*[11], the architecture provides the potential to make decisions earlier in the T&E process, than would normally be possible. This early decision-making capability is a valuable tool as a potential for saving testing dollars, or for

[11] By the iterative approach, we are referring to how: (1) the BMFs may be developed heuristically, initially, then subsequently refined using the clustering method described in Chapter 3 as more low-level data are gathered on the system's performance; (2) the FAM rules may be developed heuristically, initially, and refined as a better understanding of the MOFP-to-MOTA transformation is gained through further testing or M&S studies; (3) the FCM provides a natural mechanism to combine information from diverse sources, as more relevant adjustment factors are discovered; and (4) the Dempster-Shafer Theory used within the Aggregation Methodology allows more evidence to be combined while maintaining the information already considered.

when a statement must be made on a system's potential early in the program, based upon a small amount of testing data[12] [56].

- Finally, not only does the *Intelligent Hierarchical Decision Architecture* provide for early decision-making capabilities, it also will provide direction for M&S studies, which in the past have been conducted for seemingly unknown reasons. With the targeted objective of refining the FAM rules or defining the level of impact of a factor in the FCM, future M&S efforts will be more focused.

With this work, a move toward solving the decision-making dilemma faced by decision-makers in the information age has been made. The door has been opened to the use of intelligent techniques in the systems analysis arena. In addition to the contributions made to the systems analysis arena, the state of the art in several aspects of fuzzy set theory has been advanced. These contributions were discussed throughout the dissertation, as they were introduced, and are summarized briefly below.

- The *Clustering Methodology's Compositional Methods* represent a new method for representing test data in terms of a possibilistic, or fuzzy, distribution.

- The *Fuzzy/Statistical Similarity Measures* developed to assess the optimal compositional method to be used within the clustering methodology offer a new means of comparing possibilistic and probabilistic distributions.

- The *Fuzzy Associative Memory* developed here handles fuzzy distributions as input and output, where current FAMs can only deal with individual data points.

---

[12] For example, Congress mandated an Early Operational Assessment (EOA) of the C-17 transport plane based upon only 50 hours of flight testing data. The resulting information from the C-17 EOA was a rather benign answer, for no definitive result could be derived using current analysis methods. The IHAS would have allowed a more definitive answer on the system performance at this early stage, thus providing a more satisfying result to the Congressional inquiry.

- The *Fuzzy Cognitive Map* used here to adjust performance indicated during the testing effort advances the current uses, which are limited to answering what-if questions in a binary fashion.

- The *Aggregation Method* making use of Dempster's Rule of Combination brings the final solution back to the probabilistic realm, where decision-makers are more comfortable.

- The fuzzy-entropy-based *information content measure* used to assess the adequacy of the overall analysis structure combines two previously diverse areas to yield a single measure.

7.5 FUTURE RESEARCH DIRECTIONS

This work has served to open the world of fuzzy set theory and intelligent techniques to the world of systems analysis, particularly to the analysis of Operational and Test and Evaluation data. With this initial work completed, the potential applications of intelligent techniques to this arena should continue to be explored. In the course of this research, many interesting extensions surfaced, which should now be pursued using the work outlined here as the foundation.

First, application of intelligent techniques to develop a ***parallel hierarchical structure*** that would allow information from both field testing and modeling and simulation to contribute to the decision-making process simultaneously, should be pursued. In the current era of diminished research and development budgets, both within the Department of Defense and within the technical community in general, the systems analysis community is under increasing pressure to use M&S as an enhancement to, and a substitute for, expensive field testing efforts. Current efforts have been limited to conducting field testing to address certain testing objectives and to using modeling and simulation to address untestable objectives, without an effort to use both sources of information to complement each other in answering a single analysis objective. A method is needed through which both sources of information can be combined to provide a *richer source of information* for the decision-maker. The research effort would involve developing an ***intelligent parallel hierarchical structure*** through which field test data analyzed using the *Intelligent Hierarchical Decision Architecture* described in this work, and M&S results from currently-used models, beginning at the same initial conditions, are compared and coupled. Conflict resolution could be accomplished through the use of the Dempster-Shafer theory and a Fuzzy Cognitive Map could be used to meld the information from both the field testing and the model to produce decisions based on both sources of information. In addition to the development of the structure for conflict resolution and information melding, this effort would require the extension of the information content metric developed in this work. That measure would quantify the information content of the testing and modeling results, and be used to determine if adequate information exists to make informed decisions. Further, the metric could be used to determine if gathering further information would move the information content measure in the correct direction.

Second, application of ***intelligent neuro-fuzzy decision models*** to the control loops in currently used missile engagement models to more adequately model human operator responses would drastically increase the accuracy and realism of those models. Current missile engagement models, such as ESAMS, do not adequately model the threat system response under jamming conditions for a number of reasons. One of those reasons is that no mechanism is included in the model to replicate the human operator's decision process of taking control away from the automatic controller when indications warrant that the guidance system is being jammed. Fuzzy logic decision modeling offers an ideal mechanism for modeling the actions the threat system operator would take under various electronic combat conditions. The fuzzy logic controller that models this decision process could be developed by melding data and expert opinion to build the fuzzy rule base. The system could be built based upon *data measurements* of human operator response already collected by the Armstrong Aeromedical Laboratory and *expert opinion* provided by the ESAMS developer, Dr. Sam Baty at The BDM Corporation. Once the fuzzy logic controller has been successfully implemented, it can be included within the current missile engagement models, thus improving the model's performance and making it more adequate for use in analysis of electronic combat systems.

Finally, development of a neuro-fuzzy mechanism to model differences in threat system performance against electronic combat systems, based upon differing alignments of the components within the threat system would dramatically increase the reliability of conclusions drawn from testing and analysis efforts. Current efforts sponsored by the Susceptibility Modeling and Range Test (SMART) Program aimed at Verification, Validation & Accreditation of engagement models in an electronic combat environment have revealed that the performance of the jamming system is dependent on the alignment of the threat system (i.e. the tuning of various filters and components within the threat system). Although this phenomenon has been observed, and is a hindrance to the development of robust engagement models and jamming techniques, no mechanism for quantifying the effects of the differing alignments on the threat system has been established. ***Intelligent neuro-fuzzy techniques*** provide an ideal mechanism for the development of a means to quantify, in fuzzy terms, the effects of differing alignments and the resulting effects on the jammer effectiveness against the threat system. Once the mechanism for quantifying and modeling the alignment effects has been developed, it can be applied to the development of robust and adaptive jamming techniques.

These three research projects would extend the work started here and extend the reach of intelligent techniques further into the realm of the systems analysis arena.

APPENDIX A

CURRENT ANALYSIS METHODS

This appendix provides an overview of the OT&E community's current analysis techniques, broken into three broad categories -- Statistical Analysis Tools, Statistical Model Building Techniques, and Modeling and Simulation.

A.1 STATISTICAL ANALYSIS TOOLS

Standard statistical methods, including *Hypothesis Testing, Decision Theoretic Approaches, Bayesian Analysis, Analysis of Variance,* and *Experimental Design* [57] are the most widely used T&E analysis tools today.

A.1.1 HYPOTHESIS TESTING

The classical *Hypothesis Testing* approach [58] views the outcome of each test event as a realization of a random variable. A parameter, θ, representing real system performance, is estimated as $\hat{X}$, through observation of the test measurements $x_1, x_2, \ldots, x_N$. A hypothesis is built showing that the parameter does or does not lie within a certain interval, usually built around the requirement of interest, for example

$$H_0 : \theta \geq T_R$$
$$H_1 : \theta < T_R \qquad \textbf{(A-1)}$$

where T_R is a system requirement. The test data are used to prove or disprove the null hypothesis, H_0. Two types of error are possible when conducting hypothesis testing -- rejecting the null hypothesis when it is really true, "throwing away a diamond," and accepting the null hypothesis when it is really false, "buying a lemon" [59]. These errors are termed Type I and Type II errors respectively, and are quantified by risk measures α, called the size of the test, and β, the power of the test. The value chosen for α determines the amount of risk associated with making a Type I error, referred to as the producer's risk, and is used to determine the size of the test. Once the risk the producer is willing to take is determined, the decision criteria value, C_1, is set based upon

$$\alpha = \Pr[\hat{X} < C_1 | \vartheta \geq T_R] \qquad \textbf{(A-2)}$$

Once the value for α is chosen, the value for β, the consumer's risk, is determined -- the tester can only control one type of error using this approach.

The *Decision Theoretic Approach* allows the "cost" associated with making both a Type I or Type II error to be considered in the test design, therefore optimizing the test from both the producer's and consumer's perspectives. This is accomplished by constructing a decision rule minimizing the risk, or expected loss, defined as

$$risk = \alpha L_1 + (1- \beta) L_2 \qquad \textbf{(A-3)}$$

where L_1 and L_2 are the costs associated with making Type I and Type II errors, respectively.

The *Bayesian Approach* offers a method of incorporating previously gathered information, or *a priori* information, when building the *posterior* distribution used for decision making at the conclusion of testing [60]. The Bayesian framework for hypothesis testing expands upon the Decision Theoretic Approach, by incorporating the use of the prior information. The Bayes' decision rule is to reject H_0 if

$$\hat{X}_{BAYES} < C_{BAYES} \tag{A-4}$$

where C_{BAYES} is chosen to satisfy

$$L_1 * \Pr[\hat{X}_{BAYES} < C_{BAYES}] = L_2 * \Pr[\hat{X}_{BAYES} \geq C_{BAYES}] \tag{A-5}$$

The danger associated with the use of a Bayesian Approach is in the choice of the prior distribution. An improper choice can bias the results. For example, in the choice of a prior distribution that does not include the true parameter's value, no matter how much further testing is done, the posterior distribution will never converge to the true value. On the other hand, if the prior distribution contains the true value, but is concentrated away from it, additional testing data will eventually compensate for the prior distribution and converge on the true value, but at a cost of many additional test events [57].

Finally, in an effort to limit both types of risk, a variation of the standard *Hypothesis Testing* method using a three-part decision region, was adopted by the OT&E analysis community. A three-part decision region can be constructed that limits both the consumer's and producer's risks to acceptable values and creates an undetermined region in the middle. Using this method, two decision criteria are developed, one limits the producer's risk and one limits the consumer's risk, then the decision rule becomes

$$\text{If } \hat{X} > C_1 \text{, Accept } H_0$$
$$\text{If } \hat{X} < C_2 \text{, Reject } H_0 \qquad \textbf{(A-6)}$$
$$\text{If } C_2 \leq \hat{X} \leq C_1 \text{, Undetermined}$$

The *critical value*, C_2, is defined as the minimum acceptable value of a performance criterion, the distribution of critical value observations forms the critical distribution. The *objective value*, C_1, is a "nice to have" value of system performance that the system user's would like the system to achieve. The two distributions' means are separated by some distance, *D*. Assuming the use of a normal distribution, a *Z*-value is defined associated with the desired producer's and consumer's risk. Knowing the actual mean and variance of the critical and objective distributions allows the calculation of the sample size required to eliminate the undetermined region, as [4]

$$N = ((\frac{Z_\alpha + Z_\beta}{D})(\frac{\sigma_C + \sigma_O}{2})^2 \qquad \textbf{(A-7)}$$

Assuming a knowledge of the true system variance, a sample variance can be calculated using

$$\sigma_{Sample} = \frac{\sigma_{True}}{\sqrt{N}} \qquad \textbf{(A-8)}$$

where *N* is the sample size. To satisfy the two risk values, two evaluation criteria must be established, one associated with the producer's risk and the objective distribution, TH_α, representing a "reject if below value" and one associated with the consumer's risk and the critical distribution, TH_β, an "accept if above" value. These criteria are determined using

$$TH_{\alpha} = \mu_O - Z_{\alpha}\sigma_{O(True)} \tag{A-9}$$

$$TH_{\beta} = \mu_C + Z_{\beta}\sigma_{C(True)} \tag{A-10}$$

Although this method appears to be a good compromise between consumer's and producer's risk, the undetermined rating region has caused major problems for the OT&E community. Because the testing involved in OT&E typically involves very small sample sizes, the sample variances are very large (see equation (A-10) for the relationship of sample size and sample variance), therefore, the undetermined region is very large.

Table A-1 illustrates the problem with the undetermined rating region. If enough testing is not accomplished to make the sample variances small enough, a region where no performance assessment can be made results. The example in Table A-1 is for a system with $\mu_C = 26.3$, $\mu_O = 28.2$, $\sigma_C = 1.6$, and $\sigma_O = 1.5$; consumer and producer both want 5% risk; therefore, $Z_{\alpha} = Z_{\beta} = 1.645$.

Table A-1 Sample Size and Undetermined Rating Region Correlation

Sample Size	TH_{α} "Reject if Below"	TH_{β} "Accept if Above"	Undetermined Region Size
1	25.73	28.93	3.2
2	26.45	28.16	1.7
4	26.96	27.62	0.65
6	27.19	27.37	0.18
8	27.32	27.23	0

Table A-1 illustrates the potential for a test program to conclude without the test community being able to make any claims about system performance. Such was the case on the B-1B Defensive Avionics System OT&E program [61]. During that testing, the undetermined region played a major role in the inability of analysts to draw conclusions on system performance. In testing to measure the Percentage of Threat Detected, six threats were tested using over 500 test events. All events exceeding the critical requirement, yet the measure was rated as Undetermined. The measurement of Percentage of Threat Identified, experienced the same condition -- over 500 events exceeding the requirement, resulting in an Undetermined rating. Finally, testing to measure Percentage of Correct Mode ID, tested six threats using over 500 test events with 5/6 exceeding the requirement, and again the result was an Undetermined rating. After this experience, the Air Force OT&E community adopted a practice of limiting its statistical analysis to calculating the mean of the test measurements and comparing it to the system requirement; making a pass/fail determination based upon which side of the system requirement the test sample mean lies.

As a comparison of the methods discussed so far, Table A-2 shows the analysis of the same data using the different methods [57]. The example is an analysis of a system having a true probability of 0.70 and a system requirement of 0.75. Ten test events were conducted and the data were analyzed using the methods shown above against a hypothesis of

$$H_0: P > 0.75$$
$$H_1: P \leq 0.75$$

Table A-2 Decision Probabilities Using Various Statistical Approaches

Approach	**Probability of:**		
	Accept H_0	**Undetermined**	**Reject H_0**
Classical Hypothesis Testing	0.90	0	0.10
3-Part Hypothesis Testing	0.02	0.88	0.10
Decision Theoretic	0.47	0	0.53
Bayesian	0.21	0	0.79

It should be noted that the correct decision for this example would have been to reject H_0, because the actual system's performance is below the requirement. The 3-Part

Hypothesis Testing Method gives an unacceptably large probability of an "undetermined" rating, which has been the experience of the Air Force's OT&E agency on many large and expensive test programs, causing an abandonment of this approach. Further, the Bayesian Approach although most likely to produce the correct decision, has not been used extensively within the OT&E community due to the lack of information available to construct the prior distributions required by the approach. Finally, it should be noted that the Classical Hypothesis Testing approach, most often used by the OT&E community, would most likely result in the *wrong* decision being made.

A.1.2 ANALYSIS OF VARIANCE

Analysis of Variance (ANOVA) is a statistical technique for analyzing measurements when several factors are operating simultaneously, to determine which are important, and to estimate the magnitude of the effects [62]. The ANOVA model is a linear combination of the quantities being examined, plus an error term, such as

$$y_i = x_{1i}\beta_1 + x_{2i}\beta_2 + \ldots + x_{pi}\beta_p + e_i \qquad \textbf{(A-11)}$$

where the x_{ij} are indicator variables, taking on the values 0 or 1, and the β_i are the effects. An ANOVA can be carried out to examine the effect of a single variable on the outcome of a process, to examine the effects of blocking variables, or to look at the effects of several variables and their interactions. By systematically examining each factor and comparing its effect to the variance in the process, conclusions can be drawn on its significance to the experimental outcome.

The single variable analysis, or one-way ANOVA, examines the effect of variations in a single factor, for example, the effect of different aircraft altitudes on bombing accuracy. The model is [63]

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij} \qquad \textbf{(A-12)}$$

where μ is the overall mean, τ_i is the treatment effect, and ε_{ij} is the error term. The analysis tests the hypothesis that all the treatments means are statistically equal, or

$$H_0\text{: } \tau_i = 0, \ \ \forall i \qquad \textbf{(A-13)}$$

against the alternative hypothesis that one of the treatment means differs from the rest. The data are gathered, then a mean for each treatment and an overall mean is calculated. The sum of squares of the treatment, *SStr*, sum of squares of the error, *SSE*, and the total sum of squares, *SST*, are calculated using the formulas

$$SStr = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\bar{y}_{i.} - \bar{y}_{..})^2 \qquad \textbf{(A-14)}$$

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - y_{i.})^2 \qquad \textbf{(A-15)}$$

$$SST = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{..})^2 = SStr + SSE \qquad \textbf{(A-16)}$$

where y_{ij} are the individual observations, $\bar{y}_{i.}$ are the treatment means, and $\bar{y}_{..}$ is the overall mean of all the observations taken together. If there are k treatments, n_i observations of each treatment and N observations overall, the mean squares of the treatment and error are calculated as shown in Table A-3.

Table A-3 One-Way ANOVA Table

Source of Variation	Sum of Squares (SS)	Degrees of Freedom (dof)	Mean Square (MS)	f_0 (F-statistic)
Treatment	SStr	k-1	$MStr = \dfrac{SStr}{k-1}$	$f_0 = \dfrac{MStr}{MSE} \sim F_{(k-1),(N-k)}$
Error	SSE	N-k	$MSE = \dfrac{SSE}{N-k}$	
Total	SST	N-1		

The f_0 value is compared to the value derived from an F-distribution table using the desired level of confidence and the degrees of freedom associated with the treatment and error. If f_0 exceeds the F-distribution value, the hypothesis is rejected. This indicates that at least one of the treatment means differs statistically from the overall mean. If the hypothesis is rejected, various methods are available to determine which treatment(s) differ from the rest, such as Fisher's Least Significant Difference Method or the Duncan Multiple Range Comparison Test. Those methods are not discussed here.

As an example of the one-way ANOVA technique, consider the testing of an aircraft-mounted jamming system designed to generate track error within a threat missile's guidance system, causing the missile to miss hitting the aircraft. The measure used to gauge performance of the jamming system is Missile Miss Distance, a measure of the closest point of approach of the missile to the target aircraft. During the testing, three different jamming techniques are used against a threat missile system, and the resulting miss distances are measured as shown in Table A-4. {NOTE: The values shown in Table A-4 were created to illustrate the methodology, they do not represent the performance measures of any actual system.}

Table A-4 Miss Distance Generated by Various Jamming Techniques

Jamming Technique #1	Jamming Technique #2	Jamming Technique #3
49 31 41 26 52	54 48 36 53 45	59 51 58 45 53
39 46 40 37 58	31 49 42 46 44	41 50 44 38 56
43 34 54 28 48	41 51 45 59 39	68 47 64 32 55
40 22 32 35 45	50 33 47 43 57	50 42 62 36 49
	27 37 40 46 63	
$\bar{y}_{1.} = 40$	$\bar{y}_{2.} = 45.04$	$\bar{y}_{3.} = 50$

The sum of squares calculation is shown below where $\bar{y}_{..}$, the overall mean, is equal to 45.01.

$SStr = 20(40\text{-}45.01)^2 + 25(45.04\text{-}45.01)^2 + 20(50\text{-}45.01)^2 = 1001.1$

$SSE = (49\text{-}40)^2 + (31\text{-}40)^2 + \ldots + (49\text{-}50)^2 = 5321.0$

The resulting ANOVA table is shown in Table A-5

Table A-5 Miss Distance ANOVA Table

Source	**SS**	**dof**	**MS**	**f_0**
Jamming	1001.1	2	500.6	5.832
Error	5321.0	62	85.82	
Total	6322.1	64		

Using a 5% confidence level, $F_{2,62} = 3.07$. Since $f_0 > F_{2,62}$ at least one of the jamming techniques has a significant effect on missile miss distance. The techniques mentioned above (Fisher's Significant Difference Method or Duncan Multiple Range Test) are used to determine which technique is significant.

A.1.3 DESIGN OF EXPERIMENTS

Experimental Design techniques are used within the OT&E community as a systematic test planning tool. A Full Factorial Design, or orthogonal array, examines each factor of interest, at each possible level, as a separate test run [64]. Once the test runs defined in the Full Factorial Design have been accomplished, there are various methods used to determine which factors have a significant outcome on the effect being measured. In most cases in the OT&E context, a Full Factorial Design will design a test with many more test runs than can be accomplished due to test resource constraints. Therefore, the Fractional Factorial Design techniques are used to systematically reduce the size of the test. In most cases, the OT&E analyst makes use of one of the many pre-published Fractional Factorial Design schemes to reduce the size of the test. The example that follows shows the Fractional Factorial Design for the Joint Standoff Weapon (JSOW), currently in the test concept development phase at the Air Force Operational Test and Evaluation Center. In this example, the factors of interest are defined and a test matrix is developed that defines the levels of each factor in each test run. Once the test runs have been accomplished, the data will be analyzed in accordance with (A-17) through (A-20). These equations give the result on the outcome of the test based upon each of the main effect factors, which is subsequently used to determine if that factor is a significant effect on the experimental outcome.

A subset of the JSOW operational users' requirements, those factors important for the system to be able to perform its intended mission, are listed below [65].

- Capable against fixed, relocatable, and moving targets
- Usable day, night, and in or through adverse weather
- Low or high altitude capability from outside target point defenses
- Targeting options must allow for preplanned, inflight self targeting, or inflight third party

From these users' requirements, four test factors were defined: target type (fixed, relocatable, moving), day type (day, night, adverse weather), launch altitude (low, medium, high), and targeting option (preplanned, self-targeting, third-party). A Full Factorial Design of four factors, each at three levels, would require 3^4, or 81 test runs, clearly too many test events when a half million-dollar missile is expended on each run. Using a 3^{4-2} design [66], the number of test runs can be reduced from 81 to nine. Using run numbers 1, 14, 27, 33, 43, 47, 62, 66, 76 from the Full Factorial Design matrix, the Fraction Factorial Design test matrix is shown in Table A-6 [67].

Table A-6 Orthogonal Array Design Matrix for JSOW Testing

Run Number	**Factors**			
(y_i)	**Target Type**	**Day Type**	**Launch Altitude**	**Targeting**
1 (FFD #1)	Fixed	Day	Low	Preplanned
2 (FFD #14)	Fixed	Night	Medium	Self
3 (FFD #27)	Fixed	Adverse Weather	High	Third-Party
4 (FFD #33)	Relocatable	Day	Medium	Third-Party
5 (FFD #43)	Relocatable	Night	High	Preplanned
6 (FFD #47)	Relocatable	Adverse Weather	Low	Self
7 (FFD #62)	Moving	Day	High	Self
8 (FFD #66)	Moving	Night	Low	Third-Party
9 (FFD #76)	Moving	Adverse Weather	Medium	Preplanned

The measure used in this testing will be the number of target kills per weapon at each experimental condition. The analysis for this test will proceed as follows.

Step 1, Main Effect Estimation: As the data are collected, each is given the designation y_i corresponding to the Run Number in Table A-6. From these data, main effect estimations are calculated as shown below.

Target type main effect calculations:

$$\hat{\theta}_{TT,Fix} = \frac{y_1 + y_2 + y_3}{3}, \hat{\theta}_{TT,Rlt} = \frac{y_4 + y_5 + y_6}{3}, \hat{\theta}_{TT,Mvg} = \frac{y_7 + y_8 + y_9}{3} \qquad \textbf{(A-17)}$$

Day type main effect calculations:

$$\hat{\theta}_{DT,Day} = \frac{y_1 + y_4 + y_7}{3}, \hat{\theta}_{DT,Ngt} = \frac{y_2 + y_5 + y_8}{3}, \hat{\theta}_{DT,Adv} = \frac{y_3 + y_6 + y_9}{3} \qquad \textbf{(A-18)}$$

Launch Altitude main effect calculations:

$$\hat{\theta}_{Lnch,Lo} = \frac{y_1 + y_6 + y_8}{3}, \hat{\theta}_{Lnch,Med} = \frac{y_2 + y_4 + y_9}{3}, \hat{\theta}_{Lnch,Hi} = \frac{y_3 + y_5 + y_7}{3} \qquad \textbf{(A-19)}$$

Targeting Option main effect calculations:

$$\hat{\theta}_{TO,PP} = \frac{y_1 + y_5 + y_9}{3}, \hat{\theta}_{TO,Self} = \frac{y_2 + y_6 + y_7}{3}, \hat{\theta}_{TO,TrdP} = \frac{y_3 + y_4 + y_8}{3} \qquad \textbf{(A-20)}$$

Step 2, Calculate Statistical Significance of Estimates: The main effect estimates are the numbers calculated above. Main effect plots can be made to give an informal, visual perspective on the factor effects. For a more rigorous treatment, the Lenth's Method will be used to determine significance of the factor effects on the test outcome, as shown in equations (A-21) through (A-23) below.

First, place the absolute values of the effect estimates in ascending order. Compute an initial estimate of the sample error as 1.5 multiplied by the median of the effect estimate absolute values.

$$s_{\hat{\theta}_{init}} = \sigma_{\hat{\theta}} = 1.5 * median(|\hat{\theta}_i|) \qquad \textbf{(A-21)}$$

Compute a final estimate of the sample error as 1.5 multiplied by the median of the effect estimate absolute values that are less than 2.5 times the initial estimate found above.

$$s_{\hat{\theta}_{final}} = \sigma_{\hat{\theta}} = 1.5 * median(|\hat{\theta}_i| < 2.5 * s_{\hat{\theta}_{init}}) \qquad \textbf{(A-22)}$$

The effect is significant if its estimate absolute value is greater than the appropriate t-distribution statistic times the sample error.

$$|\hat{\theta}_i| > (t_{\frac{\alpha}{2},\frac{m}{3}}) * (s_{\hat{\theta}_{final}}) \qquad \textbf{(A-23)}$$

<u>*Step 3, Verify Interaction Effects Assumption:*</u> In using such a highly fractionated test matrix, an assumption must be made that there are no interactions between the factors. As a final step in the analysis process, the validity of that assumption must be checked. Using the additive model shown below, calculate the prediction for each factor setting.

$$\theta(TT_i, DT_j, Lnch_k, TO_l) = \mu + \hat{\theta}_{TT\,i} + \hat{\theta}_{DT\,j} + \hat{\theta}_{Lnch\,k} + \hat{\theta}_{TO\,l} \qquad \textbf{(A-24)}$$

where μ is the overall mean, and the effect estimates at the designated factor level are represented by the $\hat{\theta}$ terms. If the prediction is close to the observed value, interactions among the factors are not significant. If the prediction is far from the value observed in testing, an important interaction may be present that has not been accounted for in the test measures taken.

The statistical methods discussed in this section form the basis of the mathematical analysis techniques used by the OT&E community. These methods depend on several assumptions for a successful analysis; such as, having large sample sizes, knowing the underlying statistical distribution of the data, and being able to separate experimental effects. Most of these assumptions are violated in OT&E. The National Research Council's Panel on Statistical Methods for Testing and Evaluating Defense Systems stated the following [68].

> **... our visit to Fort Hunter Liggett[13] made it clear that routine application of standard experimental design practice will not always be feasible given the constraints of available troops, scheduling of tests, small sample sizes, and the weighting of test environments.**
>
> **Furthermore, the panel believes that the hypothesis testing framework of operational testing is not sensible. The object of operational testing should be to provide to the decision-maker the data most valuable for deciding on the next course of action. The next course of action belongs in a continuum ranging from complete acceptance to complete rejection. Therefore, in operational testing one should concentrate on estimation procedures with statements of attendant risks. We also plan to explore the utility of other methods for combining information for purposes of evaluation, including hierarchical modeling.**

Not only are the assumptions for properly using these statistical methods violated in OT&E, but as the National Research Council's comments highlighted, these methods are not adequate to provide the information that the decision-maker wants as an output from the OT&E process. All of the statistical methods described above are limited to using gathered test data to confirm or deny a hypothesis or effect that can be observed during the testing. None of the methods just described provide a means to extrapolate to higher information levels.

[13] An Army test range used for Force-on-Force exercises and testing activities.

A.2 STATISTICAL MODEL BUILDING TECHNIQUES

Two methods of *Statistical Model Building* are currently used in the OT&E community: *Time Series Analysis* and *Regression Analysis*.

Time Series Analysis is the study of data gathered from a process over time. For example, the readings of a gauge controlling a process taken at one minute intervals, the daily closing prices of the stock exchange, or a measure of yearly crop yields. The purpose of time series analysis is to model the stochastic mechanism giving rise to an observed series of data and to predict or forecast future values of the series based upon historic data from the series [69].

Time Series Analysis techniques are based upon building models from the previously observed data. These models can fall into one of three categories: moving average (MA) models are a weighted linear combination of the present and past terms of a white noise process, autoregressive (AR) models are a linear combination of the most recent values of itself plus an "innovation" term that incorporates everything new in the series not explained by the previous observations, and the mixed autoregressive-moving average (ARMA) model is a linear combination of past observations of the series and of the white noise terms. These models are built under the assumption that the underlying process is stationary, (i.e. its statistical characteristics do not change with time). Nonstationary processes are modeled using differencing techniques, subtracting previous observations from each other in an attempt to form a stationary process, generating integrated autoregressive-moving average (ARIMA) models. The amount of overlap used in the differencing operation is referred to as the lag and usually is chosen because of a physical significance associated with the data. For example, airline passenger counts exhibit a cyclical pattern corresponding to the time/season of the year. A time series model based upon monthly passenger counts might best be treated as an ARIMA model with lag 12.

The techniques of *Time Series Analysis* are used to build models of the observed test data in the OT&E context, and then subsequently used to predict future performance. These techniques are especially useful for modeling such phenomena as missile trajectories or daily message traffic flows through a communication system. However, the assumption of stationarity is strongly violated by OT&E data that is gathered in a dynamic environment where few processes can be classified as stationary. For example, if the missile trajectory being modeled is subjected to electronic countermeasures, the actual missile flight path would likely deviate from its normal trajectory. Modeling these types of dynamics in a time series model would be difficult, if not impossible.

Regression Analysis is a statistical tool that is used to build relationships between two or more quantitative variables such that one variable can be predicted from the other(s) [70]. In regression modeling, a regression line, curve, or surface is fit to the data using techniques to minimize the mean squared error between the observations and the function. Once the model has been developed, it is used to predict the value of the dependent variable based on the value(s) of the independent variable(s).

The relationships described by regression models have been used effectively for the analysis of OT&E data. A model is built using early test data, or the desired performance of the system given in specifications or user's requirements, then subsequently used to compare predicted performance to actual tested performance of the system. One recent use of *Regression Analysis* in OT&E was demonstrated in the area of human factor performance, where system performance factor measurements were used to build a regression model, subsequently used to predict human factor effects [71].

Although *Time Series Analysis* and *Regression Analysis* provide a means of using system performance information to build models of future system performance, they suffer the same shortfalls as other statistical techniques, the inability to extrapolate to higher-level performance measures or to conditions that were not used in building the model.

A.3 AGGREGATION METHODS

The statistical and statistical model building techniques discussed thus far offer means for summarizing or modeling system performance seen on the test range or in the laboratory. None of these statistical methods provide a mechanism for aggregating the collected data into more meaningful information. Aggregation methods used in OT&E programs to date have been ad-hoc, or worse, non-existent. Three recent examples of ad-hoc aggregation schemes developed to provide information to the decision-maker at the appropriate level are from the Milstar Satellite Communication System, the F-15E Tactical Electronic Warfare System (TEWS), and the Cheyenne Mountain Upgrade's Alternate Processing and Correlation Center test programs.

In the OT&E of the Milstar Satellite Communication system, an aggregation methodology was developed through which any failure of what was termed a "critical" measure would cause the failure of the entire system. If other than critical measures failed, a group of test team members got together to make a judgment on the relevance to system performance. The final pass/fail determination was the only information provided to the decision-maker [72].

In the F-15E TEWS Operational Assessment, a color scheme was used through which system performance against each threat was given a green, yellow, or red rating. Then overall system performance was determined by the test team by looking at the spectrum of colors and making a judgment on overall system performance [73].

The Cheyenne Mountain Upgrade's Alternate Processing and Correlation Center program provides another example of an ad-hoc aggregation methodology. During the testing, measurements were taken at the functional-performance level and each was assessed against its individual requirement, and a pass/fail determination was made for each. In the final assessment, at the Measure of Effectiveness level (one information-content level above the functional performance level) the test team looked at the pass/fail results, considered "other factors" observed during the testing, and made a value judgment on the system's effectiveness. These assessments were reported in the final report to the decision-maker [74].

In other OT&E programs, no attempt has been made to aggregate system performance further than the individually-measured functional performance level -- the individual results are reported and the aggregation method is left up to the decision-maker. Clearly, a more systematic and standardized method for aggregating the low-level data into high-level information is needed. With a systematic approach, the decision-maker will be able to come to expect information at a level that is meaningful to him, and he will not have to struggle to understand the differing methodologies used by each individual program.

A.4 MODELING AND SIMULATION

The greatest emphasis in the development of new analysis capabilities for T&E has been placed in the area of Modeling and Simulation (M&S) where an increased emphasis on a model-test-model paradigm has become prevalent [75]. In the context of T&E, *modeling*, defined as the formalized representation of system characteristics, are digital system models representing varying levels of system performance. *Simulation*, defined as a time stepped execution of models used to predict system performance, is carried out using digital models, hardware-in-the-loop test facilities, and open-air ranges [4]. However, in the vernacular, the term M&S, or simply modeling, is frequently used, when the term simulation would be more correct.

Digital models used in T&E are distinguished based upon the level of detail contained in the model, the type of input/output data used, and the number of players included in the model as shown in Table A-7 [76].

Table A-7 Levels of Modeling

Model Category	Process Modeled	Players Modeled
Campaign	Multiple Battles	Many vs. Many
Battle	Multiple Missions	Many vs. Many
Mission	Mission Engagement	One vs. Many
Engagement	System Engagements	One vs. One
System	Component Functions	One Stand Alone System
Engineering & Component	Engineering Performance	System Components

This modeling hierarchy appears to allow the flow of information from engineering-level to campaign-level, by using the output of the lower-level models as the input to the next higher-level model until the answers at the campaign level are determined based upon component performance. Several issues prevent this from being the solution to decision-making dilemma faced by OT&E decision-makers. First, currently each of the models in this hierarchy are separate, developed by different organizations for different purposes; therefore, there was no thought given to an architecture that would tie these models together until long after the models were already developed. Second, in order for the models at the higher level of the spectrum to run in a reasonable amount of time, many simplifications were made in their development. These simplifications preclude them from being used as detailed analysis tools, rather, they are tools for campaign planning once system performance has been quantified. Finally, the issue of verification, validation, and accreditation (VV&A) of these models is one which is just now beginning to receive attention. No systematic mechanisms or databases are available to allow the analyst to determine a model's applicability to the task at hand. Therefore, although the skeleton of a low-to-high level analysis hierarchy exists in the digital modeling arena, there are many issues keeping it from being a satisfactory solution.

Various methods of simulation have been used in the analysis of system performance for a number of years. Hardware-in-the-loop facilities allow designers to understand how their equipment integrates with other system components early in the design process. Installed-system test facilities, such as anechoic chambers or electromagnetic pulse facilities, allow the analysis of system capabilities in environments that the system is likely to see during operation. Open-air test facilities are used to re-create scenarios that replicate stressing conditions of system operations.

