

**DIMENSIONALITY IN FUZZY SYSTEMS**

A Dissertation

by

**WALLACE EUGENE KELLY, III**

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

**DOCTOR OF PHILOSOPHY**

August 1997

Major Subject: Electrical Engineering

# DIMENSIONALITY IN FUZZY SYSTEMS

A Dissertation

by

WALLACE EUGENE KELLY, III

Submitted to Texas A&M University  
in partial fulfillment of the requirements  
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

---

John H. Painter  
(Chair of Committee)

---

Donald T. Ward  
(Member)

---

Costas Georghiades  
(Member)

---

Norman Griswold  
(Member)

---

Rajab Chaloo  
(Member)

---

Chanan Singh  
(Head of Department)

August 1997

Major Subject: Electrical Engineering

**ABSTRACT**

Dimensionality in Fuzzy Systems. (August 1997)

Wallace Eugene Kelly, III, B.S., Texas A&M University – Kingsville;

M.S., Texas A&M University – Kingsville

Chair of Advisory Committee: Dr. John H. Painter

This dissertation explores the theoretical and practical aspects of dimensionality in fuzzy systems. First, the author shows that fuzzy logic can be formulated from first principles of Bayesian probability theory. Such a formulation helps focus theoretical development of fuzzy logic techniques. For example, the effect of anomalous inputs on various forms of fuzzy inference can be understood and considered during the design process. The Bayesian interpretation of fuzzy logic has also guided a fundamental improvement in the state-of-the-art of fuzzy system engineering. Known as *hypertrapezoidal fuzzy membership functions* (HFMF), this new method of defining multidimensional fuzzy relationships is motivated by an on-going research project in smart-cockpit technologies. The Automated Safety and Training Avionics project of Texas A&M seeks to improve the safety of the general aviation industry by utilizing artificial intelligence techniques in on-board avionics systems. Efforts to enhance on-board situational awareness revealed a fundamental weakness in fuzzy logic systems. HFMFs address this weakness by enabling the design of correlated fuzzy models with relatively few parameters. HFMFs can be successfully used for automatic flight mode interpretation and hold promise for many other applications. Finally, the author suggests directions for future research related to multidimensional fuzzy engineering.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	iii
TABLE OF CONTENTS .....	iv
LIST OF FIGURES .....	vi
LIST OF TABLES .....	viii
CHAPTER	
I INTRODUCTION.....	1
Research Overview.....	1
Aircraft Metacontrol.....	2
Flight Mode Interpretation.....	4
Fuzziness and Uncertainty.....	5
Other Research Efforts.....	6
Dissertation Overview.....	7
II FUZZY DECISION THEORY .....	8
Fuzzy Logic.....	8
Fuzzy Systems.....	11
Bayesian Probability.....	18
The Bayes/Fuzzy Isomorphism.....	19
Dealing With Uncertainty.....	22
III DIMENSIONALITY IN FUZZY SYSTEMS.....	30
One-Dimensional Fuzzy Systems.....	30
PDF-Based Fuzzy Membership Functions.....	34
Hypertrapezoidal Fuzzy Membership Functions.....	35
Fuzzy System Architectures.....	41
IV MULTIDIMENSIONAL FLIGHT MODE INTERPRETATION.....	49
Automated Safety and Training Avionics.....	49
The Engineering Flight Simulator.....	50
The Flight Modes.....	51
Quantifying Results.....	52
One-dimensional FMI Baseline.....	53
One-dimensional FMI Upgrade.....	57
Multidimensional FMI Results.....	67
V CONCLUSIONS AND RECOMMENDATIONS .....	73
Innovations.....	73
Recommendations for Future Work.....	74
REFERENCES .....	78

	Page
APPENDIX A PROOFS AND DERIVATIONS.....	81
Control Density Equation.....	81
Derivation of HFMF Equations.....	82
Bayesian Interpretation of Decision Confidence.....	87
APPENDIX B FMI MATLAB™ TOOLBOX.....	89
Background.....	89
Requirements.....	89
Commands.....	90
APPENDIX C HFMF MATLAB™ TOOLBOX.....	100
Background.....	100
Requirements.....	100
Commands.....	100
VITA .....	107

## LIST OF FIGURES

FIGURE	Page
1 Original system architecture, including metacontroller.....	2
2 Updated system architecture, including pilot advisor.....	3
3 An aircraft's state space can be partitioned into operational modes.....	4
4 Common shapes for fuzzy membership functions.....	9
5 Hard and soft fuzzy connectives for sets in the same domain.....	10
6 Hard and soft fuzzy connectives for sets in different domains.....	11
7 General structure of a fuzzy system.....	12
8 Example of fuzzification