A description of how M&S has been used for the planning and analysis of various Air Force OT&E programs follows. Shown are four diverse examples of recent M&S activities within the AF OT&E community. They show how (1) an engagement-level model was used for analysis that could not be carried out through testing, (2) engagement-level models were used to extrapolate existing test data to untestable conditions, (3) mission-level models will be used to evaluate mission impact of system performance, and (4) hardware-in-the-loop facilities and mission-level models have been used together for test planning. The reader will note that the emphasis of M&S in OT&E is concentrated at the engagement and mission level; however, other than that commonality, there is no other recurring theme in the following discussion. The current state of M&S activities within the OT&E community can best be described as haphazard and disjoint -- there is no "standard" use of M&S within the community. Frequently, it is found that M&S is carried out simply to "fill a square" that the analyst believes the decision-maker wants to see filled rather than to address valid analysis objectives. Additionally, there are no "standard" models or VV&A activities that have been accomplished to ensure the model being used is adequate for the task. Therefore, much of the OT&E community's time is spent in verifying that a model will fulfill analysis needs, rather than spending time on actual analysis activities using the model.

An example of how an engagement level model was used to perform an analysis that could not be carried out through testing is seen in the Short Range Attack Missile (SRAM) II program. This M&S effort used the Enhanced Surface to Air Missile Simulation (ESAMS), an engagement-level model, to evaluate the missile's survivability against enemy surface to air missiles [77]. The SRAM II was an improved version of the original SRAM system, offering stealth capability to allow the missile to penetrate enemy air defenses. The stealth characteristics could not be field tested, instead they were to be evaluated through simulation. ESAMS simulated the engagement between the SRAM and the Surface-to-Air Missile (SAM), determining if a kill occurred based upon the engagement geometry, threat kill potential, and various other factors modeled within ESAMS. A grid of SAM locations was modeled around a simulated target. Using ESAMS, the SRAM and SRAM II were flown over the grid to attack the target. Each SAM location that engaged and killed the SRAM or SRAM II was added to the footprint for that scenario. At the conclusion of the model runs, the size of the footprints were compared -- a smaller footprint indicated a more survivable missile. Initial model runs had been accomplished in an effort to calibrate the ESAMS model when the SRAM II program was terminated by President Bush's 1991 Strategic Drawdown Initiative.

The B-1B Defensive Avionics System (an Electronic Countermeasures (ECM) system designed to protect the B-1B from attacking SAMs) provides an example of how engagement level models are used to extrapolate field testing data to conditions that can not be tested. The testing program's goal was to evaluate the Defensive Avionics System's capabilities and assess its contribution to B-1B survivability [78]. The measure used to determine the ECM effectiveness was missile or projectile miss distance from the B-1B. The test data were collected using a "golden bird" aircraft which played the role of the missile in the engagements with the B-1B. The golden bird aircraft carried an instrumentation pod to collect data indicating relative location of the target aircraft to the golden bird aircraft and a captive carry missile system provided guidance for the aircraft by indicating where the missile seeker was "looking" relative to the target aircraft. The information that could be derived from flight testing was limited due to restrictions on safety of flight considerations, limited test environment terrain, and unavailability of threat seekers; therefore, the M&S effort was used to extrapolate results to other launch conditions, terrains, and threat systems. Flight test data were collected and provided to the model developer for a comparison of model and field testing results to calibrate the models to replicate the field testing environment, and to gain confidence in the model for extrapolation activities. The data comparison (which represented the bulk of the effort) highlighted capabilities and limitations of the engagement-level models for an application of this type. Once the analysts were aware of the models' capabilities, the models were used to evaluate the system performance in the untestable conditions.

The OT&E concept for the B-1B Electronic Countermeasures Upgrade program is currently being developed at the Air Force Operational Test and Evaluation Center (AFOTEC). The test concept calls for the use of simulation for two purposes -- to assist in test criteria development and to extrapolate field testing results to mission level performance [79]. The Cost and Operational Effectiveness Analysis (COEA), a study accomplished to evaluate cost and mission tradeoffs, is currently being accomplished by the Institute for Defense Analysis (IDA). The COEA will analyze the missions that the B-1B, equipped with the new ECM system, is expected to perform, and make an assessment of the impact on mission accomplishment with varying performance values. The analysis will define critical measures of effectiveness that impact the mission accomplishment of the new system. These critical measures of effectiveness will form the basis for the test evaluation criteria. To develop test scenarios, AFOTEC planners will begin with the COEA-modeled scenarios, which represent many simultaneous B-1B missions, and separate each individual mission package from the overall scenario. Each mission flight profile will be compared to the threat asset layout available on the test range, and several test scenarios will be chosen which come closest to the model scenarios. The COEA model will be run using the planned test scenario (i.e., deleting threat systems from the modeled scenario that are not available on the test range) with expected system performance to predict mission outcome. This will provide final system performance values for the test range scenarios that must be met in order for the system to meet mission performance requirements. At the conclusion of testing, the actual measured system performance will be used in the COEA model to predict mission-level performance.

The F-22 Advanced Tactical Fighter program provides an example of simulation using hardware- in-the-loop (HWIL) facilities in conjunction with mission-level digital models to evaluate system performance and assist in the test planning [80]. The Office of the Secretary of Defense mandated a requirement that the F-22 must be twice as good as the F-15 in open-air testing. However, no specifics of what measures must be taken, or what "twice as good" meant were given. To determine what testing should be planned, AFOTEC used a combination of HWIL and digital models to evaluate candidate scenarios for use in the comparison test. Two mission-level models were used to evaluate the scenarios and highlight system sensitivities that should be examined. The Lockheed Full Mission Simulator, consisting of a two-dome system with an F-22 cockpit, a modified F-15 glass cockpit, Manned Interactive Control Stations (MICS) for adversary pilots, Ground Control Intercept (GCI) capability, and Integrated Air Defense System (IADS) capability, was used to evaluate the man-in-the-loop impacts of the candidate scenarios. The simulation was accomplished for a number of reasons; to evaluate: 1) man-in-the-loop response to scenarios developed using digital models, 2) F-22 performance relative to the F-15, 3) potential test measures to be used during OT&E, and 4) range requirements and open-air testability. This effort provided a number of valuable lessons in the use of M&S in OT&E and highlighted many F-22 system performance and M&S issues that will be used to improve and streamline future testing and analysis efforts.

These brief examples were meant to point to the variety of ways M&S has been used in the planning and analysis of recent OT&E programs. The National Research Council states "It seems clear that few if any of the current collection of simulations were designed for use in developmental or operational testing [68]." The Council raises the concerns that (1) rigorous validation of models and simulations for operational testing is infrequent, and external validation is at times used to overfit a model to field experience and (2) there is little evidence seen for use of statistical methods to help interpret results from simulations. Thus, although the M&S arena seems to offer a method for dealing with many of the inadequacies of current analytical methods, it too falls short in many respects.

APPENDIX B

FUZZY SET AND FUZZY LOGIC

This Appendix describes the basics of fuzzy set and fuzzy logic theory. In it, how fuzzy sets and crisp sets differ is discussed; the concepts of fuzzy membership functions and fuzzy set operations are defined; linguistic variables and their modification through hedges, connection and negation operators are described; fuzzy inference mechanisms and defuzzification schemes are described; fuzzy logic's use in control applications are briefly highlighted.

B.1 SET THEORY: CRISP AND FUZZY

As one progresses through a typical secondary school's mathematical education, every school year from Junior High School forward begins with it a study of set theory. In those early math classes the concept membership of a set is illustrated. A member belongs to a set. Membership is designated by the symbol, $\in$, while $\notin$ designates "not a member of." In standard sets, membership is easy to define -- an element either is, or is not, a member of the set. As an example, in the set, $S = \{3, 5, 7, 9\}$, $3 \in S$ and $4 \notin S$. A fuzzy set does not draw such a hard distinction between membership and non-membership, the boundaries between membership and non-membership are gradual and the concept of partial membership within a set is permitted.

B.2 MEMBERSHIP FUNCTIONS

An element in a fuzzy set is characterized by it membership function value -- the degree to which the set label characterizes the element. The membership function value, $\mu_A(x)$, describes the degree of membership for the element, x, within the fuzzy set, A. The values of a membership function range on the interval [0,1]; the closer the value is to unity, the more the attribute associated with the fuzzy set describes the element. More formally [81],

> **Definition:** A ***fuzzy set*, A,** on the given universe X is that, for any $x \in X$, there is a corresponding real number $\mu_A(x) \in [0,1]$ to x, where $\mu_A(x)$ is called the grade of membership of x belonging to A.

There is a mapping,

$$\mu_A : U \rightarrow [0,\ 1], \quad \mathrm{u} \mapsto \mu_A(u) \qquad \textbf{(B-1)}$$

which represents the ***membership function*** of the fuzzy set A.

Membership functions can take on a variety of regular shapes including linear, triangular, trapezoidal, Gaussian, sigmoidal, or PI-shaped curves. In addition, membership functions may take on irregular shapes based on the data used to generate them, such as a bimodal or other "non-normal" distributions. Example membership function curves, the equations which generated them, and sample applications are discussed below.

B.2.1 SIGMOIDAL CURVE

A curve that makes a smooth transition from no membership to full membership with an inflection point at the 50% membership value is the s-curve, sigmoid, or logistic curve. It is defined by three parameters: its zero membership value (α), its complete membership value (γ), and its inflection point (β). The functional representation of the s-curve [82] is shown in Equation (B-2).

$$S(x;\alpha,\beta,\gamma)=\begin{cases} 0 & \text{for } x \le \alpha \\ 2(\frac{x-\alpha}{\gamma-\alpha})^2 & \text{for } \alpha \le x \le \beta \\ 1-2(\frac{x-\alpha}{\gamma-\alpha})^2 & \text{for } \beta \le x \le \gamma \\ 1 & \text{for } x \ge \gamma \end{cases} \quad \textbf{(B-2)}$$

The s-curve resembles a continuous Cumulative Distribution Function (CDF), the probability that a given point X is less than or equal to a point y in the underlying domain.

$$F(y) = \text{Prob}\ (X \le y) \quad \textbf{(B-3)}$$

The CDF can be easily built as a cumulative histogram of data points from the collected test data; thus, this membership function can be built directly from collected test data. This type of curve is used to represent a variety of dynamics that can be approximated as a continuous random variable, such as:

- the speed of an aircraft
- the mean time between failure of a component
- the inter-arrival time of messages in a communication system.

B.2.2 THE PI CURVE

A PI-curve provides a smooth transition in both directions from a central value, where the membership value is unity, to the two points in the domain where the membership values are zero. The symmetric PI-curve is centered on the value from the domain with membership value of unity (γ) with a single parameter that indicates the width of the curve's base (β), and is constructed using two sigmoidal curves, as

$$\prod(x;\beta,\gamma)=\begin{cases} S(x;\gamma-\beta,\gamma-\beta/2,\gamma) & \text{for } x\leq\gamma \\ 1-S(x;\gamma,\gamma+\beta/2,\gamma+\beta) & \text{for } x\leq\gamma \end{cases} \qquad \textbf{(B-4)}$$

Although the shape and characteristics of the PI-curve are quite similar to the Gaussian curve, there is one important distinction which makes the PI-curve a better choice for a membership function. The PI-curve's membership value becomes zero at a discrete and specified point, it is not asymptotic as is the Gaussian curve. This feature provides a closed domain of values for each membership function, a desirable characteristic.

Although the PI-curve does not exactly replicate the Gaussian, it can be used to approximate system characteristics that can be approximated by the Gaussian curve, and thus, be built directly from gathered test data. Braae and Rutherford [83] suggested the use of the triangular membership function when sensor measurements are contaminated with noise, using the vertex as the mean and the base as the standard deviation. In OT&E data, the Central Limit Theory dictates that most of the performance can be approximated by a Gaussian or Normal Probability Distribution Function (PDF), therefore, it might be even more applicable to use Braae and Rutherford's idea of using the vertex as the mean and the base as the standard deviation and apply that standard to the selection of β and γ in the PI-curve membership function.

B.2.3 OTHER MEMBERSHIP FUNCTIONS

The s-curve and the PI-curve are appealing for their similarity to standard statistical distributions, with which the decision-makers are most familiar. However, there are other membership functions that are more commonly used, and which will be used, at least initially, in this work for their ease of mathematical manipulation. These membership functions are described briefly below.

B.2.3.1 LINEAR MEMBERSHIP FUNCTIONS

The linear membership function is probably the easiest membership function, and can be used when the range of the domain is known, but when the characteristics of the underlying distribution are not well understood. As the name suggests, the membership function goes from a point of zero membership to a point of full membership in a straight line fashion. The linear function can be either increasing (positive slope) or decreasing (negative slope) as it ranges from the beginning point to the ending point in the domain.

B.2.3.2 TRIANGULAR MEMBERSHIP FUNCTIONS

The triangular membership function, popular in fuzzy control applications, consists of a triangle with the vertex at the point where membership value equals unity and the base spanning the domain of the set. The triangular membership function provides an easy functional form and a relatively smooth transition from the various values of the membership function.

B.2.3.3 Trapezoidal Membership Functions

Another popular membership function for fuzzy control and model building applications is the trapezoidal function. Its shape, as the name suggests, is a trapezoid with the sides going from the points of zero membership function value to the full membership function value. The region where the membership value equals unity is expanded in the trapezoidal function from that provided in the triangular function.

B.2.3.4 Irregular-Shaped Functions

Finally, underlying data distributions or physical significance may drive the creation of membership functions exhibiting irregular shapes. For example, a membership function of high driving risk would be bimodal with one peak in the teenage years when daring and bravado dictate driving style and another peak in the octogenarian years as poor sight and slow reaction times take over as the dominant style.

B.3 FUZZY SET OPERATIONS

Basic crisp set operations are limited to three operations: intersection, union, and complement. Fuzzy sets complementary operations were originally introduced by Lofti Zadeh. As other researchers have applied fuzzy theory, slight modifications have been made, first using algebraic manipulations of the original functions, then using other transformations known as compensatory operators. In addition to the basic operations of intersection, union, complement and exclusive-or, because of the unique nature of the membership function of a fuzzy set, other operations can be performed on fuzzy sets which have no meaning in the crisp set world. These operations include concentration, dilation, contrast intensification, and α-level set creation. This section describes the basic and unique operations of fuzzy set theory.

B.3.1 ZADEH BASIC FUZZY SET OPERATIONS

The ***union*** [84] of two sets A and B, designated $\{A \cup B\}$ in standard set notation and $\{A \vee B\}$ in fuzzy set notation, results in a set C which contains all the members that were in either set. It is analogous to the **OR** operation in Boolean logic. The union of two fuzzy sets, A and B with membership functions $\mu_A(x)$ and $\mu_B(x)$, is a fuzzy set C with membership function:

$$\mu_C(x) = Max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \tag{B-5}$$

The ***intersection*** [84] of two sets A and B, designated $\{A \cap B\}$ in standard set notation and $\{\mathrm{A} \wedge \mathrm{B}\}$ in fuzzy set notation, results in a set C containing only the members that were in both sets; analogous to the **AND** operation in Boolean logic. The intersection of the two fuzzy sets, A and B, creates the fuzzy set C with membership function:

$$\mu_C(x) = Min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \tag{B-6}$$

The ***complement*** [84] of a fuzzy set A with membership function $\mu_A(x)$ is given by the set $\sim A$

$$\mu_{\sim A}(x) = 1 - \mu_A(x) \tag{B-7}$$

An interesting distinction between fuzzy and crisp set theory relates to the complement operation. In crisp set theory, the *Law of Noncontradiction* states that the intersection of a set and its complement is the empty set (i.e., $A \cap \overline{A} \equiv \varnothing$). However, in fuzzy theory an element can have a degree of membership in both a set and its complement due to the possibility of overlapping regions of the membership functions.

The *Law of the Excluded Middle*, stating that the union of a set with its complement results in the universal set of the underlying domain (i.e., $A \cup \overline{A} \equiv U$), is also violated by fuzzy sets. Alternative definitions of the complement operator are needed if the maintenance of these laws is important.

B.3.2 ALGEBRAIC FUZZY SET OPERATIONS

Fuzzy set researchers have developed alternative definitions for fuzzy set operations, to those proposed by Zadeh. These alternatives fall into two categories -- general algebraic operations and functional compensatory operations. The discussion that follows introduces the alternative definitions and describes their distinction from, and advantages over, the standard operations.

B.3.2.1 THE MEAN OPERATORS

The fuzzy set, *C*'s, membership function resulting from the ***fuzzy mean intersection*** [82] of fuzzy sets *A* and *B* is

$$\mu_C(x) = \frac{\mu_A(x) + \mu_B(x)}{2} \qquad \textbf{(B-8)}$$

This definition can be extended to the intersection of any number of fuzzy sets, simply by calculating the mean of their respective membership functions. This operation can be used instead of the standard intersection operator when it is desired that extreme values not have an undue influence on the outcome of the operation.

In the ***fuzzy mean union*** operation, a beta distribution is used to determine the most likely outcome, similar to the best estimate calculation used in PERT activity time estimates. In the PERT estimate the best probability date is given by

$$T_e = \frac{a+4m+b}{6}$$

By setting $a = b$, the syntax for the ***fuzzy mean union*** [82] operator results, as

$$\mu_C(x) = \frac{2 * \min(\mu_A(x), \mu_B(x)) + 4 * \max(\mu_A(x), \mu_B(x))}{6} \qquad \textbf{(B-9)}$$

B.3.2.2 THE PRODUCT OPERATORS

The product operators, as their name implies, involve the multiplication of membership functions. These operations will give a different value for each different pair operated upon, which is not the case for the strict minimum and maximum operations. This allows the operation to exhibit a characteristic of sensitivity or responsiveness to changes in one of the membership functions. Having this characteristic allows a more realistic modeling of time-varying processes.

The ***product intersection*** [82] is given by:

$$\mu_C(x) = \mu_A(x) * \mu_B(x) \qquad \textbf{(B-10)}$$

The ***product union*** [82] is given by:

$$\mu_C(x) = [\mu_A(x) + \mu_B(x)] - [\mu_A(x) * \mu_B(x)] \qquad \textbf{(B-11)}$$

B.3.2.3 THE BOUNDED SUM OPERATORS

If the objective is to create a filtering mechanism with the fuzzy set operation, the bounded sum operators can achieve that objective. The ***bounded sum intersection*** and ***bounded sum union*** [82] operators are given respectively as:

$$\mu_C(x) = \max[0, \mu_A(x) + \mu_B(x) - 1] \qquad \textbf{(B-12)}$$

$$\mu_C(x) = \min[1, \mu_A(x) + \mu_B(x)] \qquad \textbf{(B-13)}$$

The ***bounded sum intersection*** operator is highly selective: unless the combined membership functions of the two components exceeds unity, the resulting membership function value equals zero. This function is similar to a neural network activation function that does not have enough of a value to activate the neuron. The operator acts as a hurdle in the intersection space. This type of operation may prove appropriate for the analysis of systems where performance below a certain threshold is absolutely unacceptable. This situation arises in the OT&E of such systems as aircraft emergency escape systems or nuclear system fail-safe mechanisms where a certain minimum level of performance must be guaranteed.

On the other hand, the ***bounded sum union*** operator is overly accommodating. If the combination of the two constituent membership functions exceeds unity, the operator returns a value of unity -- it establishes a low membership hurdle within the union space. This operator may be appropriate for the analysis of systems where the risk of making the wrong decision is small. This situation may occur in a number of instances in OT&E analysis. For example, the testing of a system which is an upgrade of a current system based upon proven technology, the testing of a system where cost is not a major factor, or the testing of a new system that is fulfilling a need that is currently going unfulfilled -- the case where any kind of performance is better than what is currently available. All of these instances illustrate cases where the bounded sum union would be an attractive choice for combining the membership functions.

B.3.3 FUNCTIONAL COMPENSATORY OPERATORS

Table B-2 provides examples of ***functional compensatory operators*** created by fuzzy logic researchers [85]. All of these operators depend on a parameter, called here *k*, which can be varied to change the characteristics of the operator. In most cases, by changing the value of the parameter, the operator can be softened or hardened with respect to the standard minimum and maximum operations.

Table B-2 Functional Compensatory Union and Intersection Operators

Researcher	**Intersection**	**Union**
Yager	$1-\min[1,((1-a)^k+(1-b)^k)^{\frac{1}{k}}]$	$\min[1,(a^k+b^k)^{\frac{1}{k}}]$
Dubois & Prade	$\dfrac{ab}{\max[a,b,k]}$	$\dfrac{a+b-ab-\min[a,b,1-k]}{\max[1-a,1-b,k]}$
Schweizer & Sklar	$\max[0,a^{-k}+b^{-k}-1]^{-\frac{1}{k}}$	$1-\max[0,((1-a)^{-k}+(1-b)^{-k}-1)]^{-\frac{1}{k}}$

These parameterized functions, and those similar to them proposed by other authors, have been used in applications where the standard Zadeh or algebraic operators were not suitable. As the work here progresses a determination will be made as to the most appropriate operators for the task. The principle of parsimony will be applied -- first the simple Zadeh operators will be used to see if their performance is adequate, then other more difficult operators will be subsequently tried until acceptable performance is reached.

B.4 LINGUISTIC VARIABLES

Linguistic variables are common language terms used to describe fuzzy sets [86]. Linguistic variables can include such terms as *small, medium, large, short, tall, close, far*. Each of the membership functions described above would be assigned a linguistic variable to aid in the humanistic reasoning perspective of working with fuzzy sets. Using linguistic variables allows, for example, control rules to be written in common language phrases that an operator could provide, such as "if the slurry is yellowish in color, add a little more of chemical xyz." In this brief example, *yellowish* and *a little more* represent fuzzy linguistic variables for the control variables *color* and *amount.*

B.4.1 LINGUISTIC VARIABLE HEDGES

A ***hedge*** modifies a linguistic variable through a change in the shape of the membership function values using operations such as ***contrast intensification, concentration, dilution***, and other operations that transform the membership function values. Table B-2 [82] gives an overview of hedges that are used and their approximate meaning, while Section B.4.2 gives definitions for the operations associated with some of the hedges listed here.

Table B-2 Linguistic Hedges and Their Meanings

HEDGE	MEANING
about, around, near, roughly	Approximate a scalar
above, more than	Restrict a fuzzy region
almost, definitely, positively	Contrast intensification
below, less than	Restrict a fuzzy region
vicinity of	Approximate broadly
generally, usually	Contrast diffusion
neighboring, close to	Approximate narrowly
not	Negation or complement
quite, rather, somewhat	Dilute a fuzzy region
very, extremely	Intensify a fuzzy region

B.4.2 CONCENTRATION, DILATION, AND CONTRAST INTENSIFICATION

Operations that are unique to fuzzy sets (i.e., have no meaning in the crisp set world) include concentration, dilation and contrast intensification [10]. These operations serve to change the shape of the fuzzy membership functions, and are the basis for ***hedges*** of fuzzy linguistic variables.

The operation of ***concentration*** is defined as

$$CON(A) \equiv A^2 \tag{B-14}$$

This operation has the effect of decreasing the membership function values for the members of the fuzzy set. However, the decrease is not uniform across the set. The concentration effects the members with higher membership function values less than those with low membership function values. For example, if the discrete fuzzy set A is defined, using the union of fuzzy singleton notation, as

$$A = 0.4/1 + 0.6/2 + 0.8/3 + 1.0/4 + 0.8/5 + 0.6/6 + 0.4/7 + 0.2/8$$

then the fuzzy set $CON(A)$ or A^2 would be equal to

$$A^2 = 0.16/1 + 0.36/2 + 0.64/3 + 1.0/4 + 0.64/5 + 0.36/6 + 0.16/7 + 0.04/8$$

The concentration operator with the exponential value of 2, is commonly associated with the hedge *very*. So, for the example shown above, the fuzzy set $CON(A)$ or A^2 would be associated with the hedged set *very A*.

The operation of ***dilation*** is defined as

$$DIL(A) \equiv A^{0.5} \tag{B-15}$$

The effect of the dilation operator is the opposite of the concentration operator; it dilutes the value of the membership function more for those elements with large membership functions and less for those with small membership functions.

Although no strict hedge/dilation relationship is given for an exponential value of 0.5, hedges, *somewhat, quite,* or *rather* could be associated with the *DIL* operation. Milder degrees of concentration and dilation than those given by the strict CON and DIL operations are associated with the hedges *plus* and *minus*, defined in [10], as

$$plus\ x \equiv x^{1.25} \tag{B-16}$$

$$minus\ x \equiv x^{0.75} \tag{B-17}$$

Finally, the operation of ***contrast intensification*** increases the values of the membership function for those members with a value above 0.5 while simultaneously decreasing the values for those elements with a value below 0.5. This operation can be thought of one which reduces the fuzziness of the set it is applied to and can be associated with the hedges *almost, definitely,* or *positively.* The definition of the operation is

$$INT(A) \equiv \begin{cases} 2A^2 & \text{for } 0 \le \mu_A(x) \le 0.5 \\ 2(\sim A)^2 & \text{for } 0.5 \le \mu_A(x) \le 1.0 \end{cases} \tag{B-18}$$

B.5 FUZZY INFERENCE

Inference is the process through which conclusions are drawn, or decisions are made, based upon the information at hand. There are many methods of drawing inferences. Here, two are discussed: ***compositional rules of inference***, based on combining fuzzy sets using mathematical means, and ***syllogistic reasoning***, which bases conclusions on the satisfaction of premise statements.

B.5.1 COMPOSITIONAL RULES OF INFERENCE

In the literature on fuzzy logic control applications, inferences are made using ***compositional rules of inference***. Once a rule base has been established (usually through consultation with a human operator who provides heuristic control rules in linguistic form) the input is used to determine which rules within the rule base 'fire'. Once the applicable rules are identified, a compositional method combines the rules' outputs into a single fuzzy set output, which is later defuzzified to provide the crisp output value. There are many different compositional rules of inference; however, the most popular is the min-max compositional rule, due to Mamdani [87], shown below to relate $\mu_B(y)$ to $\mu_R(x,y)$ and $\mu_A(x)$

$$\mu_B(y) = \max_x \min(\mu_A(x), \mu_R(x,y)) \qquad \textbf{(B-19)}$$

This compositional method requires that the relationship between fuzzy sets A and B be established, which can be done through a variety of means. Mamdani suggests either a minimum (B-20) or a bounded sum operation (B-21) be used to construct this relation as

$$\mu_R(x,y) = \min[\mu_A(x), \mu_B(y)] \qquad \textbf{(B-20)}$$

or

$$\mu_R(x,y) = \min[1, (1 - \mu_A(x) + \mu_B(y))] \qquad \textbf{(B-21)}$$

Once the relationship is developed it can be used to determine outputs for any desired input values based upon the same relational condition. Consider the example of two fuzzy sets A of X and B of Y, both discrete and finite, with vectors of membership function values given as

$$A = [\ 1, 0.8, 0.6, 0.2, 0, 0.3\ ]$$

$$B = [\ 0.1, 0.2, 0.3, 0.4, 0.5\ ]$$

The relationship between A and B can be formed using the minimum operation as

$$R_{A \to B} = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.5 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.5 \\ 0.1 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0.3 & 0.3 & 0.3 \end{bmatrix}$$

Once this relationship is established, if an input vector, say A' which is "similar to" A has the same relative relationship to B, $R_{A \to B}$ can be used to draw the inference on what the resulting B' should be, as

$$B' = A' \circ R_{A \to B} \qquad \textbf{(B-22)}$$

So, in the example, if

$$A' = [\ 0.9, 0.8, 0.5, 0.2, 0\ 0.2\ ]$$

then

$$B' = [\ 0.1, 0.2, 0.3, 0.4, 0.5\ ]$$

Which is formed, using the first element as an illustration, as

$$\max [\, 0.9 \wedge 0.1,\, 0.8 \wedge 0.1,\, 0.5 \wedge 0.1,\, 0.2 \wedge 0.1,\, 0 \wedge 0,\, 0.1 \wedge 0.2 \,] = 0.1$$

Using this method of inference, if input/output information exists, a relationship can be developed which is subsequently used to draw inferences for similar circumstances. This form of inference is used when test data can be gathered to help define these relations. If the relationship cannot be defined, other methods of inference such as syllogistic reasoning, which forms the basis for inference in most current expert system applications, must be used.

B.5.2 SYLLOGISTIC REASONING

Syllogistic reasoning is defined as a form of deductive reasoning consisting of a major premise, a minor premise, and a conclusion. For example,

icy roads are slippery (major premise)
<u>slippery roads are dangerous (minor premise)</u>
icy roads are dangerous (conclusion)

Fuzzy syllogisms are of the general form [88]

$$\begin{array}{c} p \equiv Q_1 A\text{'s are } B\text{'s} \\ \underline{q \equiv Q_2 C\text{'s are } D\text{'s}} \\ r \equiv QE\text{'s are } F\text{'s} \end{array} \qquad \textbf{(B-23)}$$

where p is a fuzzy proposition containing a fuzzy quantifier Q_1 and fuzzy predicates A and B; the second premise, q, is a fuzzy proposition containing a fuzzy quantifier Q_2 and fuzzy predicates C and D; and the conclusion r is a fuzzy proposition containing a fuzzy quantifier Q and fuzzy predicates E and F.

The most common and important syllogisms are shown below, where $\wedge$ denotes conjunction or intersection (AND) and $\vee$ denotes disjunction or union (OR). These are given without any restrictions on the relationships between $A, B, C, D, E,$ and F. If assumptions are made on those relationships, then statements can be made as to the relationships between Q, Q_2 and Q_1.

Intersection/Product Syllogism

$$\text{if: } C = A \wedge B \quad \text{then: } E = A,\ F = C \wedge D$$
$$\text{and } Q = Q_1 \otimes Q_2 \tag{B-24}$$

where $\otimes$ denotes fuzzy set multiplication.

Chaining Syllogism

$$\text{if: } C = B \quad \text{then: } E = A,\ F = D$$
$$\text{if: } B \subset A \quad \text{then } Q \geq Q_1 \otimes Q_2 \tag{B-25}$$

where $\geq Q_1 \otimes Q_2$ is read as "at least equal to" $Q_1 \otimes Q_2$.

Consequent Conjunction Syllogism

$$\text{if: } A = C = E \quad \text{then: } F = B \wedge D$$
$$\text{and } [\, 0 \vee (\, Q_1 \oplus Q_2 \ominus 1 \,)\,] < Q < [\, Q_1 \wedge Q_2 \,] \tag{B-26}$$

where $\ominus$ denotes fuzzy set subtraction.

Consequent Disjunction Syllogism

$$\text{if: } A = C = E \quad \text{then: } F = B \vee D$$
$$\text{and} \quad [\,0 \vee (\,Q_1 \oplus Q_2 \ominus 1\,)\,] < Q < [\,Q_1 \wedge Q_2\,] \tag{B-27}$$

Antecedent Conjunction Syllogism

$$\text{if: } B = D = F \quad \text{then: } E = A \wedge C \tag{B-28}$$

For the antecedent conjunction case, an assumption is made that each proposition is independent. This can be phrased in terms of the sigma-count, a means of accounting for cardinality in fuzzy sets. The sigma-count is defined as the real number, rounded to the next highest number if necessary, that is the sum of all the membership function values in the fuzzy set. For the fuzzy set *A*

$$\sum count(A) = \sum_i \mu_i \tag{B-29}$$

The relative sigma count between two fuzzy sets can be interpreted as the number of elements in one set that are also in the other. Between the two sets *A* and *B*, the relative sigma count, denoted *Σcount(B/A),* is the proportion of elements of *B* in *A*

$$\sum count(B/A) = \frac{\sum count(B \cap A)}{\sum count(A)} = \frac{\sum_i \mu_B(x_i) \wedge \mu_A(x_i)}{\sum_i \mu_A(x_i)} \tag{B-30}$$

Then, the independence assumption may then be written as

$$\sum count(\frac{A \cap B}{C}) = \sum count(\frac{A}{C}) \sum count(\frac{B}{C}) \qquad \textbf{(B-31)}$$

Alternatively, in terms of a count referred to as the *ρsigma-count*, which is a ratio of the members of a set to the non-members of the set, defined as

$$\rho \sum sigma(B) = \frac{\sum count(B)}{\sum count(\overline{B})} \qquad \textbf{(B-32)}$$

Using this measure, the antecedent conjunction syllogism quantifier, Q, becomes

$$Q = R_1 \otimes R_2 \otimes R_3 \qquad \textbf{(B-33)}$$

where the ratios, R_i's, are defined as

$$R_1 = \rho \sum count(\frac{C}{A}) = \frac{\sum count(C / A)}{\sum count(\overline{C} / A)} = \text{ ratio of C's to non-C's among A's}$$

$$R_2 = \rho \sum count(\frac{C}{B}) = \frac{\sum count(C / B)}{\sum count(\overline{C} / B)} = \text{ ratio of C's to non-C's among B's}$$

$$R_3 = \rho \sum count(C) = \frac{\sum count(C)}{\sum count(\overline{C})} = \text{ ratio of C's to non-C's}$$

where the syllogism has been redefined using the notation shown below, due to the equalities defined in (B-28).

Q_1A's are C's

Q_2B's are C's

Q(A and B)'s are C's

B.6 DEFUZZIFICATION SCHEMES

The basis of fuzzy logic's appeal is its ability to consider the uncertainties and gradual transitions associated with the membership functions. However, giving a solution that includes an entire region of the domain space, or is a fuzzy set itself, may not be acceptable. For example, when using fuzzy logic for control applications, the controller needs a single, or crisp, value to use as the control input -- the controller hardware would not know how to handle a fuzzy set control input value. Thus, there is a need to defuzzify the results of the fuzzy inference procedure to provide a crisp answer at the end of the process. Defuzzification finds the value that best represents the information contained within the fuzzy set. Various defuzzification methods have been suggested in the literature, several of the most common are discussed below.

The most popular defuzzification scheme is the ***Center of Area Defuzzification Method***. Using this method, the centroid of the fuzzy membership function is calculated, and the domain value associated with that centroid is used as the crisp result. The center of area is calculated as [89]

$$y^{COA} = \frac{\sum_{j=1}^{p} y_j \mu_B(y_j)}{\sum_{j=1}^{p} \mu_B(y_j)} \qquad \textbf{(B-34)}$$

This is a popular method of defuzzification because the fuzzy centroid is unique and considers all the information in the fuzzy set distribution. For the case where the fuzzy sets are continuous, the summations are replaced with integrals, but the basic form of the equation remains the same.

Another common defuzzification scheme is the ***Mean of Maxima Defuzzification Scheme***. This method depends on finding the domain value having maximal membership grades. It is calculated as

$$y^{MOM} = \frac{1}{q} \sum_{j \in J^*} y_j \qquad \textbf{(B-35)}$$

where J^* is the set of elements of the universe with the maximum membership function value and q is the cardinality of J^*.

Finally, the ***Method of Heights Defuzzification Method*** uses the members of the set which attain a level of membership greater than a given α-cut to calculate the crisp value. The α-level set, A_α, is defined as [90]

$$A_\alpha \equiv \{x_i | \mu_A(x_i) \geq \alpha\} \; 0 \leq \alpha \leq 1, \; x_i \in X, \; i = 1,2\ldots,n \qquad \textbf{(B-36)}$$

Then, the *Method of Heights Defuzzification Method* is defined as, the *Center of Area Defuzzification Method* using only those members of the domain that are a member of the α-level set. The crisp value is defined as

$$y^{MOH} = \frac{\sum_{i \in A_\alpha} y_i \mu_A(y_i)}{\sum_{i \in A_\alpha} \mu_A(y_i)} \qquad \textbf{(B-37)}$$

Other defuzzification methods have been suggested [82], including *average of the maximum plateau defuzzification, average of the support region defuzzification, far edge of the support set defuzzification, near edge of the support set defuzzification, and center* *of the maximums defuzzification.* These methods offer slight deviations from the standard methods described above, and may prove applicable for a given application. However, as with the choice of a membership function, the methodology will be examined for adequacy using the standard defuzzification techniques before any of the alternative methods are applied.

B.7 FUZZY CONTROL

The problem being addressed with this research is not a controls problem. However, this section will briefly review the structure of fuzzy control because the current literature is rich with examples of fuzzy control applications and if one steps back from the problem at hand for a moment and compares it to a standard control problem, some striking similarities will be noted. For example, the methodology of defining the indicative variables, fuzzifying those variable, performing an inference procedure, and defuzzifying to get a crisp control action can be translated almost intact to the analysis task.

A standard control system uses a control law based upon differential equations which are a function of previous output and control input values. A fuzzy logic controller uses a control law based upon a knowledge-based system containing rules in IF ...THEN form. This knowledge base is built using linguistic variables from the actions that a human controller might use to control the process. Once the knowledge base is built, it can be used as the basis of the inference process, to control the process using the following steps:

- Find the firing level of each of the rules
- Find the output of each of the rules
- Aggregate the individual rule outputs to obtain the overall output.
- Defuzzify the output to find the crisp control value

A relevant example that ties control and analysis/diagnostic work together is the work done by Boeing Corporation researchers in developing a combined neural network-fuzzy logic system to identify and diagnose performance on jet engines [91]. The diagnostic and control system uses a neural network, trained to provide normal component performance under differing input conditions, to provide the "normal" performance measure. The actual system performance and the neural network performance under the same operating conditions are compared to provide a system error measurement. The system error measurement is run through a delay unit to create a change in error measurement. The error and change in error are provided as inputs to the fuzzy logic unit, which is composed of a knowledge-base outlining control actions required for various combinations of error and change in error states. Each individual fuzzy logic unit provides a control action recommendation for its component as input to an aggregation fuzzy logic unit, which serves as an aggregation and defuzzification mechanism to take the individual control actions and the previous overall control action and provides a single, crisp control action for the entire engine system as output.

APPENDIX C

FUZZY COGNITIVE MAPS

C.1 CONCEPTS AND RELATIONSHIPS

A cognitive map is a representation of relationships between elements in an environment [92]. Each node in the map represents a concept or element, and the relationships between the nodes are defined by directed arcs connecting the nodes. The relationships, either excitatory, inhibitory, or neutral; are represented by the edge values connecting associated nodes in the map. An edge value, denoted as e_{ij}, relating concept C_i and C_j, lies in the fuzzy [-1, 1] interval and takes on values in [-1, 0) indicating a negative causality (i.e., an increase in C_i will result in a decrease in C_j), values in (0, 1] indicating a positive causality (i.e., an increase in C_i results in an increase in C_j), and a value of zero if there is no causal relationship between the concepts. The original cognitive maps suggested by Axelrod only allowed edge values of {-1, 0, +1} to describe the relations between concepts; two concepts were related through a positive causality, a negative causality, or had no relation. Kosko's work extended the original theory to allow fuzzy values or fuzzy linguistic terms to describe the relations between concepts.

Kosko defined a fuzzy causal algebra to describe concepts and their causal relationships [93]. He defined a concept C_i as the fuzzy union of some fuzzy quantity set Q_i and its associated dis-quantity set $\sim Q_i$, thought of as the logical fuzzy set complement, as

$$C_i = Q_i \cup \sim Q_i \tag{C-1}$$

Then Kosko defined the causality relationships as

C_i *causes* C_j iff $[(Q_i \subset Q_j)$ AND $(\sim Q_i \subset \sim Q_j)]$

C_i *causally decreases* C_j iff $[(Q_i \subset \sim Q_j)$ AND $(\sim Q_i \subset Q_j)]$

where $\subset$ indicates fuzzy set inclusion. The degree of subsethood of C_i in C_j, or the fuzzy set membership of concept C_i in concept C_j's fuzzy power set, is then used to define the fuzzy edge values, as

$$e_{ij} = (C_i, C_j) = \mu_{F(2^{C_j})}(C_i) \tag{C-2}$$

Although this theoretical framework for defining causality exists, most applications of FCMs, including the one proposed here, depend on gathering expert judgment to define the edge values. The FCM is developed by extracting expert opinion to identify possible key results or influences for the problem at hand [94]. Once the key factors are identified, the expert identifies the causal relationships between the factors, drawing a FCM in diagram form as shown in Figure C-1, which relates vehicle speed, to police patrol frequency and accident rate.

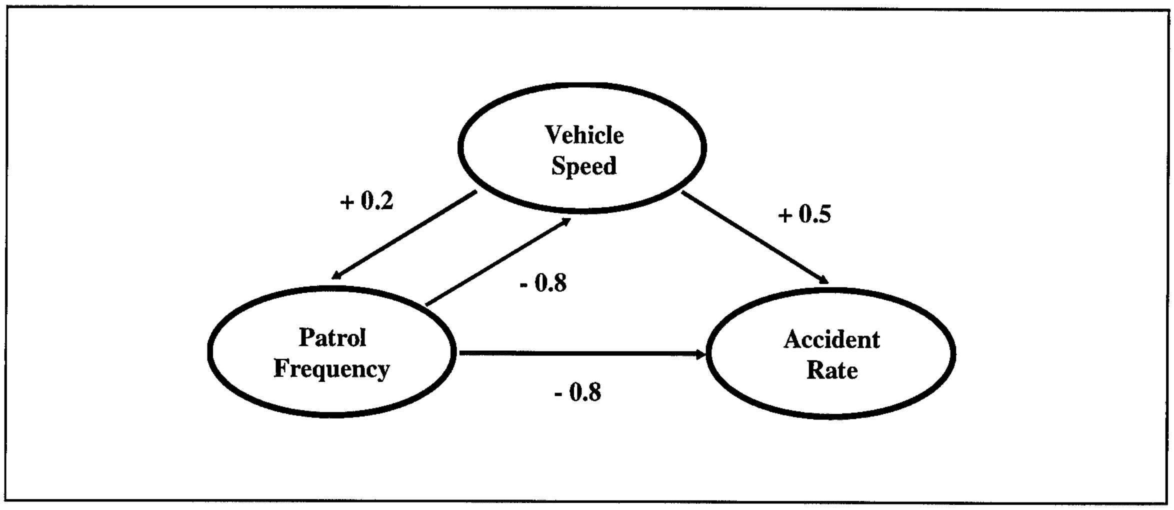

Figure C-1 Simple Fuzzy Cognitive Map

This simple map shows that as the vehicle speed increases, the patrol frequency and accident rate also increase, a positive causality between these factors. It also shows that as the patrol frequency increases, both vehicle speed and accident rate decrease, illustrating an inhibitory relationship. There is no limit to the size of an FCM; as many factors as are applicable to the situation may be included and interrelated. The relationships in an FCM may be developed as a result of simple expert opinion, or as a result of data gathered on the relative effects of the factors. In either case, after the FCM diagram is developed, an adjacency or connection matrix is defined to house the relational values between the concepts.

The adjacency matrix represents the causal edge functions of the FCM. The adjacency matrix for the FCM shown in Figure C-1 is

$$F = \begin{bmatrix} 0 & 0.5 & 0.2 \\ 0 & 0 & 0 \\ -0.8 & -0.8 & 0 \end{bmatrix} \qquad \textbf{(C-3)}$$

Section C.2 discusses how FCMs gathered from various sources can be combined to form a single, global FCM, thus allowing consideration of various experts' opinions and information from other data sources in the decision-making process. Once that concept is illustrated, and a thorough understanding of the source of the cognitive map has been gained, Section C.3 will show how the adjacency matrix is used to draw inferences based upon the information coded in the FCM.

C.2 COMBINING INFORMATION FROM DIVERSE SOURCES

While expert systems built using a tree structure suffer from a severe limitation in their ability to combine information from diverse sources, FCMs readily combine. The FCM structure allows combination of individual FCMs to form a global FCM through an augmentation, and if desired, weighting process. Each individual FCM, contributed by a different expert or data source, can include different conceptual nodes. To combine the FCMs, each individual FCM is augmented to include a row and a column containing all zeroes for all concepts included in other FCMs not included in that FCM. Once all the FCMs have been augmented, they are simply added together, using matrix addition to yield a global FCM. Although none of the current literature discusses a normalization step after the FCMs have been combined, that seems a logical concluding step resulting in a FCM with edge values falling within [-1, 1] exclusively.

The resulting global FCM is potentially a more reliable information source than any of the individual FCMs because it is derived from a variety of sources, making the effect of errors in any one source less important [95]. This assertion assumes that the Law of Large Numbers holds and that the expert opinion is *i.i.d.* (independent and identically distributed). The *i.i.d.* assumption is justified because each FCM represents the opinion of an independent expert or source of data, all focused on the same problem. By combining the information from a large number of sources, the result is a reliable information source which tends to be self-weighting. The self-weighting aspect comes from an assumption that most experts will agree on similar concepts, therefore, the relationships they define will enhance each other as the FCMs are combined. As the global FCM is constructed, if the self-weighting feature does not emphasize the importance of the data collected in an appropriate fashion, a forced weighting scheme can be implemented.

If the information from some sources (experts) carries more credence than other information sources, a credibility weighting scheme can be included when the individual matrices are combined to form the global matrix, as [96]

$$F = \sum_{i=1}^{NE} w_i F_i \qquad \textbf{(C-4)}$$

where i is the index number associated with the number of experts or information sources being considered, w_i is the weighting factor associated with the individual, augmented FCM F_i, and F is the resulting global matrix.

As an alternative to simply summing the individual FCMs to form a global FCM, other methods of fuzzy knowledge combination can be considered. These fuzzy knowledge combination methods can range from the very pessimistic, intersection operator, to the very optimistic, union operator. A number of conditions have been proposed, which aid in choosing an appropriate combination operator. Pelaez [97] discusses a ***knowledge combination function***

$$\varnothing : K^n \rightarrow K \qquad \textbf{(C-5)}$$

to be used when combining individual FCMs into a global FCM. His knowledge combination function addresses the case where the same links are defined by different experts using different fuzzy linguistic labels, and a mechanism is needed to determine what the global linguistic label should be for the disputed link. In the development of his *knowledge combination function*, Pelaez considers a number of conditions which must be satisfied in order for the knowledge combination function to be adequate. Those conditions are discussed briefly below, followed by the definition of the knowledge combination function chosen by Pelaez for his work, which also will be adopted here.

C.2.1 KNOWLEDGE COMBINATION FUNCTION CONDITIONS

First, Pelaez denotes **S** as a set of query stimuli, **K** is a partially ordered set of knowledge responses to those query stimuli. Each knowledge source (i.e., expert) will provide a different mapping from the query to an answer. He calls the set of mappings from query to response the set **X**, such that $X_i : \mathbf{S} \rightarrow \mathbf{K}$. Therefore, his knowledge combination function yields a single piece of information, *k*, from all the knowledge responses of all the knowledge sources.

$$\varnothing((X_1(s_1), X_2(s_2), \ldots, X_n(s_n)) = k \qquad \textbf{(C-6)}$$

The knowledge combination functions depend on the intersection and union operators with respect to a knowledge response vector **X**, and its components **X(s)**. Thus, for notational simplicity, they are denoted as

$$l = \min_i X_i(s) \qquad \textbf{(C-7)}$$

$$m = \min_i X_i(s) \qquad \textbf{(C-8)}$$

The ***boundedness*** condition considers that *at least* the information represented by the intersection of all the experts' opinions and *at most* the information represented by the union of all the experts' opinions, is known. Therefore, the knowledge combination function should fall within this range.

$$l \leq \varnothing \leq m \qquad \textbf{(C-9)}$$

The ***symmetry*** condition requires that permutations of the original response vector will yield the same knowledge combination result. Let **P(n)** be a permutation of the original response vector **X(s)**, and $X_p(s) = ((X_{p_i}(s),\ldots,X_{p_n}(s))$ be a mixture of the knowledge response vector components, then the symmetry condition is defined as

$$\varnothing(X(s)) = \varnothing(X_p(s)) \text{ for all } p \in P(n) \qquad \textbf{(C-10)}$$

The ***knowledge gap***, the difference between what all experts agree upon (information contained in the intersection set) and the total diverse expert opinion (information contained in the union set) is considered in the following conditions. This knowledge gap, is represented by the difference between the minimum and maximum values calculated above, (i.e., *m-l*). This measure can be associated with the measure of dispersion in a statistical distribution, characterized, in a normal distribution by the standard deviation.

The ***conservatism*** condition requires that as the gap in knowledge increases, the decision method should tend toward the conservative. With the most conservative knowledge combination method being the intersection or fuzzy minimum operator, this condition is stated as

$$\varnothing \downarrow l \text{ as } m - l \uparrow 1 \qquad \textbf{(C-11)}$$

If an assumption is made that all that is really known once information sources are combined is the minimum and maximum values, and that the information gap contains the extent of the fuzzy knowledge, then the ***nonparametricism*** condition is stated as

$$\varnothing(X(s)) = \varnothing(l,m) \qquad \textbf{(C-12)}$$

Finally, the ***leniency*** condition reflects the property that as the knowledge gap shrinks, a more lenient approach to knowledge combination can be adopted, such that

$$\varnothing \uparrow m \text{ as } m - l \downarrow 0 \qquad \textbf{(C-13)}$$

Even with the restrictions imposed by conditions (C-9) through (C-13), there is still a large set of knowledge combination techniques that could be adopted. Pelaez [97] suggests selecting a knowledge combination function, $\varnothing$, determining if it meets the conditions described above, (i.e., belongs to the function space of $\boldsymbol{\Phi}$, the admissible knowledge combination operators which satisfy the above conditions) and seeing how it behaves. His selection for $\varnothing$, which will also be adopted for this work is shown below.

$$\varnothing(l,m) = \min(m,\ 1 - m + l) \qquad \textbf{(C-14)}$$

This knowledge combination operator includes the information contained within the knowledge gap with the second term in the minimum operation. That term represents the negation of the knowledge gap. This choice for an operator also strikes a compromise between a strict minimum or maximum-based operation; a good compromise between conservatism and leniency.

C.3 DRAWING INFERENCES USING AN FCM

Drawing inferences based upon the information contained within a global FCM is a simple matter of repeated matrix multiplication and thresholding operations. The result is a "walk" through the states of the FCM until a limit state or cycle is reached; indicating the final stable state(s) of the system resulting from the original input condition. Based upon the limit cycle existence, the FCM has proven to be an ideal tool to answer "what if" questions. This research will extend that use, to show that FCMs can also be used to adjust input membership functions as a result of the influence of the factors included in the FCM. The inference technique used within this research will be a combination of the inference techniques discussed below.

Once the FCM has been constructed, it can be used to draw inferences on what concepts contained in the FCM will result if one of the concepts is activated. For example, an FCM built to illustrate the interrelationships between various aspects of South African politics by Kosko based upon a syndicated article by Walter Williams is shown in Figure C-2 [1].

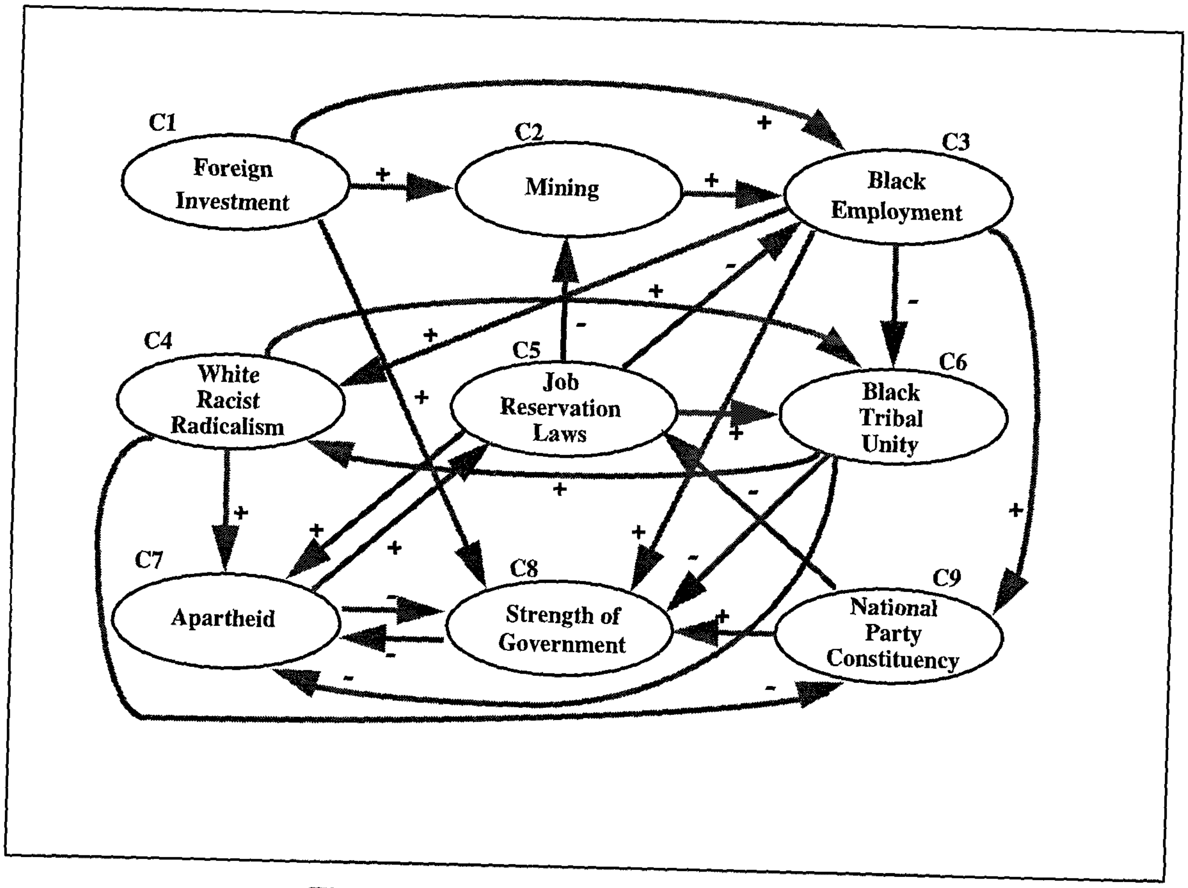

Figure C-2 South African Political FCM

Based upon the interrelationships of factors illustrated in the FCM of Figure C-2, what will be the result if the Foreign Investment Policy is followed? The FCM can be used to answer this question through a series of vector/matrix multiplication and thresholding steps until a limit cycle is reached, indicating the final state and the final inference. First, the adjacency matrix associated with Figure C-2 is defined as

$$F = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad \textbf{(C-15)}$$

Next, the vector with the concept of interest "clamped" to a unity value and the remainder of the elements set to zero is defined.

$$C_1 = [\,1\,0\,0\,0\,0\,0\,0\,0\,0\,] \qquad \textbf{(C-16)}$$

Now, the concept vector and adjacency matrix are multiplied, then the thresholding operation is performed; indicated below by the arrow symbol. The thresholding operation here[14] is defined as a transformation of any positive value to a value of one and any

[14] The thresholding operation illustrated here is that suggested in [93]. However, the thresholding operation performed in other literature consists of clamping values greater than zero to a value of +1, clamping values less than zero to a value of -1, and clamping a value of zero to a value of 0. The result from the differences in the clamping value will, in general, result in more concepts being activated in the final limit state.

negative value to a value of zero and reclamping the original concept element to a value of one, as

$$\begin{aligned} C_1F &= [\,0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\,] \\ &\rightarrow [\,1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\,] = C_2 \end{aligned} \qquad \textbf{(C-17)}$$

Then, the multiplication and thresholding operations are repeated until a limit state or cycle is reached.

$$\begin{aligned} C_2F &= [\,0\ 1\ 2\ 1\ \text{-}1\ \text{-}1\ \text{-}1\ 4\ 2\,] \\ &\rightarrow [\,1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\,] = C_3 \end{aligned} \qquad \textbf{(C-18)}$$

$$\begin{aligned} C_3F &= [\,0\ 1\ 2\ 1\ \text{-}1\ 0\ 0\ 4\ 1\,] \\ &\rightarrow [\,1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\,] = C_3 \end{aligned} \qquad \textbf{(C-19)}$$

In this example, the FCM reaches a limit state at the third iteration[15], giving the final inference to the starting concept. It infers the concepts { C_1, C_2, C_3, C_4, C_8, C_9 } given the policy "what-if" question of {C_1}.

Synchronous FCMs act as Temporal Associative Memories (TAM) [1] where an n-dimensional autoassociative feedback network encodes an ordered sequence of n-dimensional bit vectors as a stable limit cycle in the signal state space $\{0, 1\}^n$. This limit cycle represents the states of the system and is commonly designated as its length, or the number of interceding states before the system repeats itself [98]. For example, an FCM that cycles through the cycle shown in Table C-1 would be a 4-L limit cycle, whereas, the FCM that settles at one point and remains there is a 1-L or fixed point limit cycle.

[15] If the +1/0/-1 clamping had been used in this example the resulting limit state would have been {C1, C2, C3, C5, C6, C7, C8, C9} after five iterations vs. the {C1, C2, C3, C4, C8, C9} result after three iterations of the +1/0 clamping operation shown here.

Table C-1 4-L Limit Cycle Example

Vector	Iteration
0 0 1 0	Input
0 0 0 1	First Output
1 0 0 0	Second Output
0 1 0 0	Third Output
0 0 1 0	Fourth Output
0 0 0 1	Fifth Output

In practice, most FCMs settle into a very short limit cycle, or exhibit fixed point behavior. However, depending on the allowed edge values, the maximum number of limit cycles can theoretically be as large as 2^n for a Bivalent State, 3^n for a Trivalent State, and infinite for a Continuous State FCM.

Another means of drawing inferences from FCMs is based upon Kosko's fuzzy causal algebra, derived originally from Axelrod's idea of ***indirect*** and ***total causal effects*** in cognitive maps [93]. When a path from two concepts within the FCM is examined, the *indirect effect* is the causality C_i imparts on C_j via each possible path between the two concepts, whereas, the *total effect* is the aggregated causal effect across all possible paths between the two concepts. Axelrod suggested using the signs of each leg of the path to determine the indirect and total effects, however, this method resulted in indeterminate conclusions on a regular basis. Therefore, Kosko extended the method by using minimum and maximum operations. The calculation of the indirect effect is a matter of calculating the minimum value along the path (if fuzzy linguistic variables are being used to describe the links, an ordering of the terms must be determined prior to this operation). Then the total effect is a maximum operation over all the indirect effects. The minimum and maximum can be replaced with any t-norm and t-conorm operator to suit the application. This series of minimum and maximum operations gives a satisfying intuitive interpretation to the knowledge processing. The indirect effect amounts to specifying the weakest causal link in the path and the total effect is specifying the strongest of the weakest links -- a very intuitively satisfying approach to evaluating a system.

Pelaez and Bowles [99] used FCMs to perform reliability and safety evaluations on complex systems, developing an automated Failure Modes and Effects Analysis (FMEA) methodology. Once the FCM for the system had been developed, they used a max-min inference approach to identify critical portions of the system that should be subjected to further examination or redesign. Their max-min inference approach consisted of the following:

1. Take the weakest (minimum) linguistic term of the links in the path and its weight as the path strength.

2. If the activated concepts in the path are linked by relationships described by the same linguistic term, that term is set to the highest (maximum) truth value of all the links included in the path.

3. If there is more than one path from the activating to the concluding concept, and it is described by different linguistic terms, the conclusion is set to the result of the defuzzification operation performed on the linguistic membership functions and their confidence values using a Weighted Mean of Maxima defuzzification scheme.

APPENDIX D

DEMPSTER-SHAFER THEORY

The *frame of discernment* represents the domain of the problem. Usually represented as Θ, this is the set of all possible hypotheses, mutually exclusive and exhaustive, in the conclusion space [100]. The frame of discernment is chosen such that it is relevant to the application. Shafer describes the process of choosing as [38]

> **It should not be thought that the possibilities that comprise Θ will be determined and meaningful independently of our knowledge. Quite to the contrary: Θ will acquire its meaning from what we know or think we know; the distinctions that it embodies will be embedded within the matrix of our language and its associated conceptual structures and will depend on those structure for whatever accuracy and meaningfulness they possess.**

For example, if a description of conclusions in an expert system are described in the (object, attribute, value) triple form, then the frame of discernment would be all the triples with the same object and attribute [101].

The ***basic probability assignment (bpa)*** or ***mass distribution*** is given to each subset of Θ, denoted by $m(\Psi)$ where $\Psi \subseteq \Theta$, indicating the portion of the total belief exactly committed to the hypothesis set Ψ, given a piece of evidence. The bpa must satisfy the following conditions [102]:

$$\sum_{\Psi \subseteq \Theta} m(\Psi) = 1 \qquad \textbf{(D-1)}$$

$$m(\varnothing) = 0 \qquad \textbf{(D-2)}$$

$$0 \le m(\Psi) \le 1, \text{ for all } \Psi \subseteq \Theta \qquad \textbf{(D-3)}$$

$m(\Psi)$ represents the direct support of evidence on the subset Ψ. One of the most appealing aspects of the D-S theory is that it does not require that the remaining belief be associated with the negation or complement of Ψ, it allows an assignment of belief to the entire frame of discernment, (i.e., $m(\Theta)$) thus allowing for the distinction between the known and the unknown. The value of $m(\Theta)$ represents the portion of the total belief that is uncommitted to any of the hypotheses after all the evidence has been gathered.

The ***belief function*** and ***plausibility function*** represent a lower and upper bound on the probability function and are calculated as [103]:

$$Bel(B) = \sum_{A \subset B} m(A) \qquad \textbf{(D-4)}$$

$$Pl(B) = \sum_{A \cap B \neq \varnothing} m(A) = 1 - Bel(\overline{B}) \qquad \textbf{(D-5)}$$

The belief function of a hypothesis, or its lower probability, is the sum of all the basic probability assignment values for all the proper subsets of that hypothesis. The distinction between the basic probability assignment and the belief function is that the bpa represents the amount of belief committed *exactly* to the hypothesis, while the belief function represents the *total* amount of belief committed to the hypothesis [104]. On the other hand, the plausibility function of a hypothesis, or its upper probability, represents the maximum amount of belief that can be committed to the hypothesis based upon the evidence. Alternatively, the plausibility can be viewed as the amount of belief that is not committed to the negation of the hypothesis.

The ***belief interval*** can be given for each hypothesis as the interval indicating the minimum (represented by the belief function) and the maximum (represented by the plausibility function) probability bounds on the hypothesis. The extent of the belief interval is a measure of the belief that is neither committed to the hypothesis nor it negation. Therefore, the belief interval can be interpreted as a measure of the ignorance or uncertainty in the hypothesis: the closer the value is to zero the more certain is the hypothesis, while the closer to unity the more uncertain. In most cases the belief interval will be somewhere in the range [0,1].

Finally, the ***degree of certainty*** is an index of the certainty associated with a decision, and is defined as [105]

$$DOC(X) = m(X) - Bel(\overline{X}) \qquad \textbf{(D-6)}$$

Using the definition of the plausibility function given in (D-5), the DOC can also be calculated directly from the basic probability assignment and plausibility function values, as

$$DOC(X) = m(X) + Pl(X) - 1 \qquad \textbf{(D-7)}$$

The range of values for DOC is [-1, 1], however, there are some values of the index indicating special circumstances about the decision being made. For example, when $DOC(\mathrm{X}) = 1$ all the evidence points exclusively to X because $m(\mathrm{X}) = 1$. DOC(X) = 0, indicates a case of total ignorance, since $m(\mathrm{X}) = \mathrm{Bel}(\overline{X})$. Finally, when DOC(X) = -1, the $\mathrm{Bel}(\overline{X})=1$ indicating that no evidence supports the piece of information represented by X.

The ***Dempster Rule of Combination*** provides a means of combining evidence once the bpa associated with each piece of evidence is derived. Given two pieces of evidence that provide information on the hypothesis Ψ denoted $m_1(\Psi)$ and $m_2(\Psi)$, the combined bpa is denoted $m_{12}(\Psi)$ and is given by:

$$m_{12}(\Psi) = \frac{\sum_{A \cap B = \Psi} m_1(A) m_2(B)}{1 - K} \qquad \textbf{(D-8)}$$

Where K is a normalization factor and is given by:

$$K = \sum_{A \cap B = \varnothing} m_1(A) m_2(B) \qquad \textbf{(D-9)}$$

The intersection tableau method of combination is suggested in [101], giving a straightforward and illustrative method of performing the combination. For example, consider the task of predicting a medical diagnosis based upon evidence gathered during an examination of the patient. One observation supports the diagnosis of hepatitis or cirrhosis of the liver to degree 0.6; m_1\{Hep, Cirr\} = 0.6. The other observation supports a diagnosis of cirrhosis, gallstones, or pancreatic cancer to a degree 0.7; m_2\{Cirr, Gall, Pan\} = 0.7. The combination of these two pieces of evidence are shown in the intersection tableau in Table D-1.

Table D-1 Intersection Tableau for m_1 with m_2

	$m_2\{Cirr, Gall, Pan\} = 0.7$	$m_2\{\Theta\} = 0.3$
$m_1\{Hep, Cirr\} = 0.6$	m_{12}\{Cirr\} = 0.42	m_{12}\{Hep, Cirr\} = 0.18
$m_1\{\Theta\} = 0.4$	m_{12}\{Cirr, Gall, Pan\} = 0.28	$m_{12}\{\Theta\} = 0.12$

$m_{12}\{\ \} = 0$ for all other subsets of Θ. In this example K is zero because there are no null intersections. If there is another observation that confirms the diagnosis of hepatitis to the degree 0.8, that evidence is combined with the current combination to get the final hypothesis as shown in the tableau of Table D-2.

Table D-2 Intersection Tableau for m_{12} with m_3

	$m_3\{Hep\} = 0.8$	$m_3\{\Theta\} = 0.2$
$m_{12}\{Cirr\} = 0.42$	$m_{123}\{\varnothing\} = 0.336$	$m_{123}\{\text{Cirr}\} = 0.084$
$m_{12}\{Hep, Cirr\} = 0.18$	$m_{123}\{\text{Hep}\} = 0.144$	$m_{123}\{\text{Hep, Cirr}\} = 0.036$
$m_{12}\{Cirr, Gall, Pan\} = 0.28$	$m_{123}\{\varnothing\} = 0.224$	$m_{12}\{\text{Cirr, Gall, Pan}\} = 0.056$
$m_{12}\{\Theta\} = 0.12$	$m_{123}\{\text{Hep}\} = 0.096$	$m_{123}\{\Theta\} = 0.024$

In this case, there are null entries and K is given the value of the sum of these. Therefore, the final basic probability assignments due to these combinations are

$K = 0.336 + 0.224 = 0.56$

$m_{123}\{\text{Hep}\} = (0.144 + 0.096) / (1 - 0.56) = 0.545$

$m_{123}\{\text{Cirr}\} = (0.084) / (1- 0.56) = 0.191$

$m_{123}\{\text{Hep, Cirr}\} = 0.36 / (1- 0.56) = 0.082$

$m_{123}\{\text{Cirr, Gall, Pan}\} = 0.056 / (1- 0.56) = 0.127$

$m_{123}\{\Theta\} = 0.024 / (1- 0.56) = 0.055$

APPENDIX E

SOURCE CODE

This appendix contains the C-language source code implementing the *Intelligent Hierarchical Decision Architecture's* Methodology. It is broken into three stages, described below.

Stage 1 (Clustering Methodology): The program reads the raw test data (for one logical system division and all the MOFPs) from a file (testdata.dat), forms the MOFP-level COMMFFYs using four compositional methods, then selects the optimal COMMFFY based upon the calculation of the COA/Mean Similarity Measure. The program then passes the optimal MOFP-level COMMFFYs (one for each MOFP) to Stage 2.

Stage 2 (Fuzzy Associative Memory): The optimal MOFP-level COMMFFYs from Stage 1 are used as input to stimulate the Fuzzy Associative Memory of Stage 2. The FAM serves to transform the MOFP-level COMMFFYs to a single MOTA-level COMMFFY. The code accomplishes this transformation by first forming a Degree of Membership vector representing each COMMFFY, then the FAM rules are used to make the transformation from each MOFP to the MOTA-level. Finally, the Reduction Theorem is used to aggregate across the MOTA-level vectors to form a single Degree of Membership vector at the MOTA-level. This Degree of Membership vector is then transformed into the MOTA-level COMMFFY which is passed to Stage 3.

Stage 3 (Fuzzy Cognitive Map): The MOTA-level COMMFFY is adjusted for factors which could not be controlled or included in the testing, using a Fuzzy Cognitive Map. The user is prompted for the Best-Case and Worst-Case adjustments to be made to the testing-based performance. Once these values (in the form of linguistic tags) have been supplied, the program adjusts the MOTA-level COMMFFY membership function values accordingly. The output from this stage is the MOTA-level, adjusted COMMFFY for a single logical division of the system.

The output from the program is dumped into two files. The entire set of results, including all interim results, are put into a file for review (decision.res). The Best-Case and Worst-Case MOTA-level COMMFFY values (for the logical division represented by the original data values) are put into a separate file to facilitate the processing of the aggregation in the final step of the *Intelligent Hierarchical Decision Architecture* (fcmreslt.res). The Aggregation Methodology, which is the final stage of the *Intelligent Hierarchical Decision Architecture* methodology, is accomplished off-line using the Dempster-Shafer Theory of Evidential Reasoning and therefore, is not included here.

E.1 MAIN PROGRAM CODE

Below is the main program, followed by the source code for all the functions included within the main. Comments are shown, when necessary to explain the code's rationale, in italics and right justified on the line prior to the code. The break between the three stages described above are highlighted with a comment that is bolded, italicized and centered.

Include Files

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

Define & Constant Declarations

```
#define MAX(A,B) (((A) > (B)) ? (A) : (B))
#define MIN(A,B) (((A) < (B)) ? (A) : (B))
#define SQR(X)  ((X)*(X))

#define MOFPS 6
#define DATAPTS 10
#define DOM 9
#define IndCOMMFFYSize 21
#define COMMFFY 101
#define MFPARAMS 5
#define METHODS 4
#define MAXMAX 0
#define MAXALL 1
#define MINMAX 2
#define MINALL 3
```

Function Precasts

```
FILE *Open_Data_File( char *filename, char *mode );
void Read_Data_File( FILE *, float[][DATAPTS] );
void Calc_Standard_Statistics( int, float[], float[] );
void Generate_MOFP_BMF( int MOFPNum, float BMFMOFP[DOM][MFPARAMS] );
void Read_MOFPMF_File( FILE *, float[][MFPARAMS] );
void Gen_DOM_Max_Max( float[], float[][MFPARAMS], float[] );
float Calc_Inc_MF( float, float, float );
float Calc_Dec_MF( float, float, float );
void Gen_DOM_Max_All( float[], float[][MFPARAMS], float[] );
void Gen_DOM_Min_Max( float[], float[][MFPARAMS], float[] );
void Gen_DOM_Min_All( float[], float[][MFPARAMS], float[] );
```

```
void Fill_COMMFFY_Vector( float[], float COMMFFYVec[] );
void Gen_Tri_COMMFFY( int, int, int, float, float[] );
void Gen_Open_Left_COMMFFY( int, int, int, float, float[] );
void Gen_Open_Right_COMMFFY( int, int, int, float, float[] );
float DeFuzz_By_COA( int, float[] );
float Calc_COA_Similarity_Measure( float, float );
int Determine_Optimal_Method( float[] );
void MOFP_to_MOTA_Transformation( float[][DOM], float[][DOM] );
void Calc_FAM_Aggregation( float[], float[][DOM], float[] );
void Get_FCM_Adjustment( float[] );
void Adjust_MOTA_COMMFFY( float, float[], float[] );


void main( void )
{
                                                  Input and output file pointers
   FILE *inputpt, *outputpt, *fcmresults;

                                                                 Raw Data Matrix
   float Data[MOFPS][DATAPTS];

                                               3-D Matrix holding all MOFP BMFs
   float MOFPBasicMF[MOFPS][DOM][MFPARAMS];

                             Degree of Membership Matrix to be used during optimization
   float METHMOFPDOM[METHODS][MOFPS][DOM];

                                   Degree of Membership Matrix to hold optimal results
   float MOFPDOM[MOFPS][DOM];

                                  COMMFFY Value Matrix to be used during optimization
   float METHMOFPCOMMFFY[METHODS][MOFPS][COMMFFY];

                                        COMMFFY Value Matrix to hold optimal result
   float MOFPCOMMFFY[MOFPS][COMMFFY];

                                                      Mean & Standard Deviation
   float Stats[MOFPS][2];

                               Results of Center of Area Defuzzification for all methods
   float COADeFuzz[MOFPS][METHODS];

                             Results of Center of Area Similarity Measures for all methods
   float COAMetric[MOFPS][METHODS];
```

Index of Optimal Method

```
int OptIndex[MOFPS];
```

MOTA-level Degree of Membership Matrix prior to reduction theorem aggregation

```
float MOTADOM[MOFPS][DOM];
```

Weights for the MOFP aggregation

```
float MOFPWEIGHTS[] = {1.0, 1.0, 1.0, 1.0, 1.0, 1.0};
```

Aggregated MOTA DOM using MIN method

```
float FAMMOTADOM[DOM];
```

MOTA COMMFFY generated from MIN aggregation

```
float FAMMOTACOMMFFY[COMMFFY];
```

Numerical values associated with best and worst case FCM hedges

```
float FCMHedge[2];
```

Best and worst case adjusted MOTA-level COMMFFY

```
float FCMMOTACOMMFFY[2][COMMFFY];

int i,j,k;
```

Open and read test data file into data matrix

```
inputpt = Open_Data_File("c:\\arc\\qed\\data\\testdata.dat", "r");
outputpt = Open_Data_File("c:\\arc\\qed\\results\\decision.res", "w+");
fcmresults = Open_Data_File("c:\\arc\\qed\\results\\fcmreslt.res", "w+");
Read_Data_File( inputpt, Data );
for( i=0; i<MOFPS; i++ )
   for( j=0; j<DATAPTS; j++ )
     fprintf( outputpt, "Data[%d][%d] = %5.2f \n", i,j, Data[i][j] );
```

Calculate mean and standard deviation of test data for comparison to COMMFFY metrics to determine best compositional method.

```
for( i=0; i<MOFPS; i++)
   Calc_Standard_Statistics( DATAPTS, Data[i], Stats[i] );

for( i=0; i<MOFPS; i++ )
   for( j=0; j<2; j++ )
     fprintf( outputpt, "Stats[%d][%d] = %5.2f \n", i,j, Stats[i][j] );
```

Make assignments of basic membership function values for each MOFP

```
for( i=0; i<MOFPS; i++)
   Generate_MOFP_BMF( i+1, MOFPBasicMF[i] );
```

Generate Degree of Membership Vectors for each MOFP based upon raw data using all the compositional methods.

```
for( i=0; i<MOFPS; i++)
{
   Gen_DOM_Max_Max( Data[i], MOFPBasicMF[i], METHMOFPDOM[MAXMAX][i] );
   Gen_DOM_Max_All( Data[i], MOFPBasicMF[i], METHMOFPDOM[MAXALL][i] );
   Gen_DOM_Min_Max( Data[i], MOFPBasicMF[i], METHMOFPDOM[MINMAX][i] );
   Gen_DOM_Min_All( Data[i], MOFPBasicMF[i], METHMOFPDOM[MINALL][i] );
}

for( i=0; i<METHODS; i++ )
   for( j=0; j<MOFPS; j++ )
      for( k=0; k<DOM; k++ )
         fprintf( outputpt, "METHMOFPDOM[%d][%d][%d] = %5.2f \n",
                  i,j,k, METHMOFPDOM[i][j][k] );
```

Generate COMMFFY Values from DOM Values using all compositional methods

```
for( i=0; i<METHODS; i++)
   for( j=0; j<MOFPS; j++)
      Fill_COMMFFY_Vector( METHMOFPDOM[i][j], METHMOFPCOMMFFY[i][j] );

for( i=0; i<METHODS; i++ )
   for( j=0; j<MOFPS; j++ )
      for( k=0; k<COMMFFY; k++ )
         fprintf( outputpt, "METHMOFPCOMMFFY[%d][%d][%d] = %5.2f \n",
                  i,j,k, METHMOFPCOMMFFY[i][j][k] );
```

Calculate COMMFFY centroid value using the Center of Area Defuzzication Scheme

```
for( i=0; i<MOFPS; i++)
   for( j=0; j<METHODS; j++)
   {
      COADeFuzz[i][j] = DeFuzz_By_COA( i+1, METHMOFPCOMMFFY[j][i] );
      fprintf( outputpt, "COADeFuzz[%d][%d] = %5.2f \n",
               i,j, COADeFuzz[i][j] );
```

```
}
```

Calculate the COA/Mean Similarity Measure for each MOFP/Method Combination

```
for( i=0; i<MOFPS; i++ )
  for( j=0; j<METHODS; j++ )
  {
    COAMetric[i][j] = Calc_COA_Similarity_Measure(COADeFuzz[i][j],
                                                  Stats[i][0] );
    fprintf( outputpt, "COAMetric[%d][%d] = %5.2f \n",
                  i,j, COAMetric[i][j] );
  }
```

Determine optimal compositional method from COA/Mean Similarity Measure

```
for( i=0; i<MOFPS; i++ )
{
  OptIndex[i] = Determine_Optimal_Method( COAMetric[i] );
  fprintf( outputpt, "MOFP[%d] Optimal Method is %d \n",
                i, OptIndex[i] );
}
```

Fill Degree of Membership Matrix with optimal method results

```
for( i=0; i<MOFPS; i++)
{
  k = OptIndex[i];
  for( j=0; j<DOM; j++ )
  {
    MOFPDOM[i][j] = METHMOFPDOM[k][i][j];
    fprintf( outputpt, "MOFPDOM[%d][%d] = %5.2f \n",
                  i,j, MOFPDOM[i][j] );
  }
}
```

Generate MOFP-level COMMFFY from the optimal generation method

```
for( i=0; i<MOFPS; i++)
    Fill_COMMFFY_Vector( MOFPDOM[i], MOFPCOMMFFY[i] );

for( i=0; i<MOFPS; i++ )                    //Output results to file
  for( j=0; j<COMMFFY; j++ )
```

```
fprintf( outputpt, "MOFPCOMMFFY[%d][%d] = %5.2f \n",
              i,j, MOFPCOMMFFY[i][j] );
```

End Clustering Methodology, Begin Fuzzy Associative Memory

Perform MOFP to MOTA transformation using Fuzzy Associative Memory

```
MOFP_to_MOTA_Transformation( MOFPDOM, MOTADOM );

for( i=0; i<MOFPS; i++ )
   for( j=0; j<DOM; j++ )
     fprintf( outputpt, "MOTADOM[%d][%d] = %5.2f \n", i,j, MOTADOM[i][j] );
```

Aggregate MOTA-transformed matrix value into a vector using Reduction Theorem

```
Calc_Gen_Mean_Aggregation( MOFPWEIGHTS, MOTADOM, FAMMOTADOM );

for( i=0; i<DOM; i++)
   fprintf( outputpt, "FAMMOTADOM[%d] = %5.2f \n", i, FAMMOTADOM[i] );
```

Generate MOTA COMMFFY Vector from the MOTA DOM Vector

```
Fill_COMMFFY_Vector( FAMMOTADOM, FAMMOTACOMMFFY );

for( i=0; i<COMMFFY; i++)
   fprintf( outputpt, "FAMMOTACOMMFFY[%d] = %5.2f \n",
              i, FAMMOTACOMMFFY[i] );
```

End Fuzzy Associative Memory, Begin Fuzzy Cognitive Map

Prompt user for hedges representing FCM-produced adjustments, then apply those adjustments to the FAMMOTACOMMFFY generated in the previous phases

```
Get_FCM_Adjustment( FCMHedge );

for( i=0; i<2; i++)
   Adjust_MOTA_COMMFFY( FCMHedge[i], FAMMOTACOMMFFY,
                                   FCMMOTACOMMFFY[i] );
```

```
for( i=0; i<2; i++)
   for( j=0; j<COMMFFY; j++)
   {
      fprintf( outputpt, "FCMMOTACOMMFFY[%d][%d] = %5.2f \n",
                    i, j, FCMMOTACOMMFFY[i][j] );
      fprintf( fcmresults, "%5.2f \n", FCMMOTACOMMFFY[i][j] );
   }
```

Close input and output files.

```
fclose(inputpt); fclose(outputpt); fclose(fcmresults);

}
```

E.2 FUNCTION CODE

Shown below are the functions called within the main program which implements the Intelligent Hierachical Decision Architecture Methodology. The functions are shown below in the order in which they appear in the *Intelligent Hierarchical Decision Architecture* main program. Each function begins with a general explanation of its functionality, and the parameters passed in and out of the function. Then, each function is commented as necessary to further explain any of the code's intricacies. All comments are shown in italics to enhance their separability them from the source code.

```
FILE *Open_Data_File( char *filename, char *mode )
{
Opens a file in a designed mode, and returns the filepointer to that file if successfully
opened.

   FILE *filepointer;

   if ((filepointer = fopen(filename, mode)) != NULL)
      printf("\nSuccessful opening %s in mode %s.\n", filename, mode);
   else
   {
      printf("\nUnable to open %s in mode %s.\n", filename, mode);
      exit(1);
   }
   return filepointer;
}
```

```
void Read_Data_File( FILE *filepointer, float DataMatrix[MOFPS][DATAPTS] )
{
Reads a data file pointed to by *filepointer, and puts its contents in a matrix named
DataMatrix.

   float value;
   int count=0, row, col;

   while( fscanf(filepointer, "%f", &value) == 1 )
   {
      row = count / DATAPTS;
      col = count % DATAPTS;
```

```
      DataMatrix[row][col] = value;
      count++;
   }
}
```

void Calc_Standard_Statistics(int n, float observations[], float stats[])

{

Calculates the mean and standard deviation of a vector of observations passed to it.

Pass in the number of observations in the vector and the vector of observations, and passes out a vector containing the mean in the first (0) position and the standard deviation in the
second (1) position.

```
   int i;
   float accum;

   for( i=0, accum=0; i<n; i++ )
      accum += observations[i];
   stats[0] = accum/n;
   for( i=0, accum=0; i<n; i++ )
      accum += SQR( observations[i] - stats[0] );
   stats[1] = sqrt(accum/(n-1));
}
```

void Generate_MOFP_BMF(int MOFPNum, float BMFMOFP[DOM][MFPARAMS])

{

Opens a file containing the basic membership function delimiters for a given MOFP and loads them into a matrix.

Pass in the MOFP number of interest and the name of the matrix to be filled the function fills the matrix with the information from the MOFP file.

```
   int row, col;
   FILE *inputMOFP;

   if (MOFPNum == 1)
   {
      inputMOFP = Open_Data_File("c:\\arc\\qed\\data\\mop1mf.dat", "r");
      Read_MOFPMF_File( inputMOFP, BMFMOFP );
      fclose(inputMOFP);
   }

   else if (MOFPNum == 2)
```

```
    {
      inputMOFP = Open_Data_File("c:\\arc\\qed\\data\\mop2mf.dat", "r");
      Read_MOFPMF_File( inputMOFP, BMFMOFP );
      fclose(inputMOFP);
    }

    else if (MOFPNum == 3)
    {
      inputMOFP = Open_Data_File("c:\\arc\\qed\\data\\mop3mf.dat", "r");
      Read_MOFPMF_File( inputMOFP, BMFMOFP );
      fclose(inputMOFP);
    }
    else if (MOFPNum == 4)
    {
      inputMOFP = Open_Data_File("c:\\arc\\qed\\data\\mop4mf.dat", "r");
      Read_MOFPMF_File( inputMOFP, BMFMOFP );
      fclose(inputMOFP);
    }
    else if (MOFPNum == 5)
    {
      inputMOFP = Open_Data_File("c:\\arc\\qed\\data\\mop5mf.dat", "r");
      Read_MOFPMF_File( inputMOFP, BMFMOFP );
      fclose(inputMOFP);
    }
    else if (MOFPNum == 6)
    {
      inputMOFP = Open_Data_File("c:\\arc\\qed\\data\\mop6mf.dat", "r");
      Read_MOFPMF_File( inputMOFP, BMFMOFP );
      fclose(inputMOFP);
    }
}
```

void Read_MOPMF_File(FILE *filepointer, float DataMatrix[DOM][MFPARAMS])

```
{
```

Reads in the limits of the Basic Membership Function values

```
    float value;
    int count=0, row, col;

    while( fscanf(filepointer, "%f", &value) == 1 )
    {
      row = count / MFPARAMS;
      col = count % MFPARAMS;
      DataMatrix[row][col] = value;
      count++;
    }
}
```

void Gen_DOM_Max_Max(float DataVec[10], float MF[DOM][MFPARAMS], float DOMVec[10])

{

Function calculates the degree of membership using the Max-Max Compositional Method. The Max-Max method activates the basic membership function for which each data point is a maximum, then takes the maximum degree within each basic membership function.

Pass in the vector holding the data and the matrix holding the parameters of the basic membership function and this function passes out a vector containing the degrees of membership for each of the basic membership functions.

```
int i;
float DOMTemp;
                                        Initialize all BMF's to zero DOM
for(i=0; i<DOM; i++)
   DOMVec[i]=0;

for(i=0; i<DATAPTS; i++)
{
   if ((DataVec[i] <= MF[0][2]) && (DataVec[i] >= MF[0][0]))
      DOMVec[0] = 1.0;

   else if ((MF[0][2] < DataVec[i]) && (DataVec[i] < MF[0][3]))
   {
      DOMTemp = Calc_Dec_MF(MF[0][2], MF[0][4], DataVec[i]);
      DOMVec[0] = MAX(DOMVec[0], DOMTemp);
   }
   else if (DataVec[i] == MF[0][3])
   {
      DOMVec[0] = MAX( 0.5, DOMVec[0] );
      DOMVec[1] = MAX( 0.5, DOMVec[1] );
   }
   else if ((MF[1][1] < DataVec[i]) && (DataVec[i] < MF[1][2]))
   {
      DOMTemp = Calc_Inc_MF(MF[1][0],MF[1][2], DataVec[i]);
      DOMVec[1] = MAX(DOMVec[1], DOMTemp);
   }
   else if (DataVec[i] == MF[1][2])
      DOMVec[1] = 1.0;

   else if ((MF[1][2] < DataVec[i]) && (DataVec[i] < MF[1][3]))
   {
      DOMTemp = Calc_Dec_MF(MF[1][2], MF[1][4], DataVec[i]);
      DOMVec[1] = MAX(DOMVec[1], DOMTemp);
   }
   else if (DataVec[i] == MF[1][3])
```

```
{
   DOMVec[1] = MAX( 0.5, DOMVec[1] );
   DOMVec[2] = MAX( 0.5, DOMVec[2] );
}
else if ((MF[2][1] < DataVec[i]) && (DataVec[i] < MF[2][2]))
{
   DOMTemp = Calc_Inc_MF(MF[2][0], MF[2][2], DataVec[i]);
   DOMVec[2] = MAX(DOMVec[2], DOMTemp);
}
else if (DataVec[i] == MF[2][2])
   DOMVec[2] = 1.0;

else if ((MF[2][2] < DataVec[i]) && (DataVec[i] < MF[2][3]))
{
   DOMTemp = Calc_Dec_MF(MF[2][2], MF[2][4], DataVec[i]);
   DOMVec[2] = MAX(DOMVec[2], DOMTemp);
}
else if (DataVec[i] == MF[2][3])
{
   DOMVec[2] = MAX( 0.5, DOMVec[2] );
   DOMVec[3] = MAX( 0.5, DOMVec[3] );
}
else if ((MF[3][1] < DataVec[i]) && (DataVec[i] < MF[3][2]))
{
   DOMTemp = Calc_Inc_MF(MF[3][0], MF[3][2], DataVec[i]);
   DOMVec[3] = MAX(DOMVec[3], DOMTemp);
}
else if (DataVec[i] == MF[3][2])
   DOMVec[3] = 1.0;

else if ((MF[3][2] < DataVec[i]) && (DataVec[i] < MF[3][3]))
{
   DOMTemp = Calc_Dec_MF(MF[3][2], MF[3][4], DataVec[i]);
   DOMVec[3] = MAX(DOMVec[3], DOMTemp);
}
else if (DataVec[i] == MF[3][3])
{
   DOMVec[3] = MAX( 0.5, DOMVec[3] );
   DOMVec[4] = MAX( 0.5, DOMVec[4] );
}
else if ((MF[4][1] < DataVec[i]) && (DataVec[i] < MF[4][2]))
{
   DOMTemp = Calc_Inc_MF(MF[4][0], MF[4][2], DataVec[i]);
   DOMVec[4] = MAX(DOMVec[4], DOMTemp);
}
else if (DataVec[i] == MF[4][2])
   DOMVec[4] = 1.0;
```

```
else if ((MF[4][2] < DataVec[i]) && (DataVec[i] < MF[4][3]))
{
  DOMTemp = Calc_Dec_MF(MF[4][2], MF[4][4], DataVec[i]);
  DOMVec[4] = MAX(DOMVec[4], DOMTemp);
}
else if (DataVec[i] == MF[4][3])
{
  DOMVec[4] = MAX( 0.5, DOMVec[4] );
  DOMVec[5] = MAX( 0.5, DOMVec[5] );
}
else if ((MF[5][1] < DataVec[i]) && (DataVec[i] < MF[5][2]))
{
  DOMTemp = Calc_Inc_MF(MF[5][0], MF[5][2], DataVec[i]);
  DOMVec[5] = MAX(DOMVec[5], DOMTemp);
}
else if (DataVec[i] == MF[5][2])
  DOMVec[5] = 1.0;

else if ((MF[5][2] < DataVec[i]) && (DataVec[i] < MF[5][3]))
{
  DOMTemp = Calc_Dec_MF(MF[5][2], MF[5][4], DataVec[i]);
  DOMVec[5] = MAX(DOMVec[5], DOMTemp);
}
else if (DataVec[i] == MF[5][3])
{
  DOMVec[5] = MAX( 0.5, DOMVec[5] );
  DOMVec[6] = MAX( 0.5, DOMVec[6] );
}
else if ((MF[6][1] < DataVec[i]) && (DataVec[i] < MF[6][2]))
{
  DOMTemp = Calc_Inc_MF(MF[6][0], MF[6][2], DataVec[i]);
  DOMVec[6] = MAX(DOMVec[6], DOMTemp);
}
else if (DataVec[i] == MF[6][2])
  DOMVec[6] = 1.0;

else if ((MF[6][2] < DataVec[i]) && (DataVec[i] < MF[6][3]))
{
  DOMTemp = Calc_Dec_MF(MF[6][2], MF[6][4], DataVec[i]);
  DOMVec[6] = MAX(DOMVec[6], DOMTemp);
}
else if (DataVec[i] == MF[6][3])
{
  DOMVec[6] = MAX( 0.5, DOMVec[6] );
  DOMVec[7] = MAX( 0.5, DOMVec[7] );
}
else if ((MF[7][1] < DataVec[i]) && (DataVec[i] < MF[7][2]))
{
```

```
            DOMTemp = Calc_Inc_MF(MF[7][0], MF[7][2], DataVec[i]);
            DOMVec[7] = MAX(DOMVec[7], DOMTemp);
        }
        else if (DataVec[i] == MF[7][2])
            DOMVec[7] = 1.0;

        else if ((MF[7][2] < DataVec[i]) && (DataVec[i] < MF[7][3]))
        {
            DOMTemp = Calc_Dec_MF(MF[7][2], MF[7][4], DataVec[i]);
            DOMVec[7] = MAX(DOMVec[7], DOMTemp);
        }
        else if (DataVec[i] == MF[7][3])
        {
            DOMVec[7] = MAX( 0.5, DOMVec[7] );
            DOMVec[8] = MAX( 0.5, DOMVec[8] );
        }
        else if ((MF[8][1] < DataVec[i]) && (DataVec[i] < MF[8][2]))
        {
            DOMTemp = Calc_Inc_MF(MF[8][0], MF[8][2], DataVec[i]);
            DOMVec[8] = MAX(DOMVec[8], DOMTemp);
        }
        else if ((DataVec[i] >= MF[8][2]) && (DataVec[i] <= MF[8][4]))
            DOMVec[8] = 1.0;
    }
}
```

float Calc_Dec_MF(float a, float b, float x)

{

Calculates the membership function value for a triangular shaped membership function's negative slope side.

```
    return (1.0 - (x-a)/(b-a));
}
```

float Calc_Inc_MF(float a, float b, float x)

{

Calculates the membership function value for a triangular shaped membership function's positive slope side

```
    return ((x-a)/(b-a));
}
```

void Gen_DOM_Max_All(float DataVec[10], float MF[DOM][MFPARAMS], float DOMVec[10])

{

Function calculates the degree of membership using the Max-All Compositional Method. The Max-All method activates all the basic membership functions for each data point, then takes the maximum degree within each basic membership function.

Pass in the vector holding the data, and the matrix holding the basic membership function limits and the function passes out a vector containing the degrees of membership for each of the basic membership functions.

```
   int i;
   float DOMTemp1, DOMTemp2;
                                                        Initialize all BMF's to zero DOM
   for(i=0; i<DOM; i++)
      DOMVec[i]=0;

   for(i=0; i<DATAPTS; i++)
   {
      if ((DataVec[i] <= MF[0][2]) && (DataVec[i] >= MF[0][0]))
         DOMVec[0] = 1.0;

      else if ((MF[0][2] < DataVec[i]) && (DataVec[i] < MF[0][4]))
      {
         DOMTemp1 = Calc_Dec_MF(MF[0][2], MF[0][4], DataVec[i]);
         DOMTemp2 = Calc_Inc_MF(MF[1][0], MF[1][2], DataVec[i]);
         DOMVec[0] = MAX(DOMVec[0], DOMTemp1);
         DOMVec[1] = MAX(DOMVec[1], DOMTemp2);
      }
      else if (DataVec[i] == MF[1][2])
         DOMVec[1] = 1.0;

      else if ((MF[1][2] < DataVec[i]) && (DataVec[i] < MF[1][4]))
      {
         DOMTemp1 = Calc_Dec_MF(MF[1][2], MF[1][4], DataVec[i]);
         DOMTemp2 = Calc_Inc_MF(MF[2][0], MF[2][2], DataVec[i]);
         DOMVec[1] = MAX(DOMVec[1], DOMTemp1);
         DOMVec[2] = MAX(DOMVec[2], DOMTemp2);
      }
      else if (DataVec[i] == MF[2][2])
         DOMVec[2] = 1.0;

      else if ((MF[2][2] < DataVec[i]) && (DataVec[i] < MF[2][4]))
      {
         DOMTemp1 = Calc_Dec_MF(MF[2][2], MF[2][4], DataVec[i]);
         DOMTemp2 = Calc_Inc_MF(MF[3][0], MF[3][2], DataVec[i]);
         DOMVec[2] = MAX(DOMVec[2], DOMTemp1);
```

```
DOMVec[3] = MAX(DOMVec[3], DOMTemp2);
}
else if (DataVec[i] == MF[3][2])
  DOMVec[3] = 1.0;

else if ((MF[3][2] < DataVec[i]) && (DataVec[i] < MF[3][4]))
{
  DOMTemp1 = Calc_Dec_MF(MF[3][2], MF[3][4], DataVec[i]);
  DOMTemp2 = Calc_Inc_MF(MF[4][0], MF[4][2], DataVec[i]);
  DOMVec[3] = MAX(DOMVec[3], DOMTemp1);
  DOMVec[4] = MAX(DOMVec[4], DOMTemp2);
}
else if (DataVec[i] == MF[4][2])
  DOMVec[4] = 1.0;

else if ((MF[4][2] < DataVec[i]) && (DataVec[i] < MF[4][4]))
{
  DOMTemp1 = Calc_Dec_MF(MF[4][2], MF[4][4], DataVec[i]);
  DOMTemp2 = Calc_Inc_MF(MF[5][0], MF[5][2], DataVec[i]);
  DOMVec[4] = MAX(DOMVec[4], DOMTemp1);
  DOMVec[5] = MAX(DOMVec[5], DOMTemp2);
}
else if (DataVec[i] == MF[5][2])
  DOMVec[5] = 1.0;

else if ((MF[5][2] < DataVec[i]) && (DataVec[i] < MF[5][4]))
{
  DOMTemp1 = Calc_Dec_MF(MF[5][2], MF[5][4], DataVec[i]);
  DOMTemp2 = Calc_Inc_MF(MF[6][0], MF[6][2], DataVec[i]);
  DOMVec[5] = MAX(DOMVec[5], DOMTemp1);
  DOMVec[6] = MAX(DOMVec[6], DOMTemp2);
}
else if (DataVec[i] == MF[6][2])
  DOMVec[6] = 1.0;

else if ((MF[6][2] < DataVec[i]) && (DataVec[i] < MF[6][4]))
{
  DOMTemp1 = Calc_Dec_MF(MF[6][2], MF[6][4], DataVec[i]);
  DOMTemp2 = Calc_Inc_MF(MF[7][0], MF[7][2], DataVec[i]);
  DOMVec[6] = MAX(DOMVec[6], DOMTemp1);
  DOMVec[7] = MAX(DOMVec[7], DOMTemp2);
}
else if (DataVec[i] == MF[7][2])
  DOMVec[7] = 1.0;

else if ((MF[7][2] < DataVec[i]) && (DataVec[i] < MF[7][4]))
{
  DOMTemp1 = Calc_Dec_MF(MF[7][2], MF[7][4], DataVec[i]);
```

```
        DOMTemp2 = Calc_Inc_MF(MF[8][0], MF[8][2], DataVec[i]);
        DOMVec[7] = MAX(DOMVec[7], DOMTemp1);
        DOMVec[8] = MAX(DOMVec[8], DOMTemp2);
      }
      else if ((DataVec[i] >= MF[8][2]) && (DataVec[i] <=MF[8][4]))
        DOMVec[8] = 1.0;
    }
}
```

void Gen_DOM_Min_Max(float DataVec[10], float MF[DOM][MFPARAMS], float DOMVec[10])

{

Function calculates the degree of membership for percentage based basic membership functions using the Min-Max Compositional Method. The Min-Max method activates the basic membership function for which each data point is a maximum, then takes the minimum degree within each basic membership function.

Pass in the vector holding the data and the matrix holding the basic membership function limits and the function passes out a vector containing the degrees of membership for each of the basic membership functions.

```
  int i;
  float DOMTemp;
                                              Initialize all BMF's to zero DOM
  for(i=0; i<DOM; i++)
    DOMVec[i]=0;

  for(i=0; i<DATAPTS; i++)
  {
    if ((DataVec[i] <= MF[0][2]) && (DataVec[i] >= MF[0][0]))
      DOMVec[0] = 1.0;

    else if ((MF[0][2] < DataVec[i]) && (DataVec[i] < MF[0][3]))
    {
      DOMTemp = Calc_Dec_MF(MF[0][2], MF[0][4], DataVec[i]);
      if (DOMVec[0] == 0.0)
        DOMVec[0] = DOMTemp;
      else
        DOMVec[0] = MIN(DOMVec[0], DOMTemp);
    }
    else if (DataVec[i] == MF[0][3])
    {
      if (DOMVec[0] == 0.0)
        DOMVec[0] = 0.5;
      else
```

```
        DOMVec[0] = MIN( 0.5, DOMVec[0] );
    if (DOMVec[1] == 0.0)
        DOMVec[1] = 0.5;
    else
        DOMVec[1] = MIN( 0.5, DOMVec[1] );
}
else if ((MF[1][1] < DataVec[i]) && (DataVec[i] < MF[1][2]))
{
    DOMTemp = Calc_Inc_MF(MF[1][0], MF[1][2], DataVec[i]);
    if (DOMVec[1] == 0.0)
        DOMVec[1] = DOMTemp;
    else
        DOMVec[1] = MIN(DOMVec[1], DOMTemp);
}
else if (DataVec[i] == MF[1][2])
    DOMVec[1] = 1.0;

else if ((MF[1][2] < DataVec[i]) && (DataVec[i] < MF[1][3]))
{
    DOMTemp = Calc_Dec_MF(MF[1][2], MF[1][4], DataVec[i]);
    if (DOMVec[1] == 0.0)
        DOMVec[1] = DOMTemp;
    else
        DOMVec[1] = MIN(DOMVec[1], DOMTemp);
}
else if (DataVec[i] == MF[1][3])
{
    if (DOMVec[1] == 0.0)
        DOMVec[1] = 0.5;
    else
        DOMVec[1] = MIN( 0.5, DOMVec[1] );
    if (DOMVec[2] == 0.0)
        DOMVec[2] = 0.5;
    else
        DOMVec[2] = MIN( 0.5, DOMVec[2] );
}
else if ((MF[2][1] < DataVec[i]) && (DataVec[i] < MF[2][2]))
{
    DOMTemp = Calc_Inc_MF(MF[2][0], MF[2][2], DataVec[i]);
    if (DOMVec[2] == 0.0)
        DOMVec[2] = DOMTemp;
    else
        DOMVec[2] = MIN(DOMVec[2], DOMTemp);
}
else if (DataVec[i] == MF[2][2])
    DOMVec[2] = 1.0;

else if ((MF[2][2] < DataVec[i]) && (DataVec[i] < MF[2][3]))
```

```
{
   DOMTemp = Calc_Dec_MF(MF[2][2], MF[2][4], DataVec[i]);
   if (DOMVec[2] == 0.0)
      DOMVec[2] = DOMTemp;
   else
      DOMVec[2] = MIN(DOMVec[2], DOMTemp);
}
else if (DataVec[i] == MF[2][3])
{
   if (DOMVec[2] == 0.0)
      DOMVec[2] = 0.5;
   else
      DOMVec[2] = MIN( 0.5, DOMVec[2] );
   if (DOMVec[3] == 0.0)
      DOMVec[3] = 0.5;
   else
      DOMVec[3] = MIN( 0.5, DOMVec[3] );
}
else if ((MF[3][1] < DataVec[i]) && (DataVec[i] < MF[3][2]))
{
   DOMTemp = Calc_Inc_MF(MF[3][0], MF[3][2], DataVec[i]);
   if (DOMVec[3] == 0.0)
      DOMVec[3] = DOMTemp;
   else
      DOMVec[3] = MIN(DOMVec[3], DOMTemp);
}
else if (DataVec[i] == MF[3][2])
   DOMVec[3] = 1.0;

else if ((MF[3][2] < DataVec[i]) && (DataVec[i] < MF[3][3]))
{
   DOMTemp = Calc_Dec_MF(MF[3][2], MF[3][4], DataVec[i]);
   if (DOMVec[3] == 0.0)
      DOMVec[3] = DOMTemp;
   else
      DOMVec[3] = MIN(DOMVec[3], DOMTemp);
}
else if (DataVec[i] == MF[3][3])
{
   if (DOMVec[3] == 0.0)
      DOMVec[3] = 0.5;
   else
      DOMVec[3] = MIN( 0.5, DOMVec[3] );
   if (DOMVec[4] == 0.0)
      DOMVec[4] = 0.5;
   else
      DOMVec[4] = MIN( 0.5, DOMVec[4] );
}
```

```
else if ((MF[4][1] < DataVec[i]) && (DataVec[i] < MF[4][2]))
{
  DOMTemp = Calc_Inc_MF(MF[4][0], MF[4][2], DataVec[i]);
  if (DOMVec[4] == 0.0)
    DOMVec[4] = DOMTemp;
  else
    DOMVec[4] = MIN(DOMVec[4], DOMTemp);
}
else if (DataVec[i] == MF[4][2])
  DOMVec[4] = 1.0;

else if ((MF[4][2] < DataVec[i]) && (DataVec[i] < MF[4][3]))
{
  DOMTemp = Calc_Dec_MF(MF[4][2], MF[4][4], DataVec[i]);
  if (DOMVec[4] == 0.0)
    DOMVec[4] = DOMTemp;
  else
    DOMVec[4] = MIN(DOMVec[4], DOMTemp);
}
else if (DataVec[i] == MF[4][3])
{
  if (DOMVec[4] == 0.0)
    DOMVec[4] = 0.5;
  else
    DOMVec[4] = MIN( 0.5, DOMVec[4] );
  if (DOMVec[5] == 0.0)
    DOMVec[5] = 0.5;
  else
    DOMVec[5] = MIN( 0.5, DOMVec[5] );
}
else if ((MF[5][1] < DataVec[i]) && (DataVec[i] < MF[5][2]))
{
  DOMTemp = Calc_Inc_MF(MF[5][0], MF[5][2], DataVec[i]);
  if (DOMVec[5] == 0.0)
    DOMVec[5] = DOMTemp;
  else
    DOMVec[5] = MIN(DOMVec[5], DOMTemp);
}
else if (DataVec[i] == MF[5][2])
  DOMVec[5] = 1.0;

else if ((MF[5][2] < DataVec[i]) && (DataVec[i] < MF[5][3]))
{
  DOMTemp = Calc_Dec_MF(MF[5][2], MF[5][4], DataVec[i]);
  if (DOMVec[5] == 0.0)
    DOMVec[5] = DOMTemp;
  else
    DOMVec[5] = MIN(DOMVec[5], DOMTemp);
```

```
}
else if (DataVec[i] == MF[5][3])
{
  if (DOMVec[5] == 0.0)
    DOMVec[5] = 0.5;
  else
    DOMVec[5] = MIN( 0.5, DOMVec[5] );
  if (DOMVec[6] == 0.0)
    DOMVec[6] = 0.5;
  else
    DOMVec[6] = MIN( 0.5, DOMVec[6] );
}
else if ((MF[6][1] < DataVec[i]) && (DataVec[i] < MF[6][2]))
{
  DOMTemp = Calc_Inc_MF(MF[6][0], MF[6][2], DataVec[i]);
  if (DOMVec[6] == 0.0)
    DOMVec[6] = DOMTemp;
  else
    DOMVec[6] = MIN(DOMVec[6], DOMTemp);
}
else if (DataVec[i] == MF[6][2])
  DOMVec[6] = 1.0;

else if ((MF[6][2] < DataVec[i]) && (DataVec[i] < MF[6][3]))
{
  DOMTemp = Calc_Dec_MF(MF[6][2], MF[6][4], DataVec[i]);
  if (DOMVec[6] == 0.0)
    DOMVec[6] = DOMTemp;
  else
    DOMVec[6] = MIN(DOMVec[6], DOMTemp);
}
else if (DataVec[i] == MF[6][3])
{
  if (DOMVec[6] == 0.0)
    DOMVec[6] = 0.5;
  else
    DOMVec[6] = MIN( 0.5, DOMVec[6] );
  if (DOMVec[7] == 0.0)
    DOMVec[7] = 0.5;
  else
    DOMVec[7] = MIN( 0.5, DOMVec[7] );
}
else if ((MF[7][1] < DataVec[i]) && (DataVec[i] < MF[7][2]))
{
  DOMTemp = Calc_Inc_MF(MF[7][0], MF[7][2], DataVec[i]);
  if (DOMVec[7] == 0.0)
    DOMVec[7] = DOMTemp;
  else
```

```
            DOMVec[7] = MIN(DOMVec[7], DOMTemp);
        }
        else if (DataVec[i] == MF[7][2])
          DOMVec[7] = 1.0;

        else if ((MF[7][2] < DataVec[i]) && (DataVec[i] < MF[7][3]))
        {
          DOMTemp = Calc_Dec_MF(MF[7][2], MF[7][4], DataVec[i]);
          if (DOMVec[7] == 0.0)
            DOMVec[7] = DOMTemp;
          else
            DOMVec[7] = MIN(DOMVec[7], DOMTemp);
        }
        else if (DataVec[i] == MF[7][3])
        {
          if (DOMVec[7] == 0.0)
            DOMVec[7] = 0.5;
          else
            DOMVec[7] = MIN( 0.5, DOMVec[7] );
          if (DOMVec[8] == 0.0)
            DOMVec[8] = 0.5;
          else
            DOMVec[8] = MIN( 0.5, DOMVec[8] );
        }
        else if ((MF[8][1] < DataVec[i]) && (DataVec[i] < MF[8][2]))
        {
          DOMTemp = Calc_Inc_MF(MF[8][0], MF[8][2], DataVec[i]);
          if (DOMVec[8] == 0.0)
            DOMVec[8] = DOMTemp;
          else
            DOMVec[8] = MIN(DOMVec[8], DOMTemp);
        }
        else if ((DataVec[i] >= MF[8][2]) && (DataVec[i] <= MF[8][4]))
          DOMVec[8] = 1.0;
      }
    }
```

void Gen_DOM_Min_All(float DataVec[10], float MF[DOM][MFPARAMS], float DOMVec[10])

{

Function calculates the degree of membership for using the Min-All Compositional Method. The Min-All method activates all the basic membership functions for which each data point, then takes the minimum degree within each basic membership function.

Pass in the vector holding the data and the matrix holding the basic membership function limits and the function passes out a vector containing the degrees of membership for each of the basic membership functions.

```
int i;
float DOMTemp1, DOMTemp2;
                                              Initialize all BMF's to zero DOM
for(i=0; i<DOM; i++)
   DOMVec[i]=0;

for(i=0; i<DATAPTS; i++)
{
   if ((DataVec[i] <= MF[0][2]) && (DataVec[i] >= MF[0][0]))
      DOMVec[0] = 1.0;

   else if ((MF[0][2] < DataVec[i]) && (DataVec[i] < MF[0][4]))
   {
      DOMTemp1 = Calc_Dec_MF(MF[0][2], MF[0][4], DataVec[i]);
      DOMTemp2 = Calc_Inc_MF(MF[1][0], MF[1][2], DataVec[i]);
      if (DOMVec[0] == 0)
         DOMVec[0] = DOMTemp1;
      else
         DOMVec[0] = MIN(DOMVec[0], DOMTemp1);
      if (DOMVec[1] == 0)
         DOMVec[1] = DOMTemp2;
      else
         DOMVec[1] = MIN(DOMVec[1], DOMTemp2);
   }
   else if (DataVec[i] == MF[1][2])
      DOMVec[1] = 1.0;

   else if ((MF[1][2] < DataVec[i]) && (DataVec[i] < MF[1][4]))
   {
      DOMTemp1 = Calc_Dec_MF(MF[1][2], MF[1][4], DataVec[i]);
      DOMTemp2 = Calc_Inc_MF(MF[2][0], MF[2][2], DataVec[i]);
      if (DOMVec[1] == 0)
         DOMVec[1] = DOMTemp1;
      else
         DOMVec[1] = MIN(DOMVec[1], DOMTemp1);
      if (DOMVec[2] == 0)
```

```
      DOMVec[2] = DOMTemp2;
   else
      DOMVec[2] = MIN(DOMVec[2], DOMTemp2);
}
else if (DataVec[i] == MF[2][2])
   DOMVec[2] = 1.0;

else if ((MF[2][2] < DataVec[i]) && (DataVec[i] < MF[2][4]))
{
   DOMTemp1 = Calc_Dec_MF(MF[2][2], MF[2][4], DataVec[i]);
   DOMTemp2 = Calc_Inc_MF(MF[3][0], MF[3][2], DataVec[i]);
   if (DOMVec[2] == 0)
      DOMVec[2] = DOMTemp1;
   else
      DOMVec[2] = MIN(DOMVec[2], DOMTemp1);
   if (DOMVec[3] == 0)
      DOMVec[3] = DOMTemp2;
   else
      DOMVec[3] = MIN(DOMVec[3], DOMTemp2);
}
else if (DataVec[i] == MF[3][2])
   DOMVec[3] = 1.0;

else if ((MF[3][2] < DataVec[i]) && (DataVec[i] < MF[3][4]))
{
   DOMTemp1 = Calc_Dec_MF(MF[3][2], MF[3][4], DataVec[i]);
   DOMTemp2 = Calc_Inc_MF(MF[4][0], MF[4][2], DataVec[i]);
   if (DOMVec[3] == 0)
      DOMVec[3] = DOMTemp1;
   else
      DOMVec[3] = MIN(DOMVec[3], DOMTemp1);
   if (DOMVec[4] == 0)
      DOMVec[4] = DOMTemp2;
   else
      DOMVec[4] = MIN(DOMVec[4], DOMTemp2);
}
else if (DataVec[i] == MF[4][2])
   DOMVec[4] = 1.0;

else if ((MF[4][2] < DataVec[i]) && (DataVec[i] < MF[4][4]))
{
   DOMTemp1 = Calc_Dec_MF(MF[4][2], MF[4][4], DataVec[i]);
   DOMTemp2 = Calc_Inc_MF(MF[5][0], MF[5][2], DataVec[i]);
   if (DOMVec[4] == 0)
      DOMVec[4] = DOMTemp1;
   else
      DOMVec[4] = MIN(DOMVec[4], DOMTemp1);
   if (DOMVec[5] == 0)
```

```
      DOMVec[5] = DOMTemp2;
   else
      DOMVec[5] = MIN(DOMVec[5], DOMTemp2);
}
else if (DataVec[i] == MF[5][2])
   DOMVec[5] = 1.0;

else if ((MF[5][2] < DataVec[i]) && (DataVec[i] < MF[5][4]))
{
   DOMTemp1 = Calc_Dec_MF(MF[5][2], MF[5][4], DataVec[i]);
   DOMTemp2 = Calc_Inc_MF(MF[6][0], MF[6][2], DataVec[i]);
   if (DOMVec[5] == 0)
      DOMVec[5] = DOMTemp1;
   else
      DOMVec[5] = MIN(DOMVec[5], DOMTemp1);
   if (DOMVec[6] == 0)
      DOMVec[6] = DOMTemp2;
   else
      DOMVec[6] = MIN(DOMVec[6], DOMTemp2);
}
else if (DataVec[i] == MF[6][2])
   DOMVec[6] = 1.0;

else if ((MF[6][2] < DataVec[i]) && (DataVec[i] < MF[6][4]))
{
   DOMTemp1 = Calc_Dec_MF(MF[6][2], MF[6][4], DataVec[i]);
   DOMTemp2 = Calc_Inc_MF(MF[7][0], MF[7][2], DataVec[i]);
   if (DOMVec[6] == 0)
      DOMVec[6] = DOMTemp1;
   else
      DOMVec[6] = MIN(DOMVec[6], DOMTemp1);
   if (DOMVec[7] == 0)
      DOMVec[7] = DOMTemp2;
   else
      DOMVec[7] = MIN(DOMVec[7], DOMTemp2);
}
else if (DataVec[i] == MF[7][2])
   DOMVec[7] = 1.0;

else if ((MF[7][2] < DataVec[i]) && (DataVec[i] < MF[7][4]))
{
   DOMTemp1 = Calc_Dec_MF(MF[7][2], MF[7][4], DataVec[i]);
   DOMTemp2 = Calc_Inc_MF(MF[8][0], MF[8][2], DataVec[i]);
   if (DOMVec[7] == 0)
      DOMVec[7] = DOMTemp1;
   else
      DOMVec[7] = MIN(DOMVec[7], DOMTemp1);
   if (DOMVec[8] == 0)
```

```
            DOMVec[8] = DOMTemp2;
         else
            DOMVec[8] = MIN(DOMVec[8], DOMTemp2);
      }
      else if ((DataVec[i] >= MF[8][2]) && (DataVec[i] <= MF[8][4]))
         DOMVec[8] = 1.0;
   }
}
```

void Fill_COMMFFY_Vector(float DOMVec[9], float COMMFFYVec[101])

{

Function fills a two-dimensional matrix with the membership function values for each basic membership function. The values in this matrix are subsequently used to calculate the COMMFFY vector values.

Pass in the Degree of Membership vector, the function creates the two-dimensional array of membership function values within each basic membership function. Then based upon this matrix, the function creates the vector with the digitized COMMFFY values and passes that
vector out.

```
   int i, j, k, m, n;
   float Ind2COMMFFY[9][21];

   for (i=0;i<9;i++)
   {
      for (j=0;j<21;j++)
         Ind2COMMFFY[i][j] = 0.0;
   }

   Gen_Open_Left_COMMFFY( 0, 10, 20, DOMVec[0], Ind2COMMFFY[0] );

   for (j=1;j<8;j++)
      Gen_Tri_COMMFFY( j*10, j*10+10, j*10+20, DOMVec[j], Ind2COMMFFY[j] );

   Gen_Open_Right_COMMFFY( 80, 90, 100, DOMVec[8], Ind2COMMFFY[8] );

   for (i=0;i<10;i++)
      COMMFFYVec[i] = Ind2COMMFFY[0][i];

   for (i=10;i<91;i++)
   {
      j = (i/10) - 1;
      k =  i - j*10;
      m =  i/10;
```

```
        n =  i - m*10;
        COMMFFYVec[i] = MAX( Ind2COMMFFY[j][k], Ind2COMMFFY[m][n] );
    }

    for (i=91;i<101;i++)
        COMMFFYVec[i] = Ind2COMMFFY[8][i-80];
}
```

void Gen_Tri_COMMFFY(int a, int b, int c, float DOMVal, float Ind1COMMFFY[IndCOMMFFYSize])

{

Generates the digitized Composite Fuzzy Membership Function values for a triangular-shaped membership function.

Pass in the beginning, middle, and ending point of the membership function and the degree of membership within. Function passes out a vector containing the individual COMMFFY values.

```
    int i;

    for( i=a; i <= (a + DOMVal*10); i++ )
        Ind1COMMFFY[i-a] = ((float)i-a)/(b-a);

    for( i=ceil(a + DOMVal*10); i <= (c - DOMVal*10); i++ )
        Ind1COMMFFY[i-a] = DOMVal;

    for( i=ceil(c - DOMVal*10); i<=c; i++ )
        Ind1COMMFFY[i-a] = 1-(((float)i-b)/(c-b));

}
```

void Gen_Open_Left_COMMFFY(int a, int b, int c, float DOMVal, float Ind1COMMFFY[IndCOMMFFYSize])

{

Generates the digitized Composite Fuzzy Membership Function (COMMFFY) values for a triangular-shaped membership function with an open left end.

Pass in the beginning, middle, and ending point of the membership function and the degree of membership within. Function passes out a vector containing the COMMFFY values

```
    int i;

    for( i=a; i <= (c - DOMVal*10); i++ )
```

```
      Ind1COMMFFY[i-a] = MIN(DOMVal,1.0);

   for( i=ceil(c - DOMVal*10); i<=c; i++ )
      Ind1COMMFFY[i-a] = 1-(((float)i-b)/(c-b));

}
```

void Gen_Open_Right_COMMFFY(int a, int b, int c, float DOMVal, float Ind1COMMFFY[IndCOMMFFYSize])

{

Function generates the digitized composite fuzzy membership function values for a triangular-shaped membership function with an open right end.

Pass in the beginning, middle, and ending point of the membership function and the degree of membership within. Function passes out a vector containing the COMMFFY values.

```
   int i;

   for( i=a; i <= (a + DOMVal*10); i++ )
      Ind1COMMFFY[i-a] = (((float)i-a)/(b-a));

   for( i=ceil(a + DOMVal*10); i<=c; i++ )
      Ind1COMMFFY[i-a] = MIN(DOMVal,1.0);

}
```

float DeFuzz_By_COA(float COMMFFYVec[101])

{

Calculates the centroid of a COMMFFY based upon the Center of Area Defuzzification scheme.

Pass in the COMMFFY Vector, and the function returns the defuzzified value.

```
   int i;
   float defuzzval, sumoweights=0, sumoposweights=0;

   if( MOFPNum == 6 )
   {
      for( i=0; i<=20; i++ )
      {
         sumoweights += COMMFFYVec[i];
         sumoposweights += (i/10.0) * COMMFFYVec[i];
      }
```

```
    for( i=21; i<101; i++ )
    {
       sumoweights += COMMFFYVec[i];
       sumoposweights += ((i/5) -2.0) * COMMFFYVec[i];
    }
  }
  else
  {
    for( i=0; i<101; i++ )
    {
       sumoweights += COMMFFYVec[i];
       sumoposweights += i * COMMFFYVec[i];
    }
  }
  return (sumoposweights/sumoweights);
}
```

float Calc_COA_Similarity_Measure(float DeFuzzVal, float Mean)

{

Calculates the COA/Mean Similarity Measure.

Pass in the COA-generated defuzzified value and the mean the function returns the similarity measure value.

```
  return ( ( ( fabs(DeFuzzVal - Mean) ) / Mean ) * 100 );
}
```

int Determine_Optimal_Method(float Metrics[METHODS])

{

Determines which of the compositional methods yielded the minimum value of the similarity measure.

Pass in the row of the matrix containing the similarity measure the function returns the index of the method with the minimum value.

```
  int i, indexval;
  float currentminval;

  indexval = 0;
  currentminval = Metrics[0];

  for( i=1; i<METHODS; i++ )
  {
    if (currentminval > Metrics[i])
```

```
    {
        currentminval = Metrics[i];
        indexval = i;
    }
  }
  return indexval;
}
```

void MOFP_to_MOTA_Transformation(float MOFPMatrix[MOFPS][DOM], float MOTAMatrix[MOFPS][DOM])

{

Performs the MOFP to MOTA transformation, based upon the rules defined for a given system. It is just a coded look-up table with the rules hardcoded in. When the rules change, the relationships in this function must be modified.

The rules coded here are those in the FAM for the OT&E testbed case.

Pass in the matrix containing the degrees of membership for the MOFPs and the function passes out the degrees of membership for the MOTA.

Reduction in Hits to P_k transform

```
  MOTAMatrix[0][0] = MOFPMatrix[0][0];
  MOTAMatrix[0][1] = MOFPMatrix[0][1];
  MOTAMatrix[0][2] = MOFPMatrix[0][2];
  MOTAMatrix[0][3] = MOFPMatrix[0][3];
  MOTAMatrix[0][4] = MOFPMatrix[0][4];
  MOTAMatrix[0][5] = MOFPMatrix[0][5];
  MOTAMatrix[0][6] = MOFPMatrix[0][6];
  MOTAMatrix[0][7] = MOFPMatrix[0][7];
  MOTAMatrix[0][8] = MOFPMatrix[0][8];
```

Reduction in Guidance to P_k transform

```
  MOTAMatrix[1][0] = MOFPMatrix[1][1];
  MOTAMatrix[1][1] = MOFPMatrix[1][4];
  MOTAMatrix[1][2] = MOFPMatrix[1][4];
  MOTAMatrix[1][3] = MOFPMatrix[1][5];
  MOTAMatrix[1][4] = MOFPMatrix[1][6];
  MOTAMatrix[1][5] = MOFPMatrix[1][6];
  MOTAMatrix[1][6] = MOFPMatrix[1][7];
  MOTAMatrix[1][7] = MOFPMatrix[1][8];
  MOTAMatrix[1][8] = MOFPMatrix[1][8];
```

Reduction in Track Time to P_k transform

```
  MOTAMatrix[2][0] = MOFPMatrix[2][1];
  MOTAMatrix[2][1] = MOFPMatrix[2][4];
  MOTAMatrix[2][2] = MOFPMatrix[2][4];
  MOTAMatrix[2][3] = MOFPMatrix[2][5];
```

```
    MOTAMatrix[2][4] = MOFPMatrix[2][6];
    MOTAMatrix[2][5] = MOFPMatrix[2][6];
    MOTAMatrix[2][6] = MOFPMatrix[2][7];
    MOTAMatrix[2][7] = MOFPMatrix[2][8];
    MOTAMatrix[2][8] = MOFPMatrix[2][8];
```

Track on Jam to P_k transform

```
    MOTAMatrix[3][0] = MOFPMatrix[3][0];
    MOTAMatrix[3][1] = MOFPMatrix[3][2];
    MOTAMatrix[3][2] = MOFPMatrix[3][3];
    MOTAMatrix[3][3] = MOFPMatrix[3][5];
    MOTAMatrix[3][4] = MOFPMatrix[3][6];
    MOTAMatrix[3][5] = MOFPMatrix[3][7];
    MOTAMatrix[3][6] = MOFPMatrix[3][7];
    MOTAMatrix[3][7] = MOFPMatrix[3][7];
    MOTAMatrix[3][8] = MOFPMatrix[3][7];
```

Increase in Track Error to P_k transform

```
    MOTAMatrix[4][0] = MOFPMatrix[4][0];
    MOTAMatrix[4][1] = MOFPMatrix[4][1];
    MOTAMatrix[4][2] = MOFPMatrix[4][2];
    MOTAMatrix[4][3] = MOFPMatrix[4][3];
    MOTAMatrix[4][4] = MOFPMatrix[4][4];
    MOTAMatrix[4][5] = MOFPMatrix[4][5];
    MOTAMatrix[4][6] = MOFPMatrix[4][6];
    MOTAMatrix[4][7] = MOFPMatrix[4][7];
    MOTAMatrix[4][8] = MOFPMatrix[4][8];
```

Delay Time to P_k transform

```
    MOTAMatrix[5][0] = MOFPMatrix[5][6];
    MOTAMatrix[5][1] = MOFPMatrix[5][6];
    MOTAMatrix[5][2] = MOFPMatrix[5][6];
    MOTAMatrix[5][3] = MOFPMatrix[5][6];
    MOTAMatrix[5][4] = MOFPMatrix[5][5];
    MOTAMatrix[5][5] = MOFPMatrix[5][3];
    MOTAMatrix[5][6] = MOFPMatrix[5][1];
    MOTAMatrix[5][7] = MOFPMatrix[5][0];
    MOTAMatrix[5][8] = MOFPMatrix[5][0];
}
```

void Calc_FAM_Aggregation(float Weights[MOFPS], float MOTAMatrix[MOFPS][DOM], float FAMAggrMOTA[DOM])

{

Function calculates the aggregated MOTA degree of membership across all MOFPs using a modified Reduction Theorem result. If the number of MOFPs being aggregated having zero membership function exceeds half of the available, a value of zero is assigned. Otherwise, the maximum value of the membership function values is used.

Pass in the matrix containing the DOM values for each MOFP transformed to the MOTA level and a vector containing the MOFP weight values, the function aggregates across the MOFPs and passes out a vector containing the aggregation result.

```
    int col,row, zerocount;

    for( col=0; col<DOM; col++ )
    {
        zerocount=0;
        for( row=0; row<MOFPS; row++ )
            if ( MOTAMatrix[row][col] == 0 )
                zerocount++;

        if ( zerocount > (MOFPS/2) )
            FAMAggrMOTA[col] = 0;

        else
        {
            FAMAggrMOTA[col] = MOTAMatrix[0][col] * Weights[0];
            for( row=1; row<MOFPS; row++ )
                FAMAggrMOTA[col] = MAX((MOTAMatrix[row][col] * Weights[row]),
                                       FAMAggrMOTA[col]);
        }
    }
}
```

void Get_FCM_Adjustment(float FCMTags[2])

{

Prompts user to enter the value of the best-case and worst-case adjustment for the MOTA-level COMMFFYs based upon the changes suggested by the FCM.

```
    int choice;

    printf( "\n\n Input choice for BEST CASE adjustment: ");
    printf( "\n Select 1 for VERY ");
    printf( "\n Select 2 for MUCH ");
    printf( "\n Select 3 for SOME ");
    printf( "\n Select 4 for LITTLE ");
    printf( "\n Select 5 to input a numerical value ");

    scanf( "%d", &choice );

    if (choice == 1)
```

```
        FCMTags[0] = 0.5;
    else if (choice == 2)
        FCMTags[0] = 0.6667;
    else if (choice == 3)
        FCMTags[0] = 0.8;
    else if (choice == 4)
        FCMTags[0] = 0.9091;
    else if (choice == 5)
    {
        printf( "\n\n Input numerical adjustment for BEST CASE adjustment: ");
        scanf( "%f", &FCMTags[0] );
    }

    printf( "\n\n Input choice for WORST CASE adjustment: ");
    printf( "\n Select 1 for VERY ");
    printf( "\n Select 2 for MUCH ");
    printf( "\n Select 3 for SOME ");
    printf( "\n Select 4 for LITTLE ");
    printf( "\n Select 5 to input a numerical value ");

    scanf( "%d", &choice );

    if (choice == 1)
        FCMTags[1] = 2.0;
    else if (choice == 2)
        FCMTags[1] = 1.5;
    else if (choice == 3)
        FCMTags[1] = 1.25;
    else if (choice == 4)
        FCMTags[1] = 1.1;
    else if (choice == 5)
    {
        printf( "\n\n Input numerical adjustment for WORST CASE adjustment: ");
        scanf( "%f", &FCMTags[1] );
    }
}
```

void Adjust_MOTA_COMMFFY(float adjuster, float before[101], float after[101])

{

Adjusts the MOTA-level COMMFFY for the FCM-driven adjustment.

Pass in the value of the adjuster and the digitized COMMFFY and the function passes out the adjusted COMMFFY.

```
    int i;
```

```
    for ( i=0; i<COMMFFY; i++ )
    {
        if ( before[i] == 0 )
            after[i] = 0;
        else
            after[i] = exp( adjuster * log(before[i]) );
    }
}
```

Calculation to raise a value to a non-integer exponent.

APPENDIX F

TESTBED CASE RESULTS

This Appendix contains the full results for the application of the *Intelligent Hierarchical Decision Architecture's* methodology to the testbed case -- Jammer X. Shown are the raw data used as the initial input to the system and the output of each of the major stages of the *Intelligent Hierarchical Decision Architecture*: the Clustering Methodology, the Fuzzy Associative Memory, the Fuzzy Cognitive Map, and the Aggregation Methodology.

B.1 RAW TEST DATA

The raw test data gathered during the OT&E of the Jammer-X system are shown below. Ten observations of each *Measure of Functional Performance* against each threat system were gathered, yielding a total of 240 data points. The data are arranged by *Measure of Functional Performance* and Threat System in Table F-1 through Table F-6.

Table F-1 Test Data Collected for MOFP #1, Percent Reduction in Hits

	Percent Reduction in Hits			
Run Number	Threat A	Threat B	Threat C	Threat D
1	92.88	51.67	44.76	71.43
2	75.22	52.11	30.00	32.59
3	100.00	57.28	39.15	80.16
4	81.68	64.62	40.34	71.68
5	85.14	61.54	41.67	60.34
6	82.80	63.94	32.78	61.54
7	87.45	54.62	32.58	49.15
8	81.36	59.05	33.33	50.88
9	79.21	50.91	35.00	41.36
10	87.63	62.80	32.50	37.63

Table F-2 Test Data Collected for MOFP #2, Percent Reduction in Guidance

	Percent Reduction in Guidance			
Run Number	Threat A	Threat B	Threat C	Threat D
1	93.47	67.89	38.35	52.97
2	83.10	67.93	38.34	67.11
3	93.04	51.94	41.75	30.77
4	84.62	52.97	41.60	38.34
5	91.03	67.11	32.82	16.68
6	76.22	68.68	36.52	41.75
7	73.61	58.90	36.25	68.68
8	75.64	52.69	43.61	58.90
9	99.10	65.39	44.70	73.04
10	98.26	63.26	32.81	84.02

Table F-3 Test Data Collected for MOFP #3, Percent Increase in Break Locks

	Percent Increase in Break Locks			
Run Number	Threat A	Threat B	Threat C	Threat D
1	76.65	57.17	43.64	23.78
2	89.77	53.44	47.77	35.93
3	90.92	55.46	35.93	16.65
4	90.47	62.48	48.65	48.65
5	98.66	62.95	31.90	52.97
6	94.78	51.33	49.40	68.85
7	91.38	59.11	38.41	59.11
8	94.10	53.73	46.44	46.15
9	76.15	52.97	30.63	77.02
10	77.02	51.63	35.10	76.15

Table F-4 Test Data Collected for MOFP #4, Percent Track on Jam

	Percent Track on Jam			
Run Number	Threat A	Threat B	Threat C	Threat D
1	83.37	52.10	49.42	16.03
2	98.69	56.29	44.16	50.97
3	85.62	69.49	44.28	29.25
4	73.90	54.70	32.72	76.71
5	97.33	66.47	49.13	66.47
6	94.92	61.66	32.94	47.24
7	82.58	56.26	38.91	51.92
8	80.21	67.18	36.86	68.74
9	72.19	51.92	37.25	69.55
10	77.19	52.63	42.39	22.17

Table F-5 Test Data Collected for MOFP #5, Percent Increase in Break Locks

	Percent Increase in Track Error			
Run Number	**Threat A**	**Threat B**	**Threat C**	**Threat D**
1	70.98	65.21	31.75	23.99
2	93.79	68.13	38.58	18.09
3	88.70	58.72	38.77	31.75
4	84.56	59.01	49.97	88.70
5	95.54	55.61	40.88	65.21
6	98.33	56.49	48.85	46.22
7	96.11	62.93	46.40	43.93
8	80.31	65.13	31.92	67.96
9	90.17	62.16	38.92	45.53
10	88.56	50.60	42.65	62.16

Table F-6 Test Data Collected for MOFP #6, Jammer Response Time

	Response Time (Seconds)			
Run Number	**Threat A**	**Threat B**	**Threat C**	**Threat D**
1	4.55	8.36	10.16	7.83
2	3.49	8.28	11.74	5.17
3	2.50	9.52	14.30	10.16
4	4.71	9.87	13.53	8.11
5	4.62	8.60	15.04	11.74
6	4.39	9.44	12.73	13.92
7	1.99	9.89	11.94	12.52
8	2.15	8.06	13.51	6.92
9	3.65	8.81	12.75	4.61
10	3.19	8.90	11.68	2.86

F.2 CLUSTERING METHODOLOGY

The Clustering Methodology serves to form a Composite Fuzzy Membership Function from the raw test data for each Threat/MOFP combination. The Clustering Methodology, as described in Chapter 3, is composed of two basic steps. First, development of the Basic Membership Functions either heuristically or through a fuzzy clustering of previous data. Second, use of the On-line Optimization Method to form the optimal Composite Fuzzy Membership Function from the raw test data and the available Compositional Methods. The results of the Clustering Method for the testbed case are shown in Figure F-1 through Figure F-24. The input to this stage was 240 individual data points (10 observations for each threat/measure of functional performance combination) and the output is 24 functional performance level COMMFFYs (one per threat/measure of functional performance combination).

B.2.1 Threat A Composite Fuzzy Membership Functions

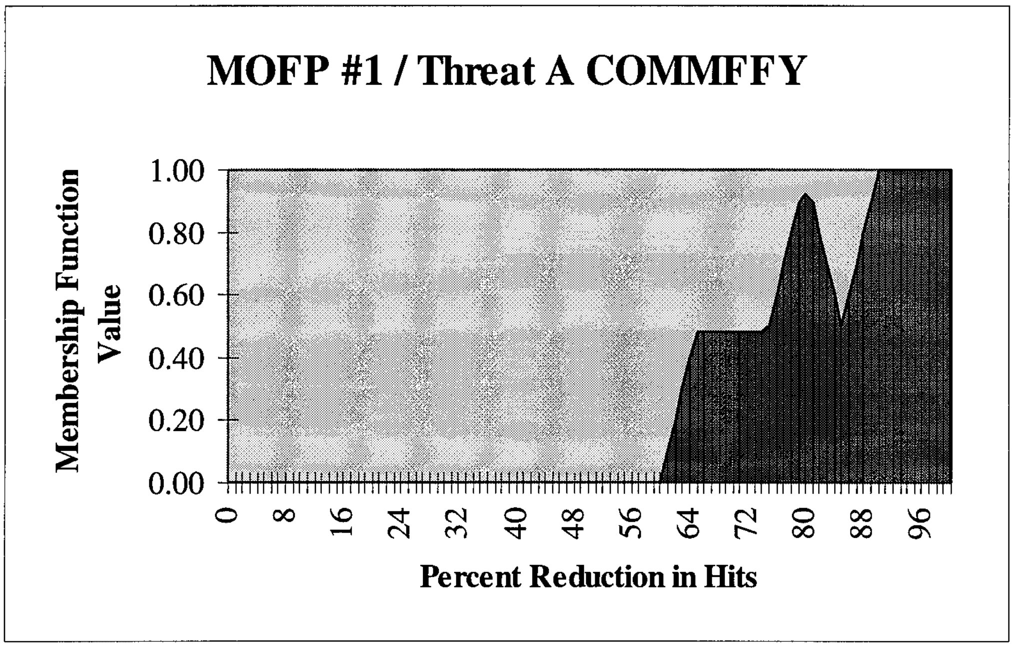

Figure F-1 COMMFFY Indicating MOFP #1 (Percent Reduction in Hits) Performance Against Threat A

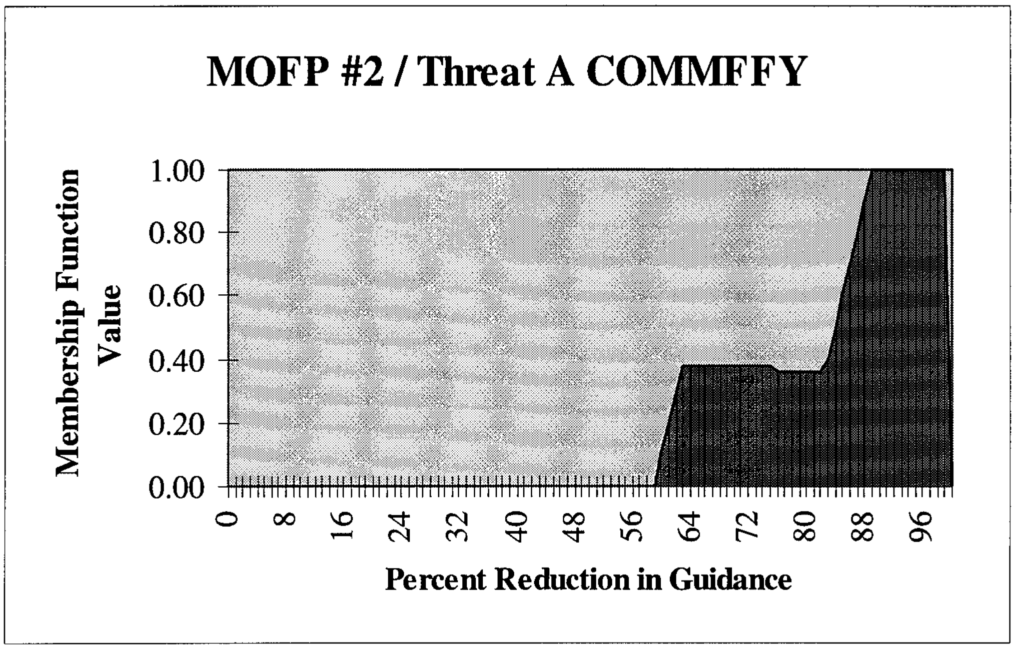

Figure F-2 COMMFFY Indicating MOFP #2 (Percent Reduction in Guidance) Performance Against Threat A

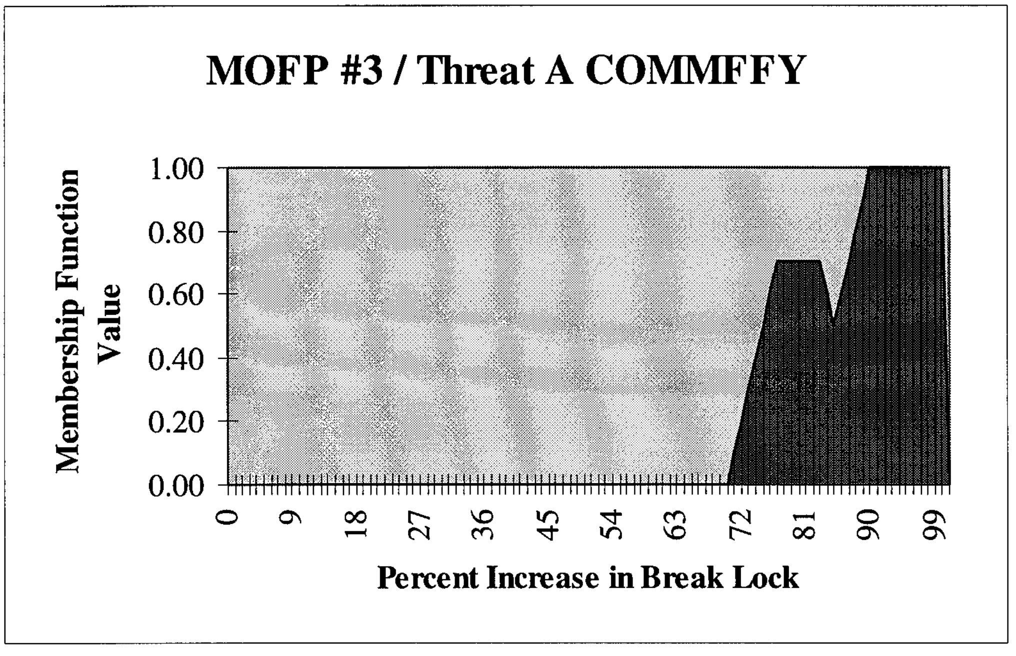

Figure F-3 COMMFFY Indicating MOFP #3 (Percent Increase in Break Locks) Performance Against Threat A

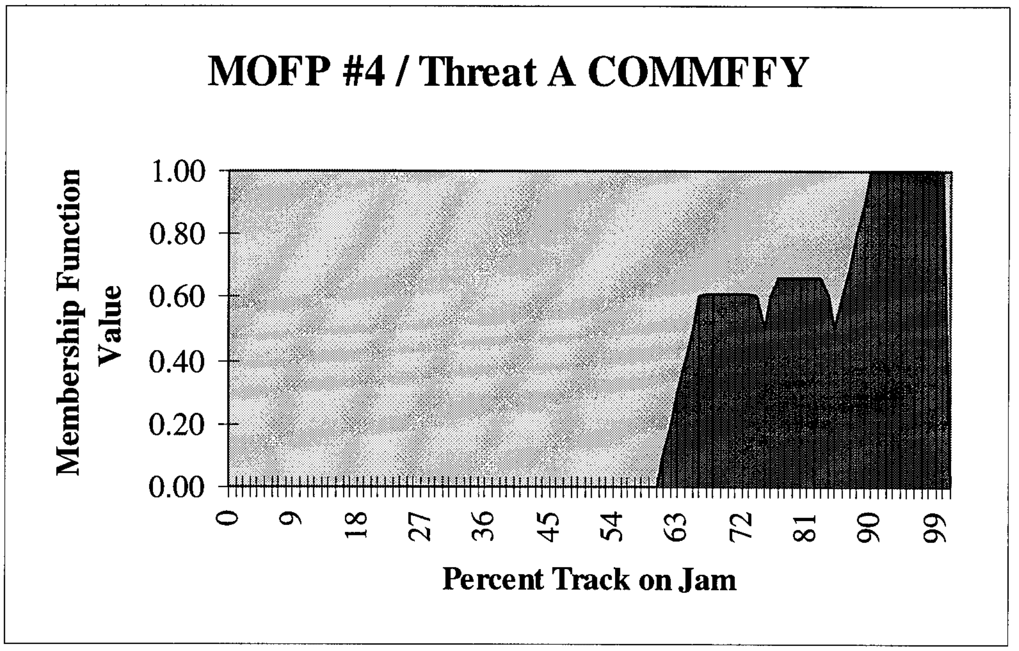

Figure F-4 COMMFFY Indicating MOFP #4 (Percent Track on Jam) Performance Against Threat A

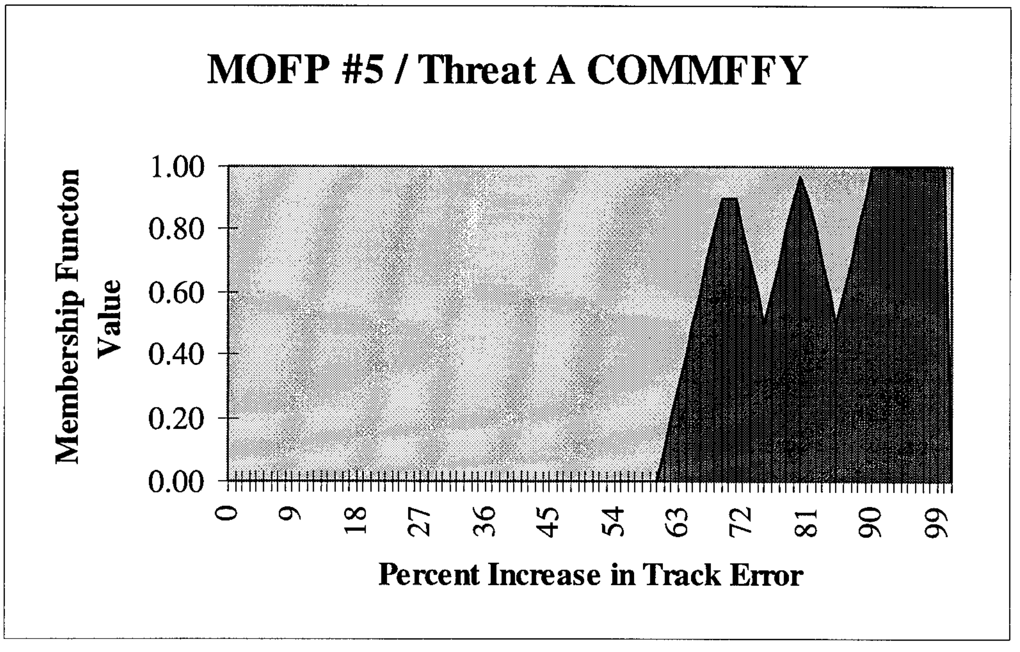

Figure F-5 COMMFFY Indicating MOFP #5 (Percent Increase in Track Error) Performance Against Threat A

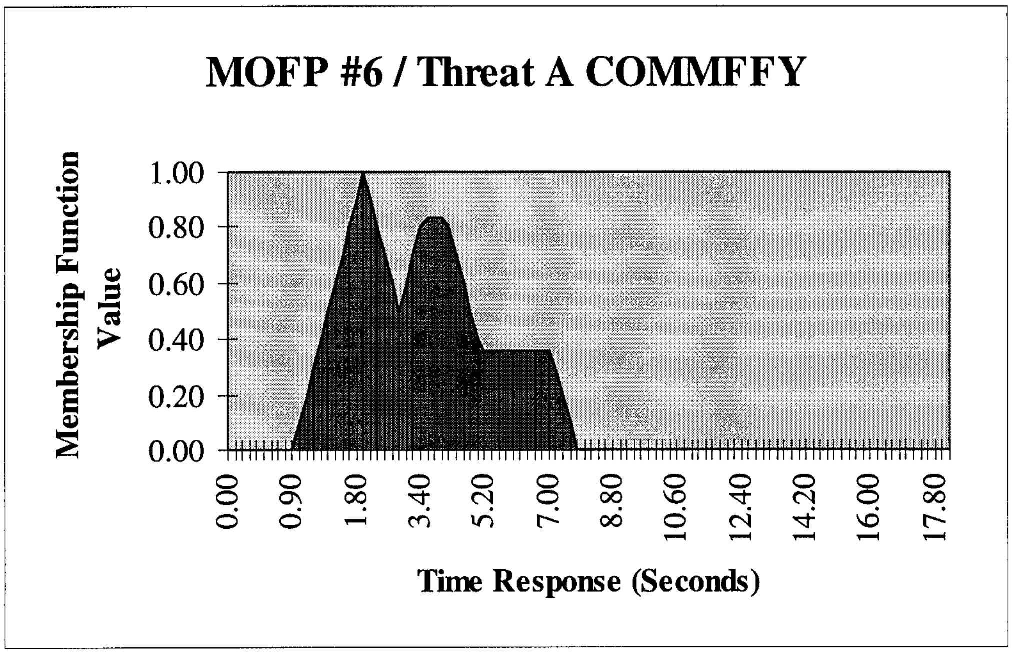

Figure F-6 COMMFFY Indicating MOFP #6 (Jammer Response Time) Performance Against Threat A

F.2.2 THREAT B COMPOSITE FUZZY MEMBERSHIP FUNCTIONS

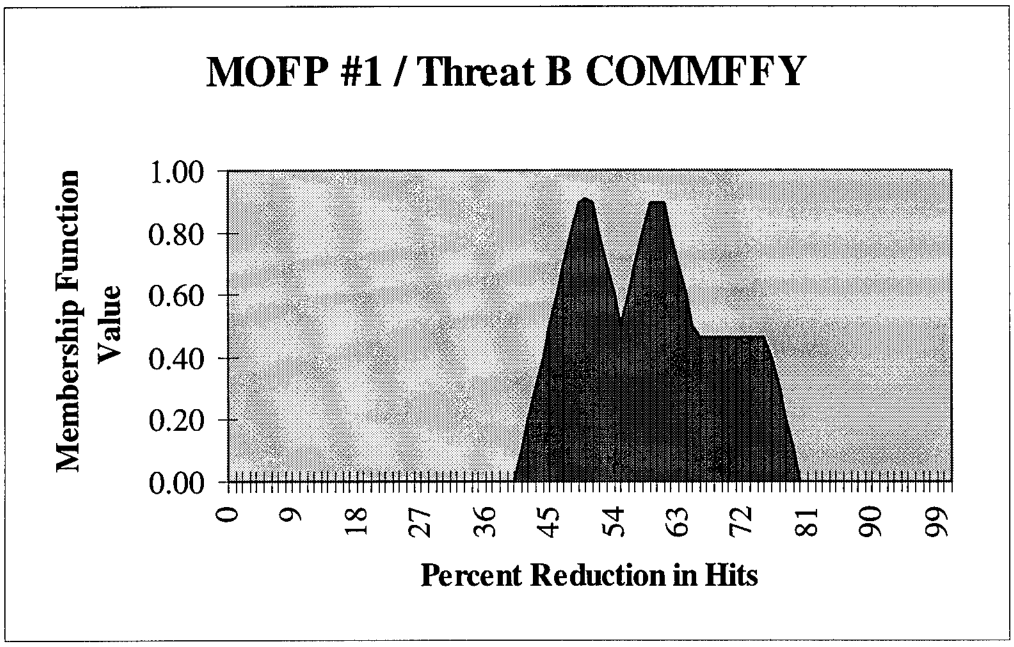

Figure F-7 COMMFFY Indicating MOFP #1 (Percent Reduction in Hits) Performance Against Threat B

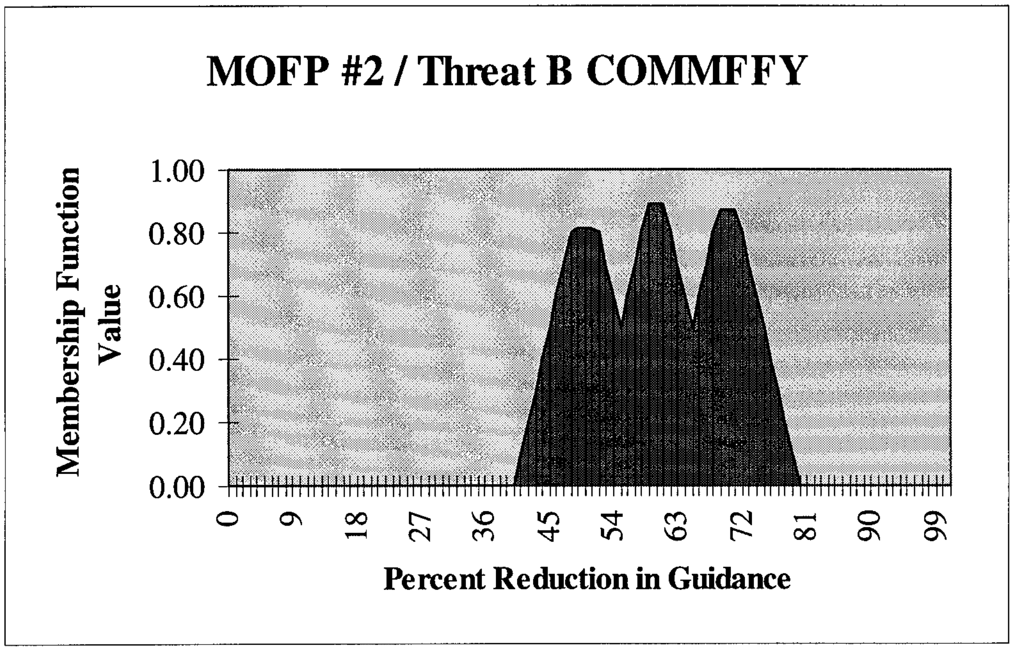

Figure F-8 COMMFFY Indicating MOFP #2 (Percent Reduction in Guidance) Performance Against Threat B

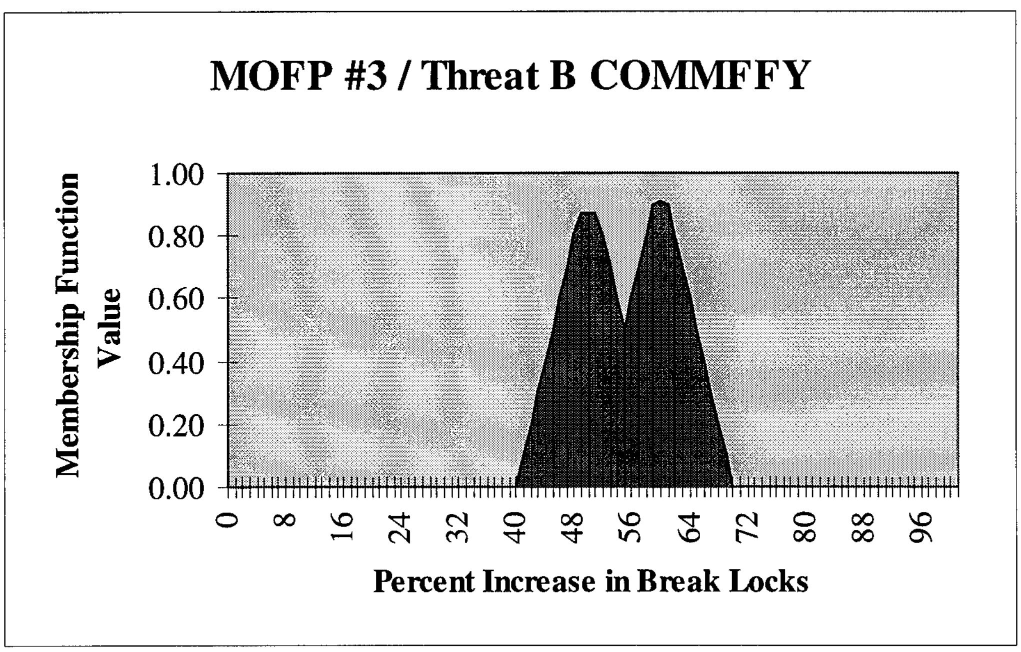

Figure F-9 COMMFFY Indicating MOFP #3 (Percent Increase in Break Locks) Performance Against Threat B

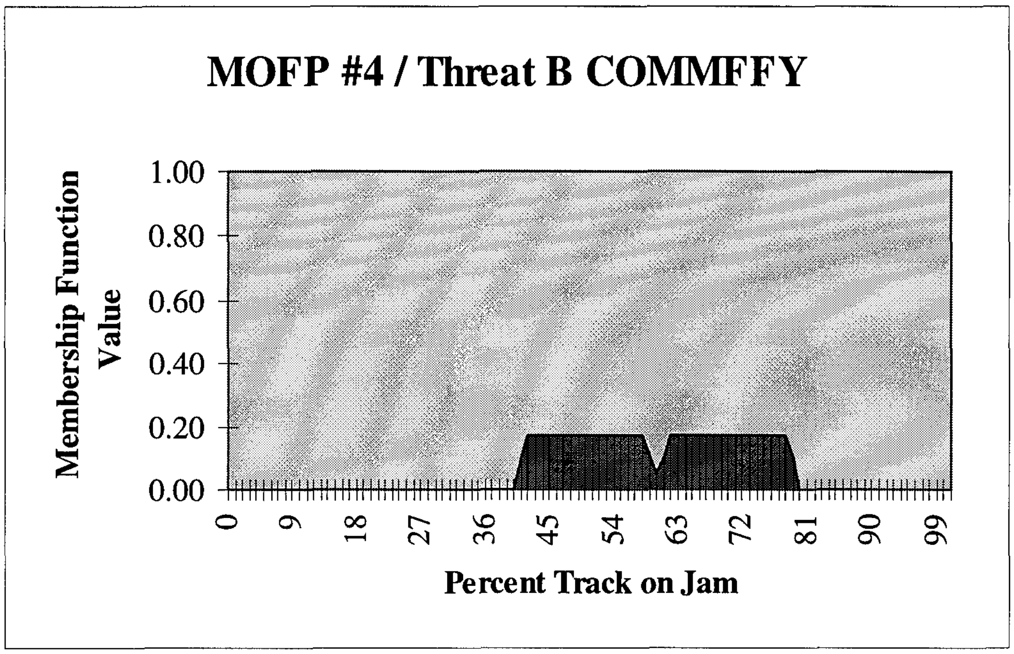

Figure F-10 COMMFFY Indicating MOFP #4 (Percent Track on Jam) Performance Against Threat B

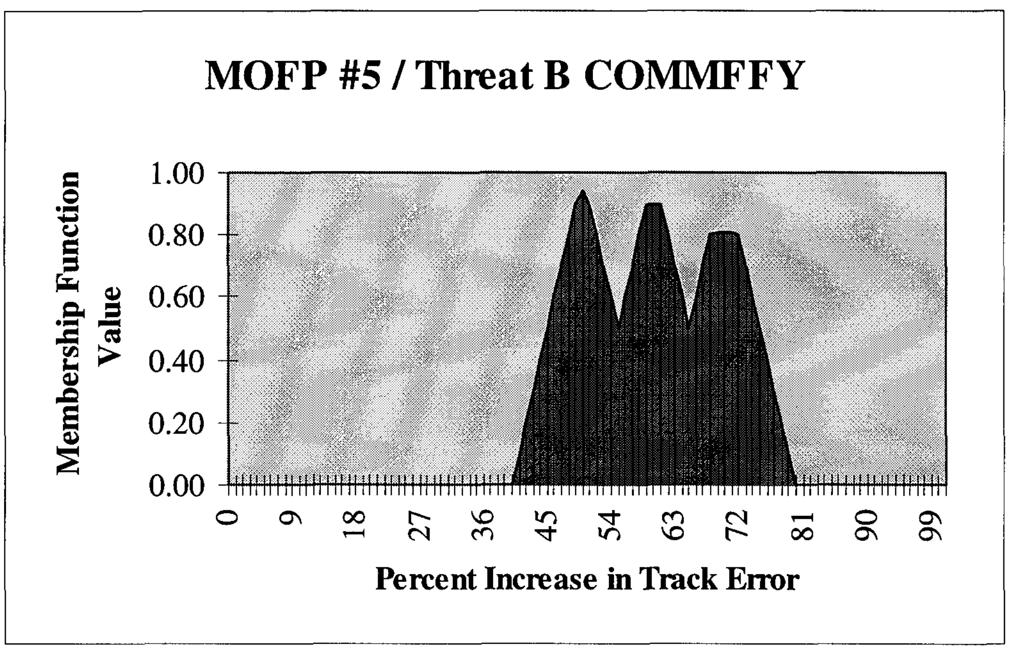

Figure F-11 COMMFFY Indicating MOFP #5 (Percent Increase in Track Error) Performance Against Threat B

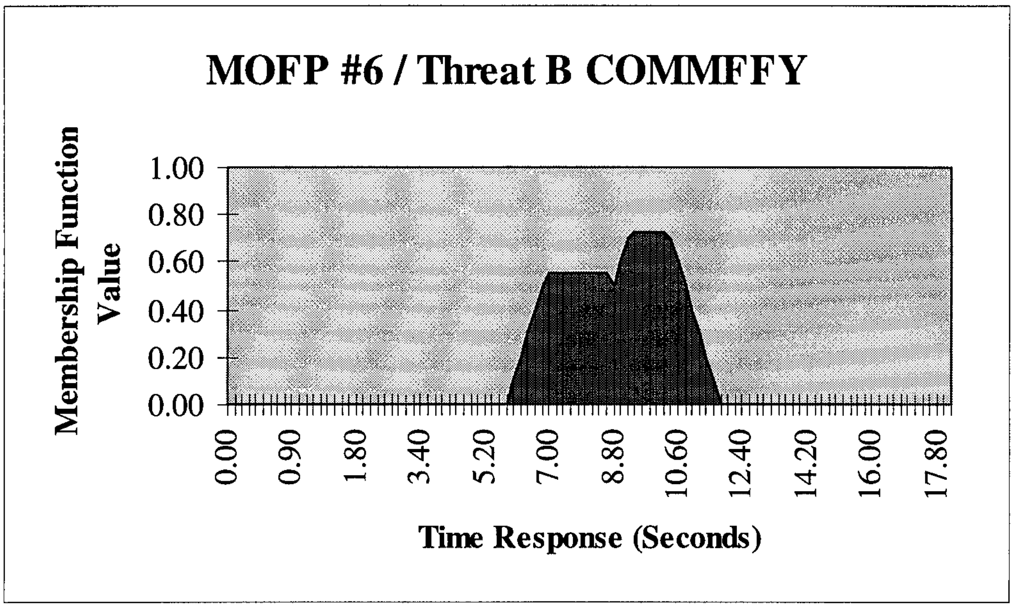

Figure F-12 COMMFFY Indicating MOFP #6 (Jammer Response Time) Performance Against Threat B

F.2.3 THREAT C COMPOSITE FUZZY MEMBERSHIP FUNCTIONS

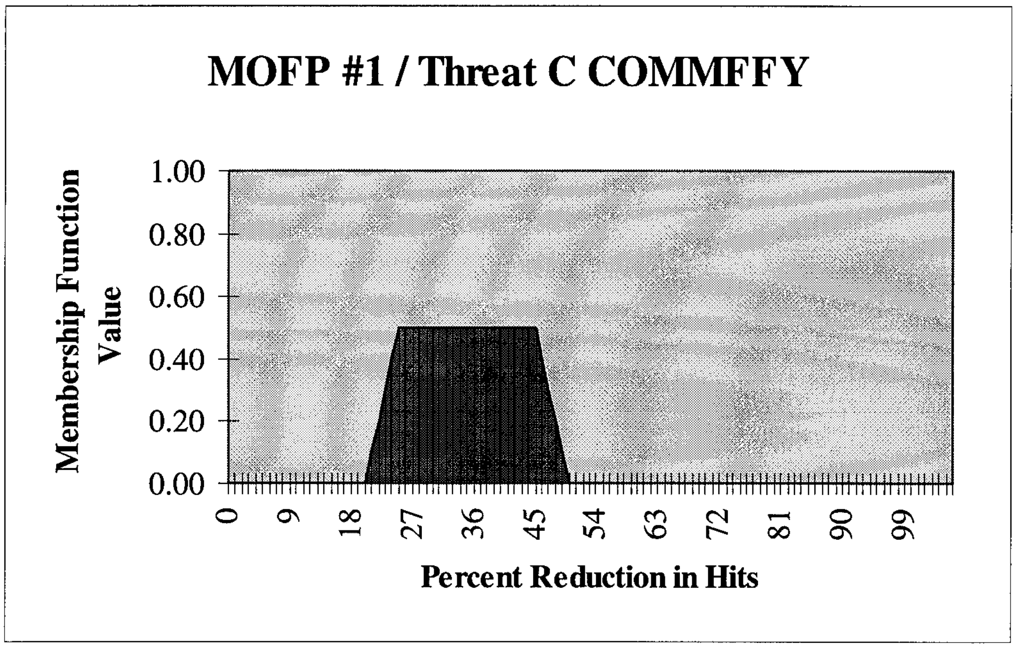

Figure F-1 3 COMMFFY Indicating MOFP #1 (Percent Reduction in Hits) Performance Against Threat C

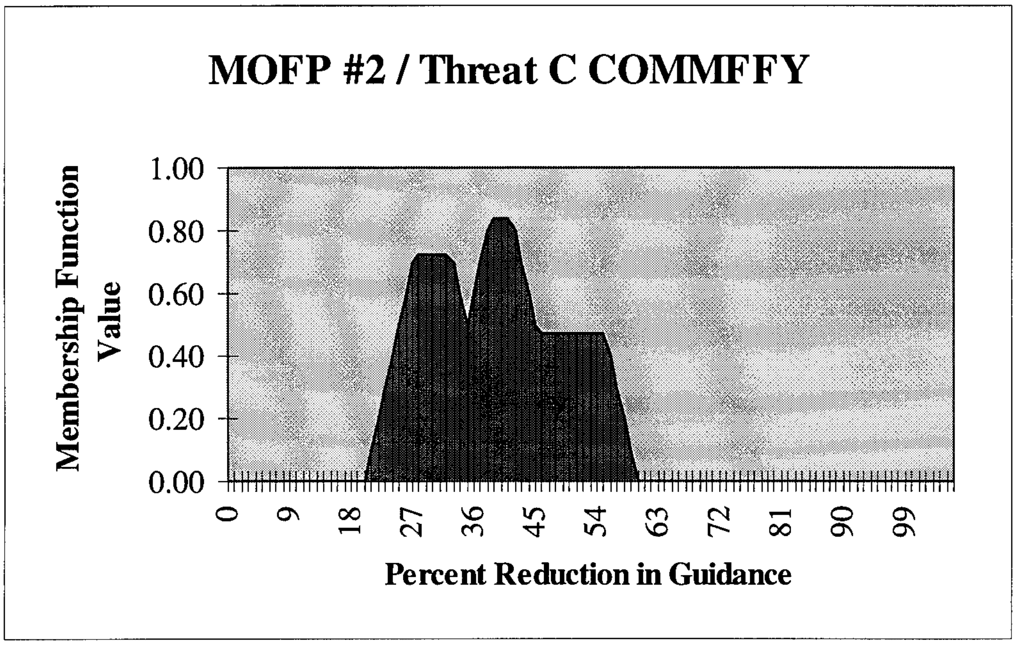

Figure F-14 COMMFFY Indicating MOFP #2 (Percent Reduction in Guidance) Performance Against Threat B

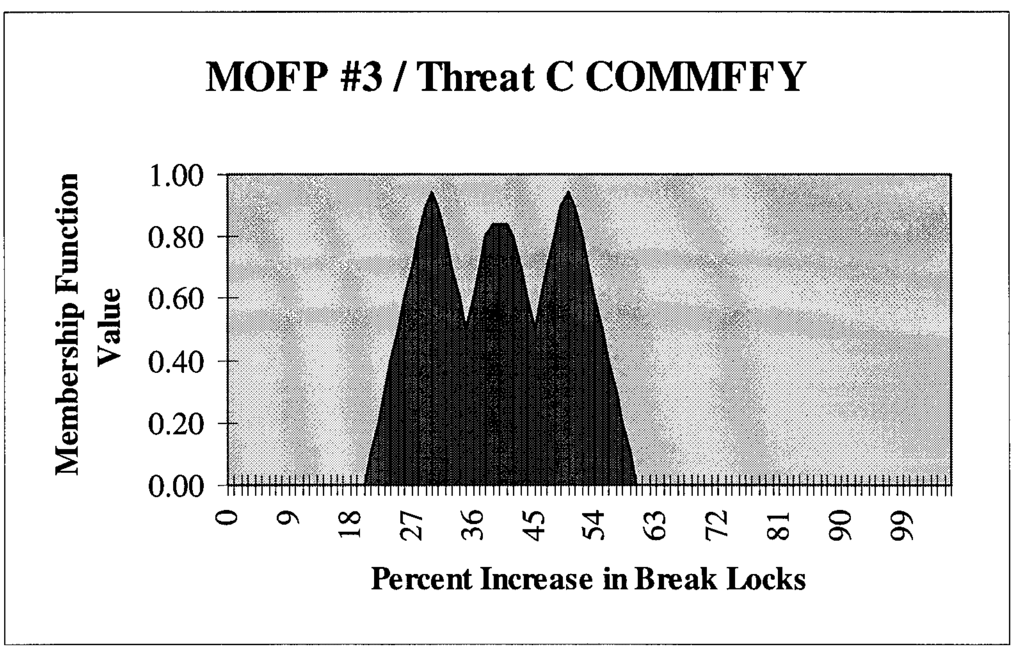

Figure F-15 COMMFFY Indicating MOFP #3 (Percent Increase in Break Locks) Performance Against Threat C

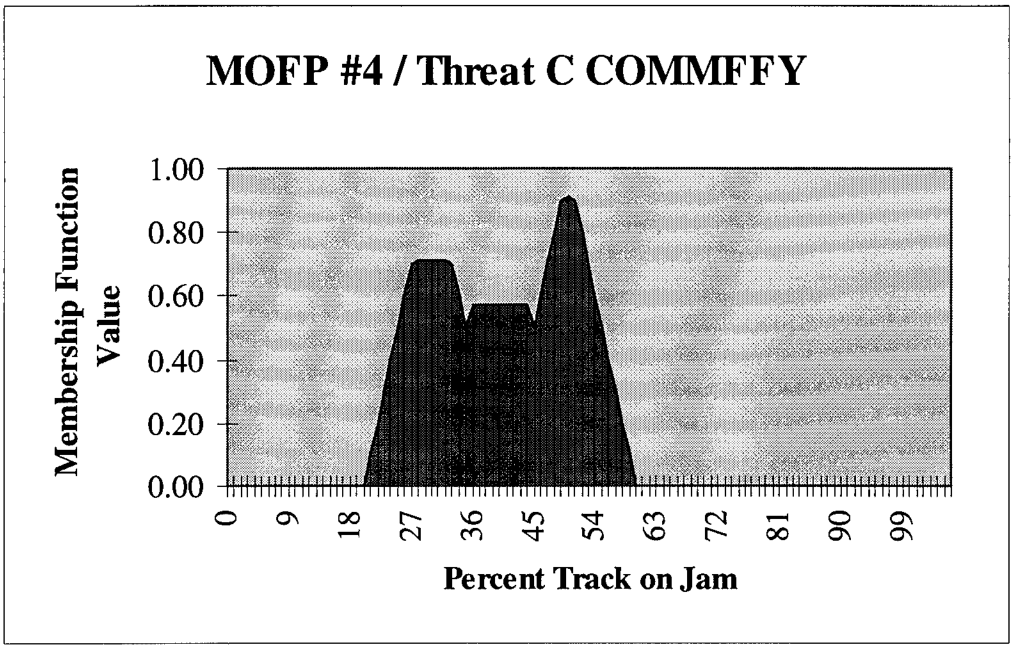

Figure F-16 COMMFFY Indicating MOFP #4 (Percent Track on Jam) Performance Against Threat C

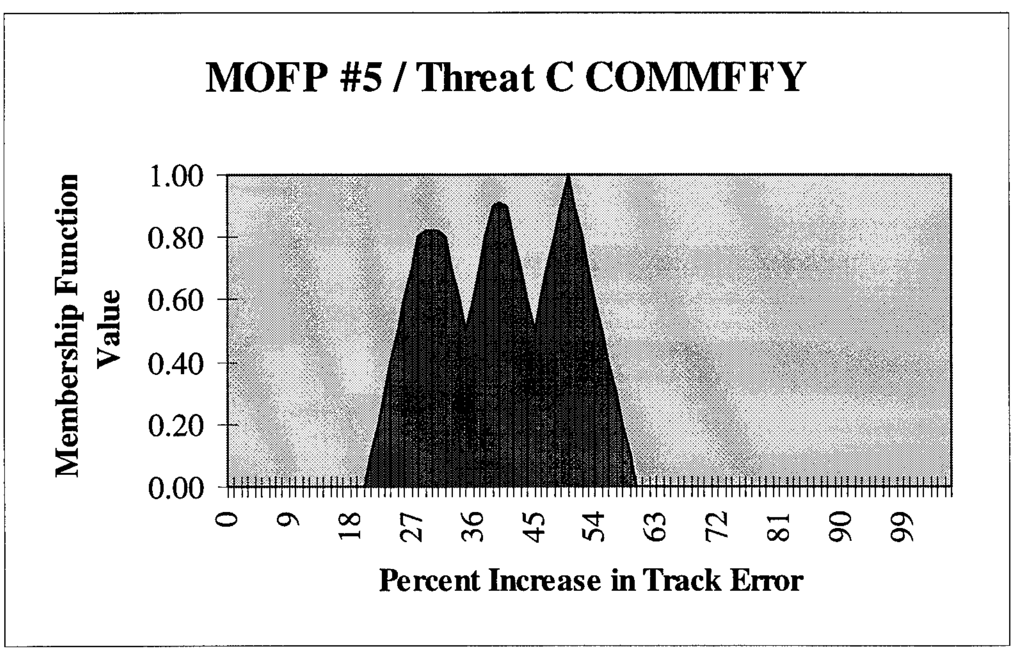

Figure F-17 COMMFFY Indicating MOFP #5 (Percent Increase in Track Error) Performance Against Threat C

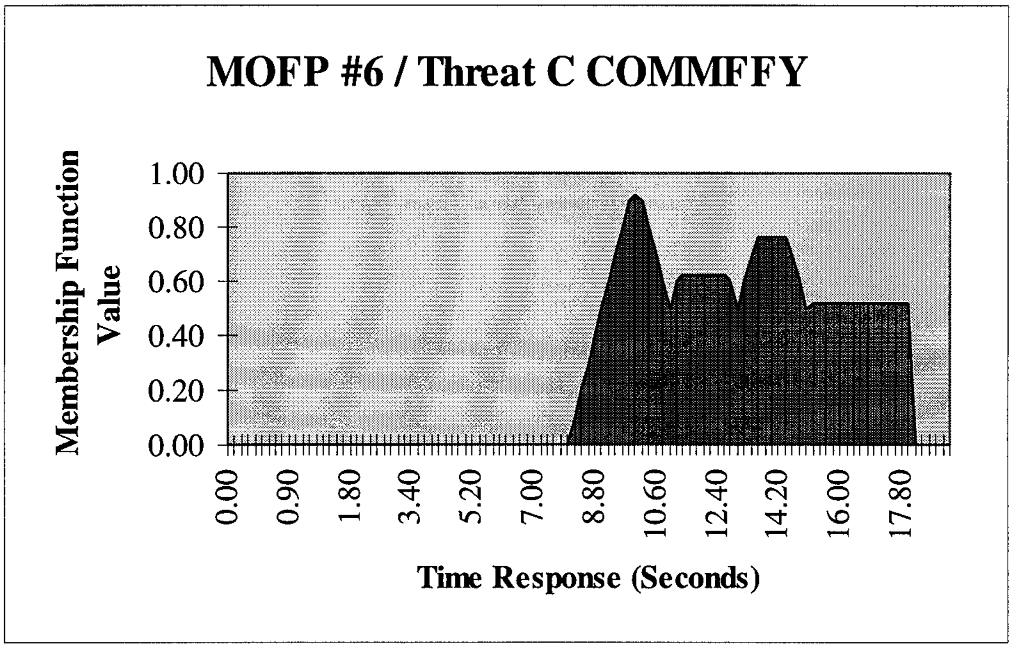

Figure F-18 COMMFFY Indicating MOFP #6 (Jammer Response Time) Performance Against Threat C

F.2.4 THREAT D COMPOSITE FUZZY MEMBERSHIP FUNCTIONS

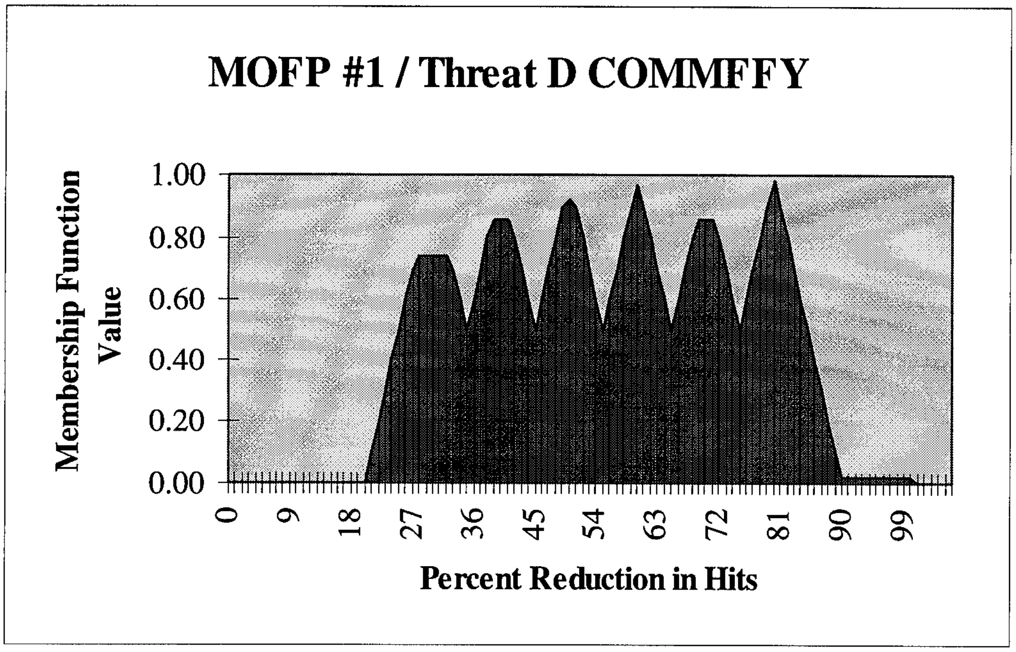

Figure F-19 COMMFFY Indicating MOFP #1 (Percent Reduction in Hits) Performance Against Threat D

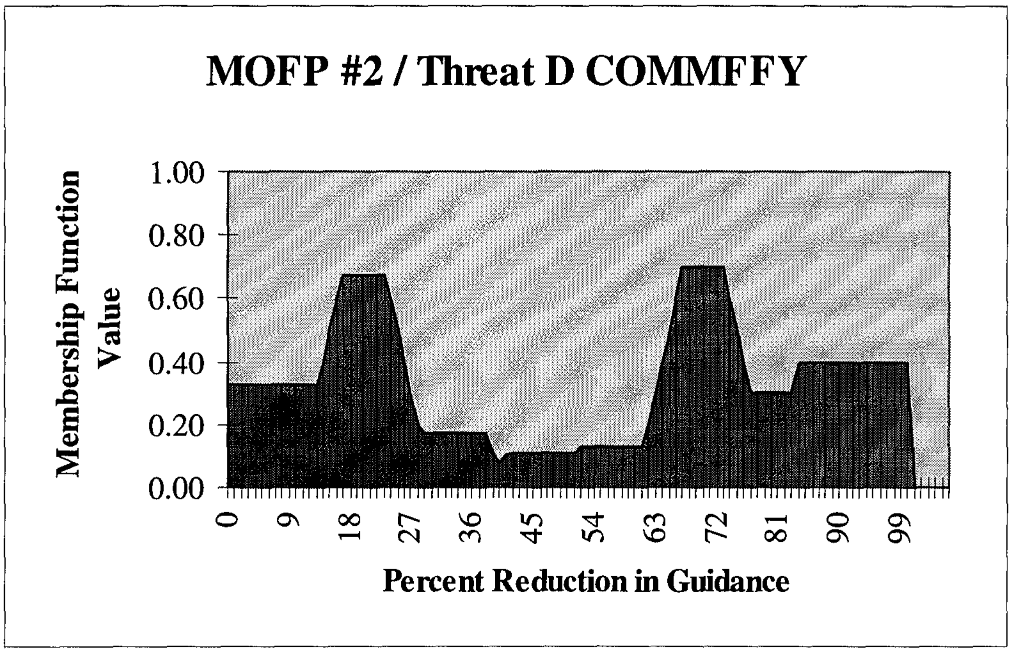

Figure F-20 COMMFFY Indicating MOFP #2 (Percent Reduction in Guidance) Performance Against Threat D

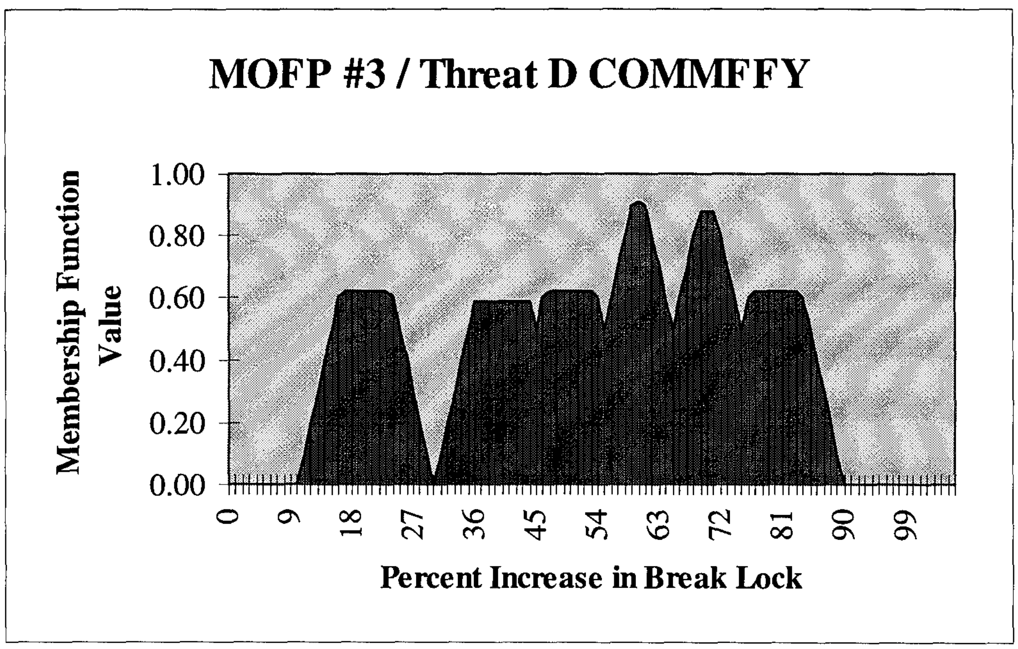

Figure F-21 COMMFFY Indicating MOFP #3 (Percent Increase in Break Locks) Performance Against Threat D

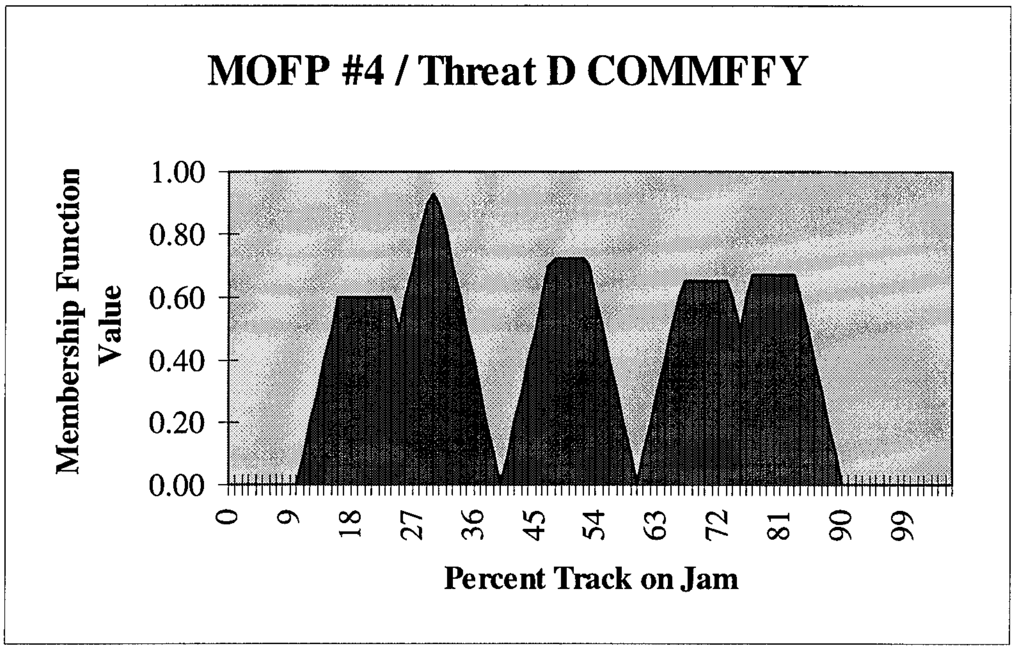

Figure F-22 COMMFFY Indicating MOFP #4 (Percent Track on Jam) Performance Against Threat D

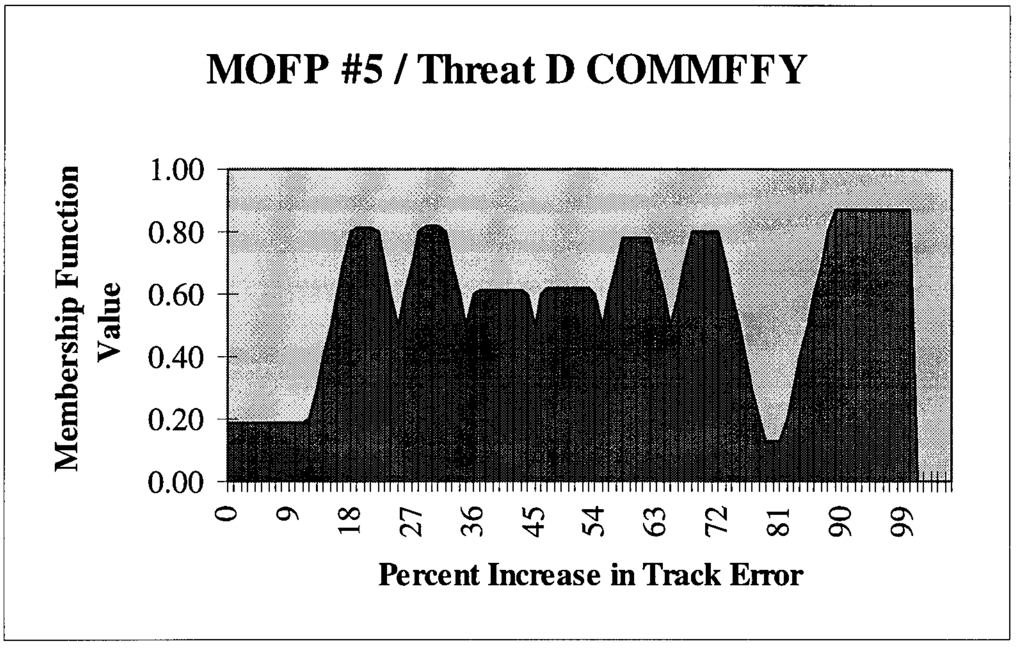

Figure F-23 COMMFFY Indicating MOFP #5 (Percent Increase in Track Error) Performance Against Threat D

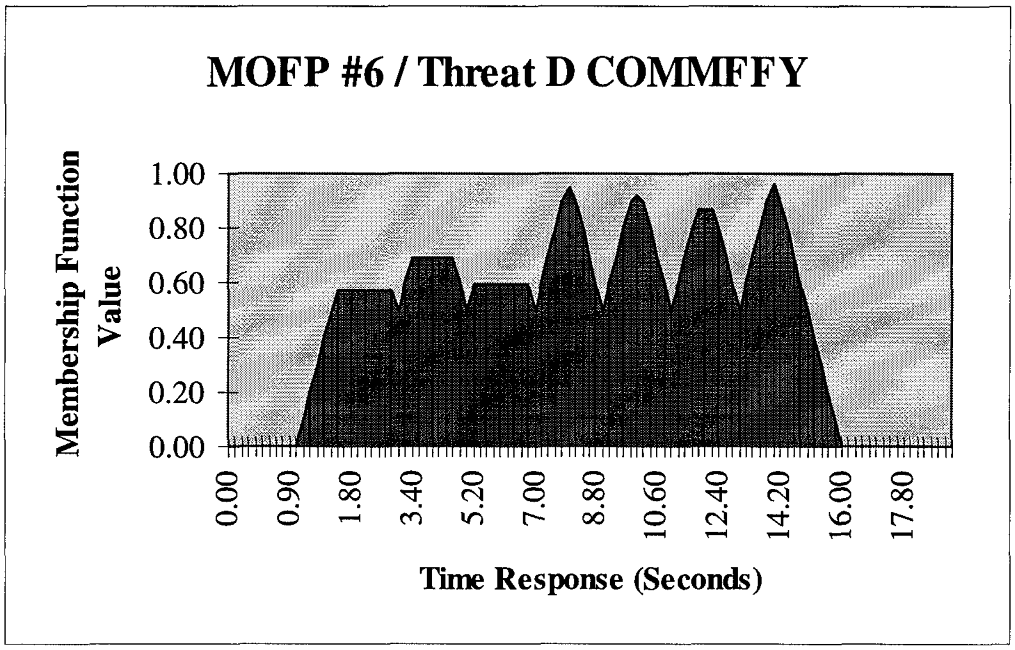

Figure F-24 COMMFFY Indicating MOFP #6 (Jammer Response Time) Performance Against Threat D

F.3 FUZZY ASSOCIATIVE MEMORY

The Fuzzy Associative Memory within the *Intelligent Hierarchical Decision Architecture* serves to transform the Composite Fuzzy Membership Functions at the functional-performance level to Composite Fuzzy Membership Functions at the task-accomplishment level. This transformation is accomplished using a rule bank with the rules written at the Basic Membership Function level (i.e., IF Reduction in Hits is *low* THEN Reduction in Probability of Kill is *low*). For the testbed case, the Fuzzy Associative Memory takes the functional-performance level COMMFFYs for each Threat and transforms them to a COMMFFY indicating the system's performance at the task-accomplishment level against each threat system. The *Measure of Task Accomplishment* used in the testbed case is Percent Reduction in Probability of Kill. Figure through Figure illustrate the four task-level COMMFFYs (one for each threat system) that are the output of the Fuzzy Associative Memory.

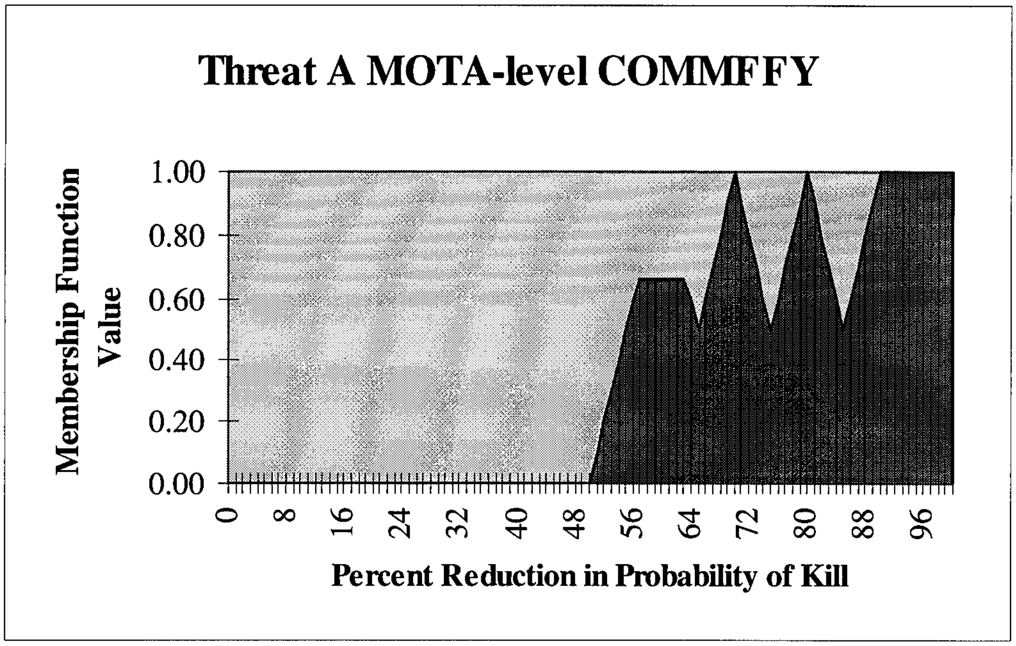

Figure F-25 Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat A

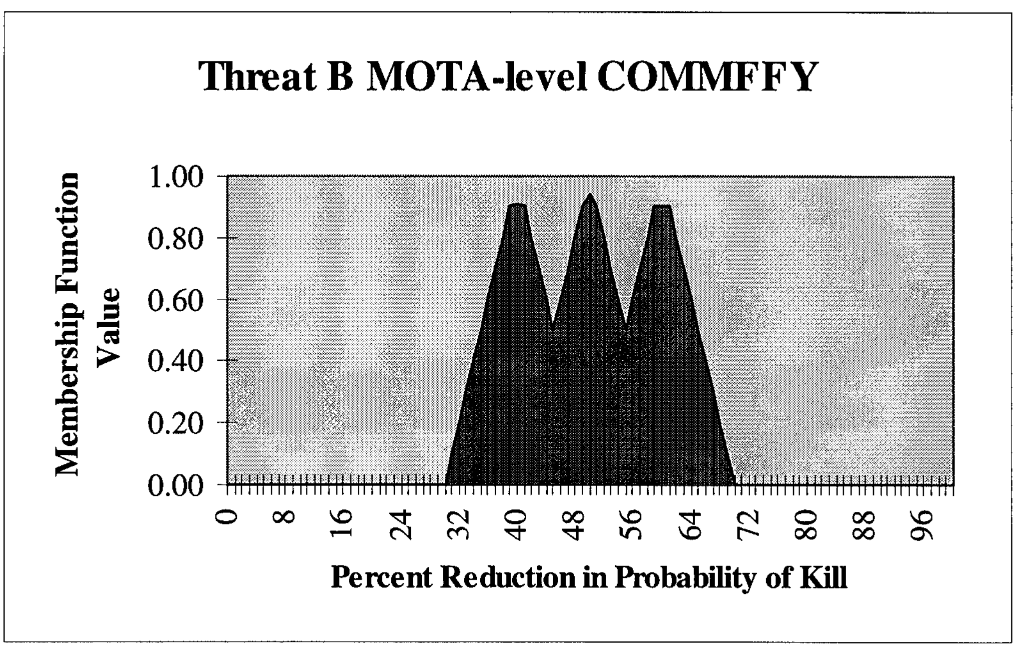

Figure F-26 Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat B

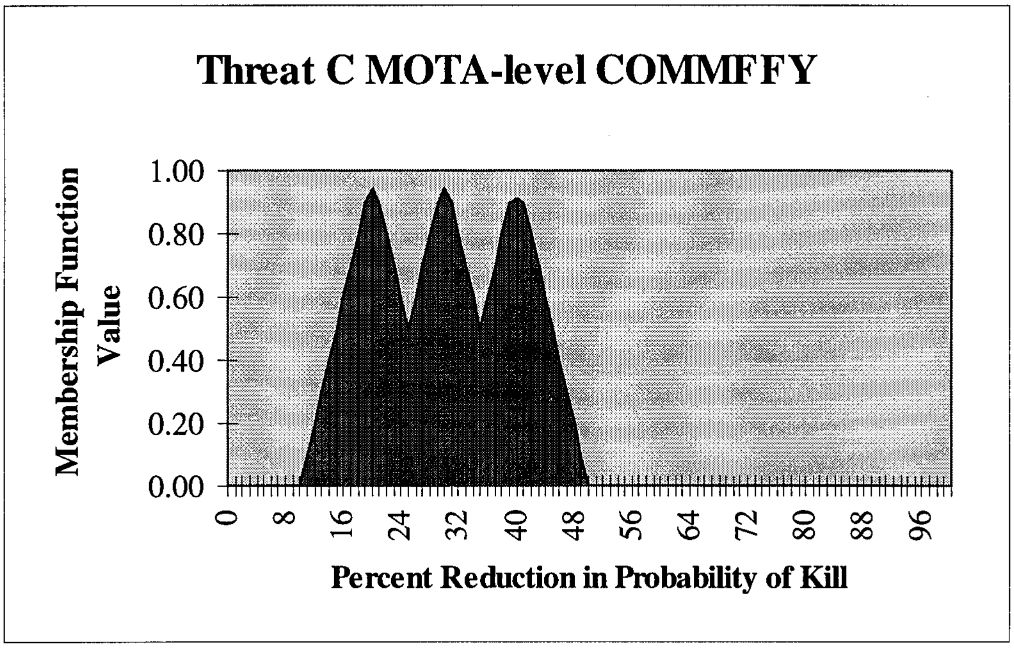

Figure F-27 Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat C

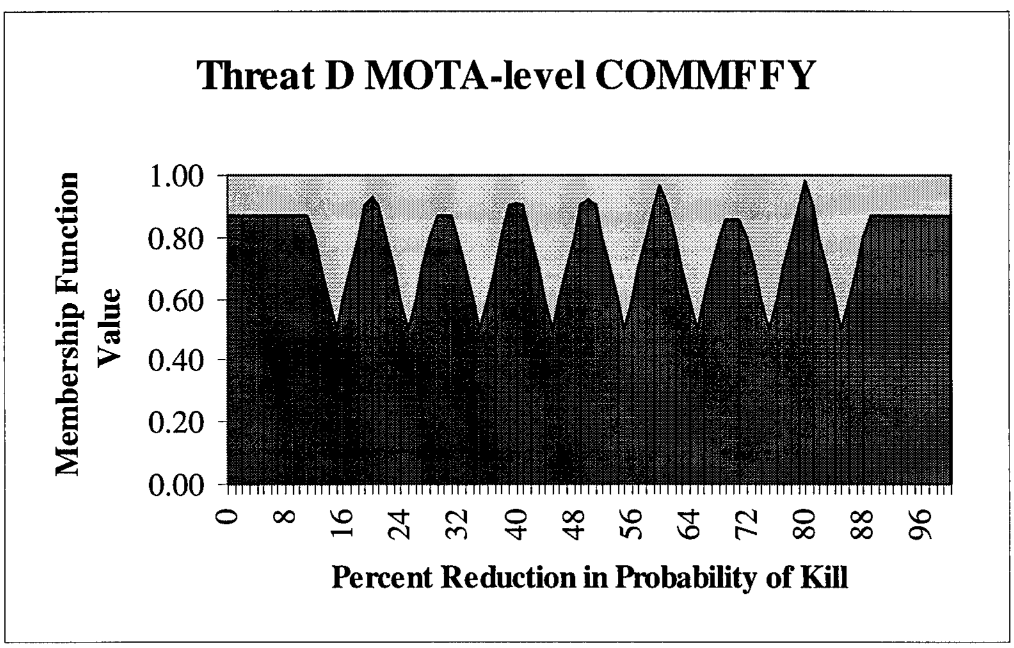

Figure F-28 Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat D

B.4 FUZZY COGNITIVE MAP

The Fuzzy Cognitive Map within the *Intelligent Hierarchical Decision Architecture* adjusts the task-accomplishment level performance determined at the output of the Fuzzy Associative Memory for factors that could not be controlled or included in the testing effort, yet are known to have an affect on the system performance measure. In the testbed case, the Fuzzy Cognitive Map analysis yielded a +*Very* best-case adjustment and a -*Very* worst-case adjustment due to those uncontrollable and untestable factors. These adjustments are applied to the MOTA-level COMMFFYs shown in Figure F-25 through Figure F-28, giving the eight MOTA-level Adjusted COMMFFYs (best-case and worst-case for each threat system) shown in Figure F-29 through Figure F-36.

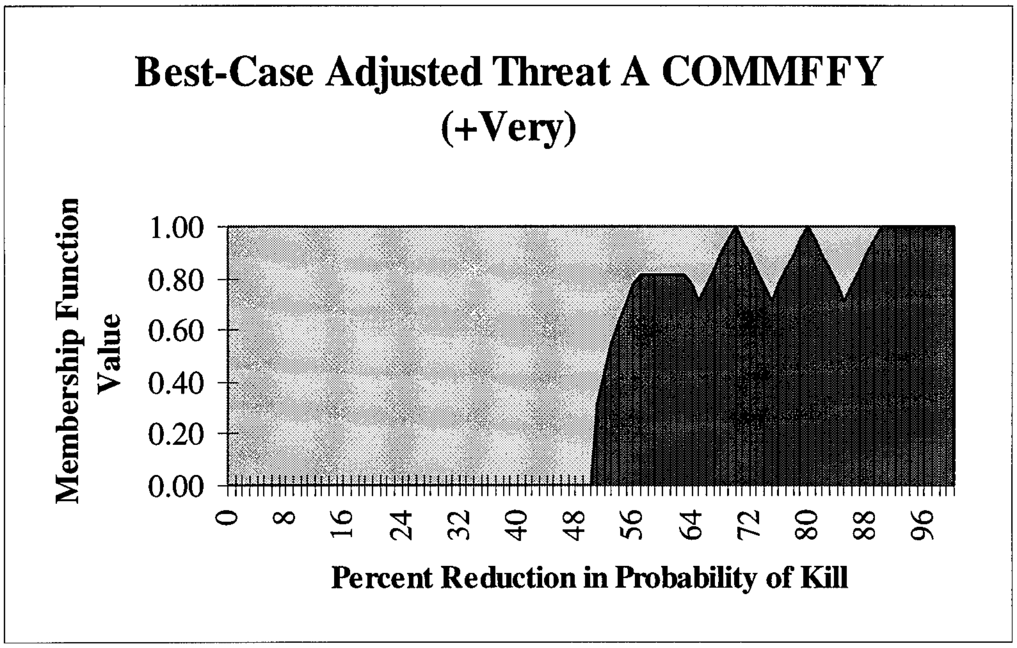

Figure F-29 Best-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat A

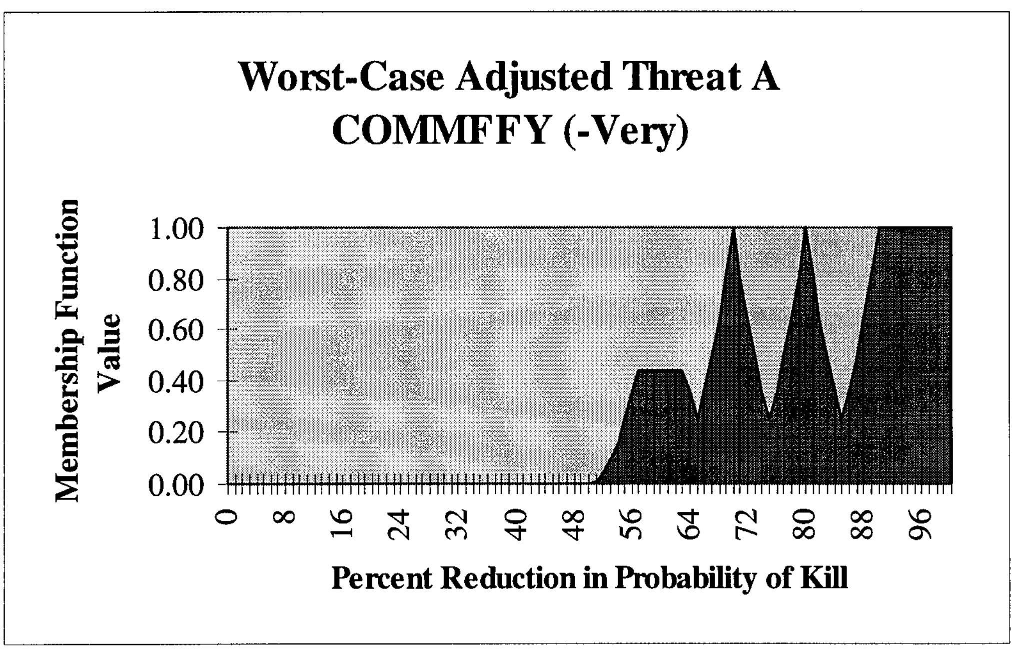

Figure F-30 Worst-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat A

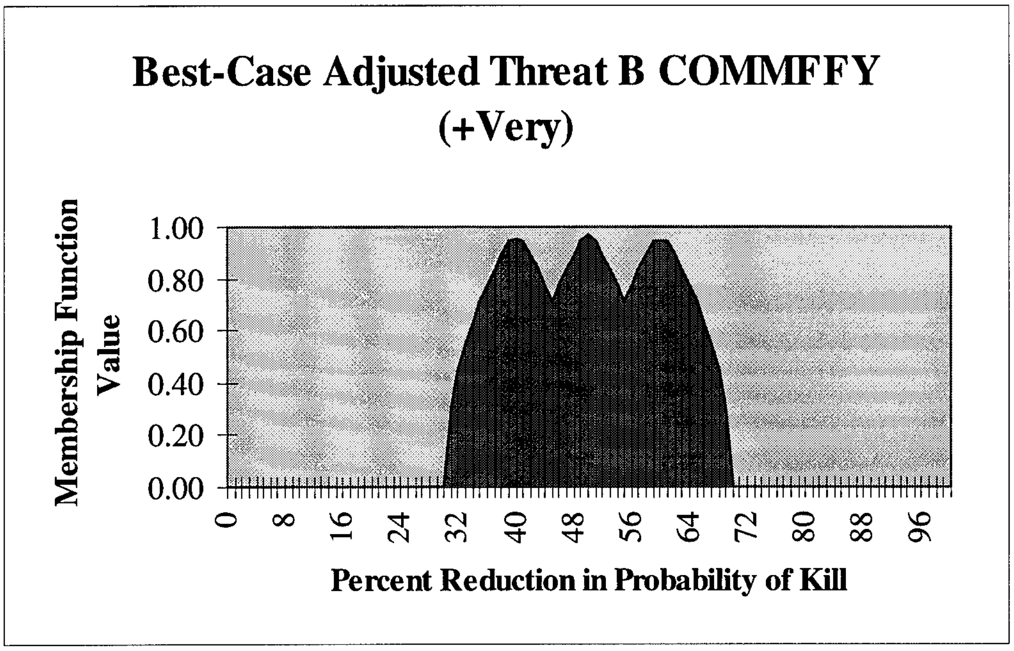

Figure F-31 Best-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat B

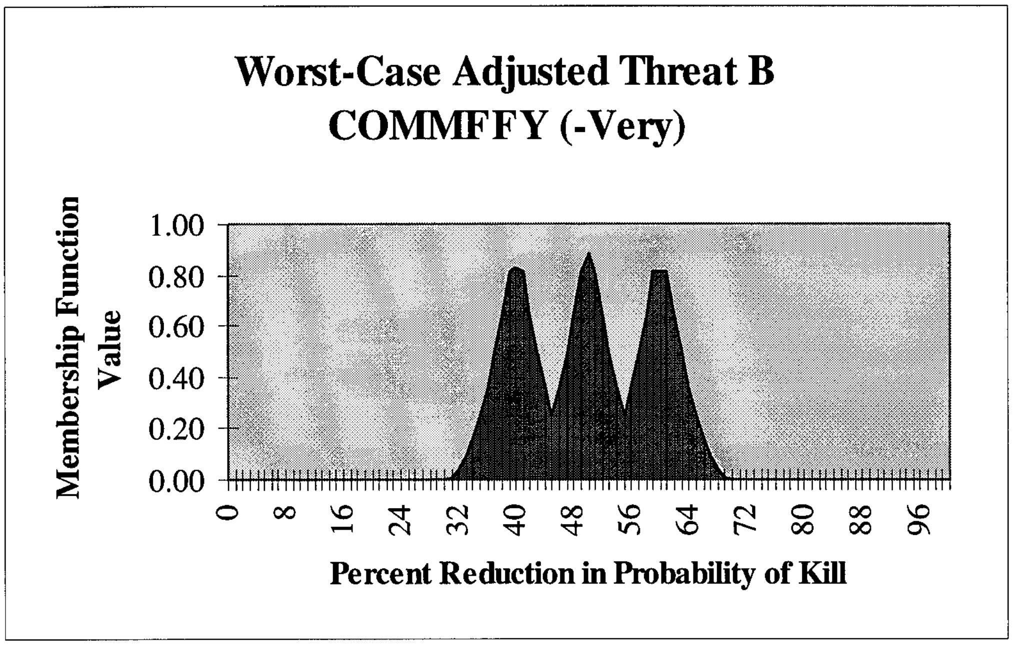

Figure F-32 Worst-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat B

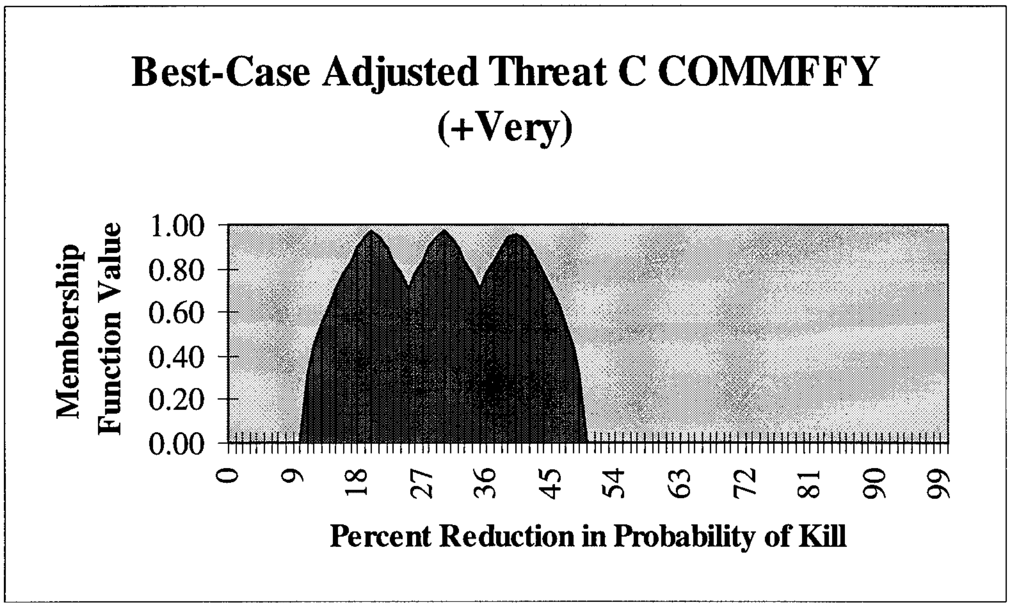

Figure F-33 Best-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat C

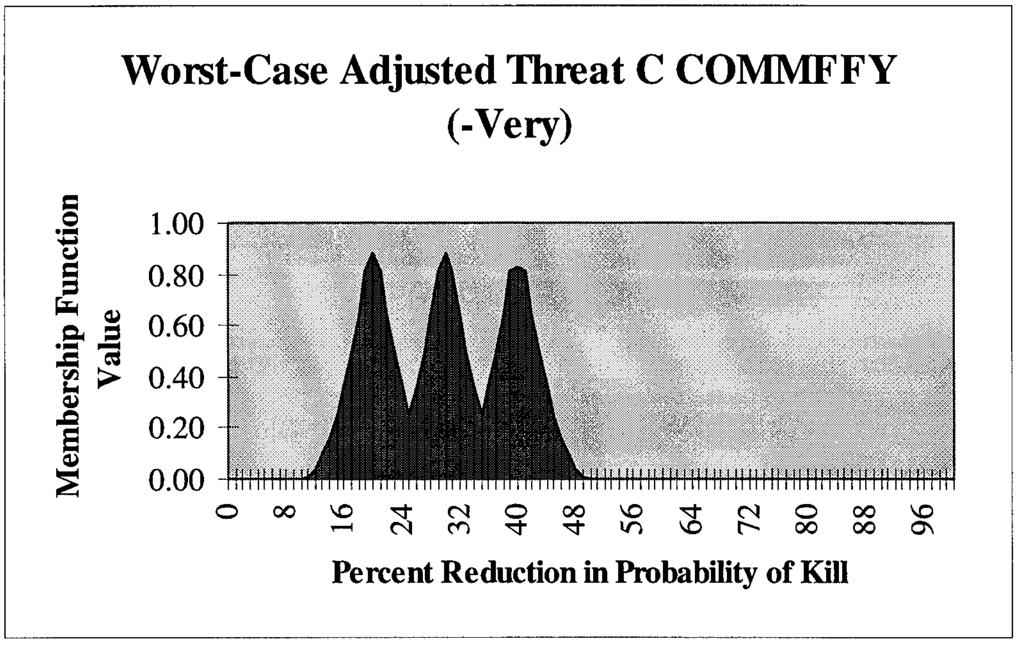

Figure F-34 Worst-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat C

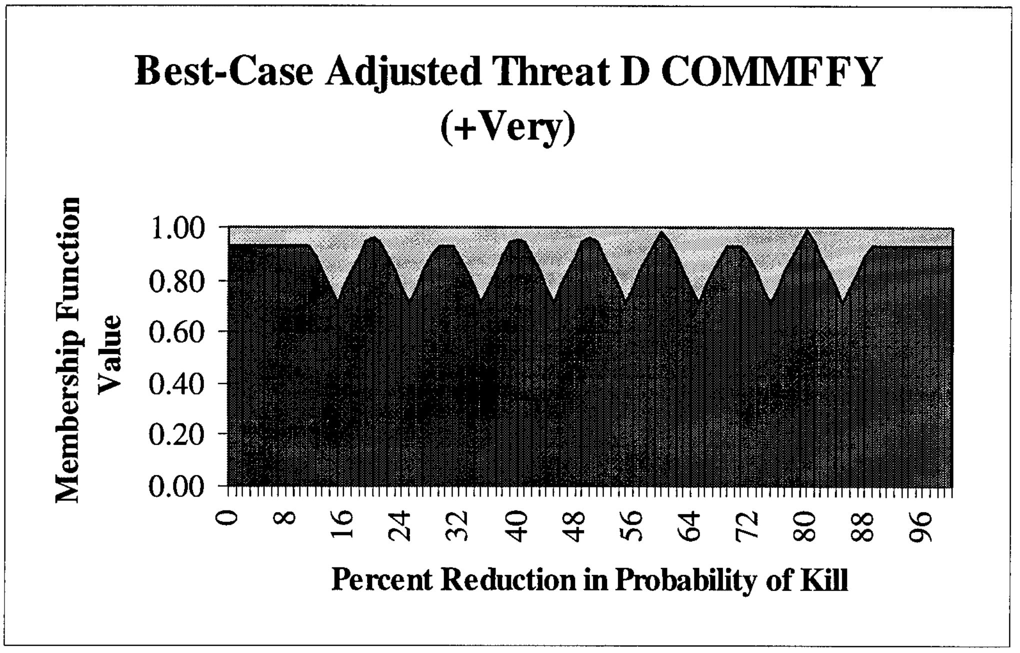

Figure F-35 Best-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat D

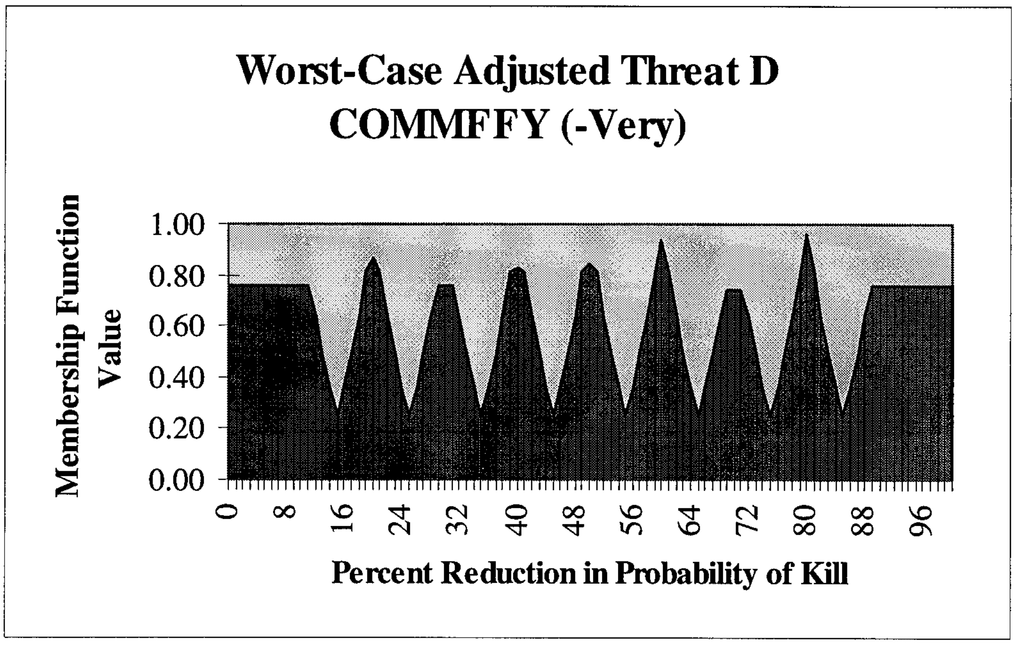

Figure F-36 Worst-Case Adjusted Task-Level COMMFFY Indicating Percent Reduction in Probability of Kill Performance Against Threat D

F.5 AGGREGATION METHODOLOGY

The final stage in the Intelligent Hierarchical Analysis Methodology consists of aggregating the information contained in the adjusted, task-level Composite Fuzzy Membership Functions for each division of the system into a single system performance bound. This aggregation is done by applying the Dempster's Rule of Combination separately to the Best-Case and Worst-Case COMMFFYs. With this combination of information, the final result is a best-case probability bound on the system performance and a worst-case probability bound on the system performance. The calculations of the aggregated probability bound are shown in Table F-7 and Table F-9 through Table F-11 for the Best-Case performance; Table F-8 and Table F-13 through Table F-15 for the Worst-Case performance. The results that would be reported to the decision-maker for the Best-Case and Worst-Case performance are given in Table F-12 and Table F-16, respectively.

The first step in the Aggregation Methodology is to take the COMMFFYs derived from the first three stages of the *Intelligent Hierarchical Decision Architecture* and transform them back to a single membership function value at the Basic Membership Function level. This will serve to make the use of the Dempster's Rule of Combination more manageable and will make the final result more meaningful (i.e., final statements about the system performance being *Medium*, or *Low*, etc.). The Basic Membership Function values for the best-case performance and the worst-case performance separately, are shown below.

F.5.1 BASIC MEMBERSHIP FUNCTION DEGREE OF MEMBERSHIP VALUES

Table F-7 Best Case Task-Level Performance at the Basic Membership Function Level

BMF Tag	BMF #	Threat A	Threat B	Threat C	Threat D
LO	0	0.00	0.00	0.00	0.93
LOMEDLO	1	0.00	0.00	0.97	0.96
MEDLO	2	0.00	0.00	0.97	0.93
HIMEDLO	3	0.00	0.95	0.95	0.95
MED	4	0.00	0.97	0.00	0.96
LOMEDHI	5	0.81	0.95	0.00	0.98
MEDHI	6	1.00	0.00	0.00	0.93
HIMEDHI	7	1.00	0.00	0.00	0.99
HI	8	1.00	0.00	0.00	0.93

Table F-8 Worst Case Task-Level Performance at the Basic Membership Function Level

BMF Tag	BMF #	Threat A	Threat B	Threat C	Threat D
LO	0	0.00	0.00	0.00	0.76
LOMEDLO	1	0.00	0.00	0.88	0.86
MEDLO	2	0.00	0.00	0.85	0.76
HIMEDLO	3	0.00	0.83	0.83	0.83
MED	4	0.00	0.88	0.00	0.85
LOMEDHI	5	0.44	0.81	0.00	0.94
MEDHI	6	1.00	0.00	0.00	0.74
HIMEDHI	7	1.00	0.00	0.00	0.96
HI	8	1.00	0.00	0.00	0.76

F.5.2 ASSIGNMENT OF BASIC PROBABILITY ASSIGNMENT BY DECOMPOSITION INTO ALPHA-LEVEL SETS

The next two steps in the Aggregation Methodology are to decompose the fuzzy sets shown above into their α-cut sets and assign the basic probability assignment to each. The result of these two steps are shown below, first for the best-case sets followed by the worst-case sets.

Best-Case:

$$m(A_{0.81}) = m\{5,6,7,8\} = 0.81$$
$$m(A_{1.00}) = m\{6,7,8\} = 0.19$$
$$m(A_{\Theta}) = m\{\Theta\} = 0.00$$

$$m(B_{0.95}) = m\{3,4,5\} = 0.95$$
$$m(B_{0.97}) = m\{4\} = 0.02$$
$$m(B_{\Theta}) = m\{\Theta\} = 0.03$$

$$m(C_{0.95}) = m\{1,2,3\} = 0.95$$
$$m(C_{0.97}) = m\{1,2\} = 0.02$$
$$m(C_{\Theta}) = m\{\Theta\} = 0.03$$

$$m(D_{0.93}) = m\{0,1,2,3,4,5,6,7,8\} = 0.93$$
$$m(D_{0.95}) = m\{1,3,4,5,7\} = 0.02$$
$$m(D_{0.96}) = m\{1,4,5,7\} = 0.01$$
$$m(D_{0.98}) = m\{5,7\} = 0.02$$
$$m(D_{0.99}) = m\{7\} = 0.01$$
$$m(D_{\Theta}) = m\{\Theta\} = 0.01$$

Worst-Case:

$m(A_{0.44}) = m\{5,6,7,8\} = 0.44$

$m(A_{1.00}) = m\{6,7,8\} = 0.56$

$m(A_{\Theta}) = m\{\Theta\} = 0.00$

$m(B_{0.81}) = m\{3,4,5\} = 0.81$

$m(B_{0.83}) = m\{3,4\} = 0.02$

$m(B_{0.88}) = m\{4\} = 0.05$

$m(B_{\Theta}) = m\{\Theta\} = 0.12$

$m(C_{0.83}) = m\{1,2,3\} = 0.83$

$m(C_{0.85}) = m\{1,2\} = 0.02$

$m(C_{0.88}) = m\{1\} = 0.03$

$m(C_{\Theta}) = m\{\Theta\} = 0.12$

$m(D_{0.74}) = m\{0,1,2,3,4,5,6,7,8\} = 0.74$

$m(D_{0.76}) = m\{0,1,2,3,4,5,7,8\} = 0.02$

$m(D_{0.83}) = m\{1,3,4,5,7\} = 0.07$

$m(D_{0.85}) = m\{1,4,5,7\} = 0.02$

$m(D_{0.86}) = m\{1,5,7\} = 0.01$

$m(D_{0.94}) = m\{5,7\} = 0.08$

$m(D_{0.96}) = m\{7\} = 0.02$

$m(D_{\Theta}) = m\{\Theta\} = 0.04$

F.5.3 Combination of Information Using Dempster's Rule of Combination

After the basic probability assignments have been established, the combination of the information using the intersection tableau method of Dempster's Rule of Combination is performed. Shown below are the tableaus resulting from first combining Threat A with Threat B, then combining Threat C with the combined Threat A-B, and finally combining Threat D with the combined Threat A-B-C evidence.

Table F-9 Intersection Tableau Combining Best-Case Threat A Performance with Best-Case Threat B Performance

Thr. B / Thr A	*m{5,6,7,8} = 0.95*	*m{6,7,8} = 0.19*	*m{Θ} = 0.00*
m{3,4,5} = 0.95	{5} = 0.7695	{∅} = 0.1805	{3,4,5} = 0.0000
m{4} = 0.02	{∅} = 0.0162	{∅} = 0.0038	{4} = 0.0000
m{Θ} = 0.03	{5,6,7,8} = 0.0243	{6,7,8} = 0.0057	{Θ} = 0.0000

K = 0.1805 + 0.0162 + 0.0038 = 0.2005

1-K = 0.7995

m{4} = 0.0000 / 0.7995 = 0.0000

m{5} = 0.7695 / 0.7995 = 0.9625

m{3,4,5} = 0.0000

m{6,7,8} = 0.0071

m{5,6,7,8} = 0.0304

m{Θ} = 0.0000

Table F-10 Intersection Tableau Combining Best-Case Threat C Performance with Combined Best-Case Threat A-B Performance

Thr. A-B / Thr. C	*m{1,2,3} = 0.95*	*m{1,2} = 0.02*	*m{Θ} = 0.03*
m{4} = 0.0000	{∅} = 0.0000	{∅} = 0.0000	{4} = 0.0000
m{5} = 0.9625	{∅} = 0.9144	{∅} = 0.0192	{5} = 0.0289
m{3,4,5} = 0.0000	{3} = 0.0000	{∅} = 0.0000	{3,4,5} = 0.0000
m{6,7,8} = 0.0071	{∅} = 0.0068	{∅} = 0.0001	{6,7,8} = 0.0002
m{5,6,7,8} = 0.0304	{∅} = 0.0289	{∅} = 0.0006	{5,6,7,8} = 0.0009
m{Θ} = 0.0000	{1,2,3}= 0.0000	{1,2} = 0.0000	{Θ} = 0.0000

K = 0.9700; 1-K = 0.0300

m{3} = 0.0000

m{4} = 0.0000

m{5} = 0.9625

m{1,2} = 0.0000

m{1,2,3} = 0.0000

m{3,4,5} = 0.0000

m{6,7,8} = 0.0071

m{5,6,7,8} = 0.0304

m{Θ} = 0.0000

Table F-11 Intersection Tableau Combining Best-Case Threat D Performance with Combined Best-Case Threat A-B-C Performance

Thr. A-B-C / Thr. D	*{0,1,2,3,4,5,6,7,8} = 0.93*	*{1,3,4,5,7} = 0.02*	*{1,4,5,7} = 0.01*	*{5,7} = 0.02*	*{7} = 0.01*	*{Θ} = 0.01*
m{3} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{4} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{5} = 0.9625	0.8951	0.0192	0.0096	0.0192	0.0096	0.0096
m{1,2} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{1,2,3} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{3,4,5} = 0.0000	0.00000	0.0000	0.0000	0.0000	0.0000	0.0000
m{6,7,8} = 0.0071	0.0066	0.0001	0.0001	0.0001	0.0001	0.0001
m{5,6,7,8} = 0.0304	0.0283	0.0006	0.0003	0.0006	0.0003	0.0003
m{Θ} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

The basic probability assignments for all the possible subsets shown in the final intersection tableau, along with the Belief and Plausibility Functions and the Degree of Certainty values, are shown in Table F-12 below.

Table F-12 Best-Case Basic Probability Assignments, Belief Function, Plausibility Function, and Degree of Certainty Values

{}	m{}	Bel{}	Pl{}	DOC{}
{1}	0.0000	0.0000	0.0000	-1.0000
{3}	0.0000	0.0000	0.0000	-1.0000
{4}	0.0000	0.0000	0.0000	-1.0000
{5}	0.9621	0.9621	0.9925	+0.9242
{7}	0.0007	0.0007	0.0379	-0.9985
{1,2}	0.0000	0.0000	0.0000	-1.0000
{1,3}	0.0000	0.0000	0.0000	-1.0000
{4,5}	0.0000	0.9621	0.9925	-0.0379
{5,7}	0.0015	0.9644	1.0000	-0.0341
{1,2,3}	0.0000	0.0000	0.0000	-1.0000
{3,4,5}	0.0000	0.9621	0.9925	-0.0379
{6,7,8}	0.0068	0.0075	0.0379	-0.9857
{1,4,5,7}	0.0000	0.9644	1.0000	-0.0356
{5,6,7,8}	0.0288	0.9712	1.0000	0.0000
{1,3,4,5,7}	0.0000	0.9644	1.0000	-0.0356
{0,1,2,3,4,5,6,7,8}	0.0000	1.0000	1.0000	0.0000
{Θ}	0.0000	1.0000	1.0000	0.0000

The same procedure is followed for the worst-case system performance, with the results shown below.

Table F-13 Intersection Tableau Combining Worst-Case Threat A Performance with Worst-Case Threat B Performance

Thr. A / Thr B	*m{3,4,5} = 0.81*	*m{3,4} = 0.02*	*m{4} = 0.05*	*m{Θ} = 0.12*
m{5,6,7,8} = 0.44	{5} = 0.3564	{∅} = 0.0088	{∅} = 0.0220	{5,6,7,8} = 0.0528
m{6,7,8} = 0.56	{∅} = 0.4536	{∅} = 0.0112	{∅} = 0.0280	{6,7,8} = 0.0672
m{Θ} = 0.00	{3,4,5} = 0.0000	{3,4} = 0.0000	{4} = 0.0000	{Θ} = 0.0000

K = 0.5236

1-K = 0.4764

m{4} = 0.0000

m{5} = 0.7481

m{3,4} = 0.0000

m{3,4,5} = 0.0000

m{6,7,8} = 0.1411

m{5,6,7,8} = 0.1108

m{Θ} = 0.0000

Table F-14 Intersection Tableau Combining Worst-Case Threat C Performance with Combined Worst-Case Threat A-B Performance

Thr. A-B / Thr. C	*m{1,2,3} = 0.83*	*m{1,2} = 0.02*	*m{1} = 0.03*	*m{Θ} = 0.12*
m{4} = 0.0000	{∅} =0.0000	{∅} = 0.0000	{∅} = 0.0000	{4} = 0.0000
m{5} = 0.7481	{∅} = 0.6209	{∅} = 0.0150	{∅} = 0.0224	{5} = 0.0898
m{3,4} = 0.0000	{3} = 0.0000	{∅} = 0.0000	{∅} = 0.0000	{3,4} = 0.0000
m{3,4,5} = 0.0000	{∅} = 0.0000	{∅} = 0.0000	{∅} = 0.0000	{3,4,5} = 0.0000
m{6,7,8} = 0.1411	{∅} = 0.1171	{∅} = 0.0028	{∅} = 0.0042	{6,7,8} = 0.0169
m{5,6,7,8} = 0.1108	{∅} = 0.0920	{∅} = 0.0022	{∅} = 0.0033	{5,6,7,8}= 0.0133
m{Θ} = 0.0000	{1,2,3}= 0.0000	{1,2} = 0.0000	{1} = 0.0000	{Θ} = 0.0000

K = 0.8800; 1-K = 0.1200

m{1} = 0.0000

m{3} = 0.0000

m{4} = 0.0000

m{5} = 0.7481

m{1,2} = 0.0000

m{3,4} = 0.0000

m{1,2,3} = 0.0000

m{3,4,5} = 0.0000

m{6,7,8} = 0.1411

m{5,6,7,8} = 0.1108

m{Θ} = 0.0000

Table F-15 Intersection Tableau Combining Worst-Case Threat D Performance with Combined Worst-Case Threat A-B-C Performance

Thr. A-B-C / Thr. D	*{0,1,2,3,4,5,6,7,8} = 0.74*	*{0,1,2,3,4,5,7,8} = 0.02*	*{1,3,4,5,7} = 0.07*	*{1,4,5,7} = 0.01*	*{1,5,7} = 0.01*	*{5,7} = 0.08*	*{7} = 0.02*	*{Θ} = 0.04*
m{1} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{3} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{4} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{5} = 0.7481	0.5536	0.0150	0.0524	0.0150	0.0075	0.0598	0.0150	0.0299
m{1,2} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{3,4} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{1,2,3} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{3,4,5} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
m{6,7,8} = 0.1411	0.1044	0.0028	0.0099	0.0028	0.0014	0.0113	0.0028	0.0056
m{5,6,7,8} = 0.1108	0.0820	0.0022	0.0078	0.0022	0.0011	0.0089	0.0022	0.0044
m{Θ} = 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Using these basic probability assignments, the final bpa, Belief Function, Plausibility Function, and Degree of Certainty values are given in Table F-16 below.

Table B-16 Worst-Case Basic Probability Assignments, Belief Function, Plausibility Function, and Degree of Certainty Values

{}	m{}	Bel{}	Pl{}	DOC{}
{1}	0.0000	0.0000	0.0000	-1.0000
{3}	0.0000	0.0000	0.0000	-1.0000
{4}	0.0000	0.0000	0.0000	-1.0000
{5}	0.7443	0.7443	0.8545	+0.4886
{7}	0.0309	0.0309	0.2557	-0.9382
{1,2}	0.0000	0.0000	0.0000	-1.0000
{1,3}	0.0000	0.0000	0.0000	-1.0000
{3,4}	0.0000	0.7443	0.8545	-0.2557
{4,5}	0.0000	0.7443	0.8545	-0.2557
{5,7}	0.0203	0.7954	1.0000	-0.1843
{7,8}	0.0029	0.0338	0.2529	-0.9634
{1,2,3}	0.0000	0.0000	0.0000	-1.0000
{1,5,7}	0.0000	0.7954	1.0000	-0.2046
{3,4,5}	0.0000	0.7443	0.8545	-0.2557
{5,7,8}	0.0023	0.8005	1.0000	-0.1972
{6,7,8}	0.1117	0.1455	0.2557	-0.7429
{1,4,5,7}	0.0000	0.7954	1.0000	-0.2046
{5,6,7,8}	0.0878	0.9122	1.0000	0.0000
{1,3,4,5,7}	0.0000	0.7954	1.0000	-0.2046
{0,1,2,3,4,5,7}	0.0000	0.8005	1.0000	-0.1995
{0,1,2,3,4,5,6,7,8}	0.0000	1.0000	1.0000	0.0000
{Θ}	0.0000	1.0000	1.0000	0.0000

The belief intervals are formed from the belief and plausibility function values, giving the final solution to the decision maker in the form of a bounded probability, independent of sample size, on which he can base his decision. The Degree of Certainty value provides an added piece of information on the amount of certainty associated with the decision being made.

REFERENCES

[1] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence,* Englewood Cliffs, NJ: Prentice Hall, 1992.

[2] S.T. Welstead, *Neural Network and Fuzzy Logic Applications in C/C++,* New York: John Wiley & Sons, 1994.

[3] AFOTEC Mission Briefing Script, HQ AFOTEC/XRX, 1993.

[4] W. Sears and T. Miller, Class Notes, EE 8345, Principles of Test and Evaluation, Georgia Institute of Technology, School of Electrical and Computer Engineering, Spring Quarter 1995.

[5] D.L. Giadrosich, *Operations Research Analysis in Test and Evaluation*, Washington D.C.: American Institute of Aeronautics and Astronautics, 1995.

[6] G.E.P. Box, W.G. Hunter, and J.S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building,* New York: John Wiley & Sons, 1978.

[7] Lieutenant General Howard Leaf, USAF Director of T&E, correspondence to Mr. Rusty Porter, President, Association of Old Crows, "EC System Analysis Challenge," August 1993.

[8] G.A. Kent and W.E. Simons, "A Framework for Enhancing Operational Capabilities", *RAND Project Air Force Report,* No. R-4043-AF, Santa Monica: RAND, 1991.

[9] Air Force Operational Test and Evaluation Center, *F-15E Tactical Electronic Warfare System Operational Assessment Test Plan,* May 1994.

[10] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol SMC-3, No. 3, pp.28-44, January 1973.

[11] L. Xuecheng, "Entropy, Distance Measure and Similarity Measure of Fuzzy Sets and Their Relations," *Fuzzy Sets and Systems*, Vol. 52, pp. 305-318, 1992.

[12] D. Bhandari and N.R. Pal, "Some New Information Measures for Fuzzy Sets," *Information Sciences*, Vol. 67, pp. 209-228, 1993.

[13] G.J. Klir and T.A. Folger, *Fuzzy Sets, Uncertainty, and Information,* Englewood Cliffs, NJ: Prentice Hall, 1988.

[14] D.N. Prescott, *Clustering Techniques in Speaker Recognition,* M.S. Thesis, Air Force Institute of Technology, March 1994.

[15] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms,* New York: Plenum, 1981.

[16] L. Wang and R. Langari, "Complex Systems Modeling via Fuzzy Logic," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* Vol. 26, No. 1, February 1996.

[17] R. Krishnapuram and J.M. Keller, "A Possibilistic Approach to Clustering," *IEEE Transactions on Fuzzy Systems,* Vol. 1, No. 2, May 1993.

[18] I. Gath and A.B. Geva, "Unsupervised Optimal Fuzzy Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 11, No. 7, July 1989.

[19] A. Kauffman, *Fuzzy Subsets--Fundamental Theoretical Elements,* Vol. 1, Academic Press, New York, 1980.

[20] A. Deluca and S. Termini, "A Definition of Non-Probabilistic Entropy in the Setting of Fuzzy Set Theory," *Information Control,* Vol. 20, pp. 301-312, 1972.

[21] B. Kosko, "Fuzzy Entropy and Conditioning," *Information Science,* Vol. 40, pp. 165-174, 1986.

[22] N.R. Pal and S.K. Pal, "Higher Order Fuzzy Entropy and Hybrid Entropy of a Set," *Information Science,* Vol. 61, No. 3, pp. 211-231, 1992.

[23] M. Mufti, *Fault Detection and Identification using Fuzzy Wavelets,* Ph.D. Dissertation, Georgia Institute of Technology School of Electrical and Computer Engineering, July 1995.

[24] B.H. Wang and G.J. Vachtsevanos, "Fuzzy Associative Memories: Identification and Control of Complex Systems," *Proceedings of the Fifth International Symposium on Intelligent Control,* pp. 910-915, 1990.

[25] J.A. Dickerson and B. Kosko, "Ellipsoidal Learning and Fuzzy Throttle Control for Platoons of Smart Cars," *Fuzzy Sets, Neural Networks, and Soft Computing,* New York: Van Nostrand Reinhold, pp. 63-84, 1994.

[26] L. Wang and J.M. Mendel, "Generating Fuzzy Rules by Learning from Examples," *Proceedings of the 1991 IEEE International Symposium on Intelligent Control,* pp. 263-268, 1991.

[27] T. Sudkamp and R.J. Hammell, "Interpolation, Completion, and Learning Fuzzy Rules," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 24, No. 2, pp. 332-342, February 1994.

[28] N. Gulley and J.-S.R. Jang, *Fuzzy Logic Toolbox User's Guide*, Natick, MA: The MathWorks Inc., January 1995.

[29] ESAMS Sensitivity Runs were made by Dr. Sam Baty at The BDM Corporation, 1801 Randolph Ave., SE, Albuquerque, New Mexico; 20 Nov 95 - 3 Dec 95. Results were provided in the graphical form shown here.

[30] R. Axelrod, *Structure of Decision: the Cognitive Maps of Political Elites,* Princeton, NJ: Princeton University Press, 1976.

[31] W.R. Taber and M.A. Siegel, "Estimation of Expert Weights Using Fuzzy Cognitive Maps," *IEEE First International Conference on Neural Networks,* Vol. 2, pp. 319-325, 1987.

[32] W.R. Taber and R.O. Deich, "Risk Weighting in Knowledge Bases Using Fuzzy Cognitive Maps," *Proceedings Fourth Annual Artificial Intelligence and Advanced Computer Technology Conference,* pp. 499-502, May 1988.

[33] M. Hagiwara, "Extended Cognitive Maps," *IEEE International Conference on Fuzzy Systems,* pp. 795-801, 1992.

[34] M. Schneider, E. Shnaider, A. Kandel, and G. Chew, "Constructing Fuzzy Cognitive Maps," *Proceedings of the 1995 IEEE International Conference on Fuzzy Systems,* Vol.4, pp. 2281-2288, 1995.

[35] R. Satur, Z.Q. Liu, and M. Gahegan, "Multi-Layered FCMs Applied to Context Dependent Learning," *Proceedings of the 1995 IEEE International Conference on Fuzzy Systems,* Vol. 2, pp. 561-568, 1995.

[36] Contributions for the development of the global EC system FCM shown in the figure were made by Mr. Paul Holt, HQ AFOTEC/XRR; Col Rick Graber, Lt Col Larry Wright, Mr. Eric Stolle, and Mr. Steve Ganger, HQ AFOTEC/TF, Kirtland Air Force Base, NM.

[37] A.P. Dempster, "Upper and Lower Probabilities Induced by a Multi-valued Mapping," *Annals of Mathematical Statistics*, Vol. 38, pp. 325-339, 1967.

[38] G. Shafer, *Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press, 1976.

[39] A. Loskiewicz-Buczak and R.E. Uhrig, "Decision Fusion by Fuzzy Set Operations," *IEEE International Conference on Fuzzy Systems,* Vol. 2, pp. 1412-1417, 1994.

[40] A. Loskiewicz-Buczak and R.E. Uhrig, "Information Fusion by Fuzzy Set Operation and Genetic Algorithm," *Simulation*, Vol. 65, No. 1, pp. 52-66, 1995.

[41] U. Kaymak and H.R. van Nauta Lemke, "Selecting an Aggregation Operator For Fuzzy Decision Making," *IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 1418-1422, 1994.

[42] Status Briefing to Major General George B. Harrison and Dr. Marion L. Williams, Headquarters, Air Force Operational Test and Evaluation Center, Kirtland AFB, New Mexico, 4 March 1996.

[43] J. Lee, *Fuzzy-Connective-Based Information Fusion Networks and Their Application to Computer Vision*, Ph.D. Dissertation, University of Missouri-Columbia, August 1990.

[44] Air Force Electronic Warfare Center Report, *Aggregation Methodology for Electronic Combat Systems*, 1990.

[45] P.L. Bogler, "Shafer-Dempster Reasoning with Applications to Multisensor Target Identification Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-17, No. 6, pp. 968-977, November/December 1987.

[46] A. Goicoechea, "Expert System Models for Inference with Imperfect Knowledge: A Comparative Study," *Journal of Statistical Planning and Inference*, Vol. 20, pp. 245-277, 1988.

[47] D. Dubois and H. Prade, "Evidence Measures Based on Fuzzy Information," *Automatica*, Vol. 21, No. 5, pp. 547-562, 1985.

[48] J. Yen, "Generalizing the Dempster-Shafer Theory to Fuzzy Sets," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 3, pp. 559-570, May/June 1990.

[49] L.A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems*, Vol. 1, No. 1, pp. 3-28, January 1978.

[50] W.J. Conover, *Practical Nonparametric Statistics*, New York: John Wiley & Sons, 1980.

[51] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco: Morgan Kaufmann Publishers, Inc., 1988.

[52] S.M. Blankenship, "Modeling the Fates," *Proceedings of the Modeling, Simulation, and Gaming of Warfare Symposium*, pp. 47-53, December 1991.

[53] C.E. Shannon, "The Mathematical Theory of Communication," in *The Mathematical Theory of Communication*, Urbana: The University of Illinois Press, 1949.

[54] N.R. Pal and J.C. Bezdek, "Measuring Fuzzy Uncertainty," *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 2, pp. 107-118, May 1994.

[55] H.E. Stephanou and S. Lu, "Measuring Consensus Effectiveness by a Generalized Entropy Criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 544-554, July 1988.

[56] Telephone Conversation with Dr. Ernest Seglie, Technical Director to the Director of Operational Test and Evaluation in the Office of the Secretary of Defense, 7 February 1996.

[57] J.M. Irvine, "Effects of A Priori Knowledge and Risk Assessment in Test Design and Analysis Planning," *Air Force Operational Test and Evaluation Center Technical Journal*, Vol. 5., No. 1, pp. 2-9, February 1992.

[58] L.H. Koopmans, *Introduction to Contemporary Statistical Methods,* Second Edition, Boston: PWS Publishers, 1987.

[59] R. Franklin and S.G. Charlton, "The Use of Hypothesis Testing in Air Force OT&E," Briefing presented to the HQ AFOTEC Commander and Staff, October 1990.

[60] N. Singpurwalla, George Washington University Department of Statistics, Correspondence to the Air Force Office of Scientific Research on the use of Bayesian Analysis for OT&E Analysis.

[61] S.M. Beers, "The Use of Statistics in Operational Testing," Briefing to the National Research Council Committee on Statistics, Washington D.C., 14 May 94.

[62] H. Scheffe, *The Analysis of Variance*, New York: John Wiley & Sons, Inc., 1959.

[63] K. Tsui, Class Notes, ISYE 6400, Design of Experiments, Georgia Tech Industrial and Systems Engineering Department, Spring 1995.

[64] G.A. Cooney and R.P. Verseput, "An Innovative Statistical Method for the Design of Industrial Experiments," *Journal of Coatings Technology,* Vol. 59, No. 754, pp. 65-74, November 1987.

[65] E.R. Thompson, "JSOW Operational Test and Evaluation Concept", Briefing on the Test Concept to MGen G.B. Harrison, HQ AFOTEC/CC, March 1995.

[66] *Orthogonal Arrays and Linear Graphs,* American Supplier Institute, Inc., Dearborn, MI, 1987.

[67] S.M. Beers, "A Proposed JSOW Experimental Design," Prepared for Dr. Marion L. Williams, HQ AFOTEC Chief Scientist, 2 June 1995.

[68] Committee on National Statistics of the National Research Council, *Statistical Methods for Testing and Evaluating Defense Systems,* Interim Report, Washington D.C.: National Academy Press, 1995.

[69] J.D. Cryer, *Time Series Analysis*, Boston: PWS-Kent Publishing Company, 1986.

[70] J. Neter, W. Wasserman and M.H. Kutner, *Applied Linear Regression Models*, Second Edition, Burr Ridge: Richard D. Irwin, Inc., 1989.

[71] S.G. Charlton, "Establishing Human Factors Criteria for Space Control Systems," *Human Factors*, Vol. 34, No. 4, pp. 485-501, 1992.

[72] Air Force Operational Test and Evaluation Center Test Plan, *MILSTAR Low Data Rate Operational Test and Evaluation (OT&E) Test Plan*, August 1995.

[73] Air Force Operational Test and Evaluation Center Report, *F-15E Tactical Electronic Warfare System Operational Assessment (OA) Final Report*, AFOTEC TR-94-015, December 1994.

[74] Conversation with Captain Richard Shelley, USAF, HQ AFOTEC Test Manager for the Cheyenne Mountain Upgrade's Alternate Processing and Correlation Center, 12 February 1996.

[75] S.H. Starr, "Technology Assessment," *Journal of the International Test and Evaluation Association,* Vol. 15, No. 3, pp. 17-23, October 1994.

[76] P.K. Davis, "Distributed Interactive Simulation in the Evolution of DoD Warfare Modeling and Simulation," *Proceedings of the IEEE,* Vol. 83, No. 8, pp. 1138-1155, August 1995.

[77] J. Donovan, and D.J. Jaecks, *SRAM II Operational Test and Evaluation Plan,* October 1990.

[78] Major General G.B. Harrison, "Modeling and Simulation in the Acquisition Process," Briefing to the Georgia Tech Continuing Education Program, September 1994. Information for this portion provided by Ms. Cheryl Black, HQ AFOTEC/SAN M&S Analyst for the B-1B program.

[79] Information from B-1B ECM Upgrade Program Test Concept Development meetings at HQ AFOTEC, Summer 1994.

[80] Major General G.B. Harrison, "Modeling and Simulation in the Acquisition Process," Briefing to the Georgia Tech Continuing Education Program, September 1994. Information for this portion provided by Lt Col Catts, HQ AFOTEC/TF F-22 Operational Test Manager.

[81] H.X. Li and V.C. Yen, *Fuzzy Sets and Fuzzy Decision-Making,* Boca Raton, FL: CRC Press, 1995.

[82] E. Cox, *The Fuzzy Systems Handbook: A Practitioners Guide to Building, Using and Maintaining Fuzzy Systems,* Boston: Academic Press Professional, 1994.

[83] M. Braae and D.A. Rutherford, "Selection of Parameters for a Fuzzy Logic Controller," *Fuzzy Sets and Systems,* Vol. 2, pp. 185-199, 1979.

[84] L.A. Zadeh, "Fuzzy Sets," *Information and Control,* Vol. 8, pp. 338-353, 1965.

[85] R.R. Yager and D.P. Filev, *Essentials of Fuzzy Modeling and Control,* New York: John Wiley & Sons, 1994.

[86] L.A. Zadeh, "Making Computers Think Like People," *IEEE Spectrum,* Vol. 21, Part 2, pp. 26-32, August 1984.

[87] E.H. Mamdani, "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," *IEEE Transactions on Computers,* Vol. C-26, No. 12, pp. 1182-1191, December 1977.

[88] L.A. Zadeh, "Syllogistic Reasoning in Fuzzy Logic and it Application to Usuality and Reasoning with Dispositions," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-15, No. 6, pp. 754-763, December 1985.

[89] R.R. Yager and D.P. Filev, "On the Reasoning in Fuzzy Logic Control and Fuzzy Expert Systems," *1993 IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 839-844, March 1993.

[90] L.A. Zadeh, "Fuzzy Probabilities," *Information Processing and Management*, Vol. 20, No. 3, pp. 363-372, 1984.

[91] J.J. Choi, K.H. O'Keefe and P.K. Baruah, "Non-linear System Diagnosis Using Neural Networks and Fuzzy Logic," *IEEE International Conference on Fuzzy Systems,* pp. 813-820, March 1992.

[92] W. Zhang and S. Chen, "A Logical Architecture for Cognitive Maps," *IEEE International Conference on Neural Networks,* Vol. 1, pp. 231-238, 1988.

[93] B. Kosko, "Fuzzy Cognitive Maps," *International Journal of Man-Machine Studies,* Vol. 24, pp. 65-75, January 1986

[94] M. Caudill, "Using Neural Nets: Fuzzy Cognitive Maps," *AI Expert,* Vol. 5, No. 6, pp. 49-53, June 1990.

[95] R. Taber, "Knowledge Processing with Fuzzy Cognitive Maps," *Expert Systems with Applications,* Vol. 2, pp. 83-87, 1991.

[96] B. Kosko, "Adaptive Inference in Fuzzy Knowledge Networks," *IEEE First International Conference on Neural Networks,* Vol. 2, pp. 261-268, 1987.

[97] C.E. Pelaez, *A Fuzzy Cognitive Map Knowledge Representation for Performing Failure Modes and Effects Analysis,* Ph.D. Dissertation, University of South Carolina, 1994.

[98] R. Taber, "Fuzzy Cognitive Maps Model Social Systems," *AI Expert,* Vol. 9, No. 7, pp. 19-23, July 1994.

[99] C.E. Pelaez and J.B. Bowles, "Using Fuzzy Cognitive Maps as a System Model for Failure Modes and Effects Analysis," *Information Sciences,* Early 1996 (expected).

[100] R.A. Hummel and M.S. Landy, "A Statistical Viewpoint of the Theory of Evidence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 2, pp. 235-247, March 1988.

[101] J. Gordon and E.H. Shortliffe, "A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space," *Artificial Intelligence*, Vol. 26, No. 3, pp. 323-357, July 1985.

[102] J.B. Yang and M.G. Singh, "An Evidential Reasoning Approach for Multiple-Attribute Decision Making with Uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 1, pp. 1-18, January 1994.

[103] A. de Korvin and M.F. Shipley, "A Dempster-Shafer-Based Approach to Compromise Decision Making with Multiattributes Applied to Product Selection," *IEEE Transactions on Engineering Management*, Vol. 40, No. 1, pp. 60-67, February 1993.

[104] S.J. Henkind and M.C. Harrison, "An Analysis of Four Uncertainty Calculi," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No. 5, pp. 700-714, September/October 1988.

[105] I. Kim, *A Hybrid Analytical/Intelligent Methodology for Sensor Fusion*, Ph.D. Dissertation, Georgia Institute of Technology School of Electrical and Computer Engineering, December 1992.

Vita

Suzanne Beers was born the only daughter of a research librarian and an industrial engineer on [redacted]. She earned the B.S. (Summa Cum Laude) in Physics from Miami University in Oxford, Ohio and the B.S. in Mechanical Engineering from Ohio State University in Columbus prior to her being commissioned a Second Lieutenant in the United States Air Force.

She entered active duty in November 1983 at Kirtland Air Force Base, New Mexico where she served as a Nuclear Systems Physicist, developing design criteria for cruise missile programs such as the Air Launched Cruise Missile, Sea Launched Cruise Missile, and the Advanced Cruise Missile. During her off-duty hours, she earned an M.B.A. from the New Mexico Highlands University. In October 1987, with a promotion to the rank of Captain, she was assigned to the Ground Launched Cruise Missile's 487th Tactical Missile Wing at Comiso Air Station, Sicily. While at the wing, Captain Beers served as a Launch Control Officer, Weapon System Instructor, Emergency Actions Procedure Instructor, and Battle Planner. Returning from overseas, she was assigned to the Air Force Operational Test Evaluation Center Headquarters at Kirtland Air Force Base, New Mexico. There Captain Beers served as the Lead Operational Effectiveness Analyst for the Short Range Attack Missile II Program, the Survivability Analyst for the Milstar Satellite Communications System, the Test Manager for the Cheyenne Mountain Upgrade Program, and as the Advisor to the Chief Scientist. During her off-duty hours, she earned an M.S.E.E. from the New Mexico State University in Las Cruces, New Mexico, completing the program with a 4.0 GPA.

Selected for the Air Force Institute of Technology / Air Force Operational Test and Evaluation Center - sponsored Ph.D. program, Captain Beers entered the Georgia Institute of Technology in the fall of 1994. Promoted to the rank of Major in October 1995, Major Beers will resume her duties as the Advisor to the Chief Scientist at Headquarters, Air Force Operational Test and Evaluation Center upon the completion of her degree.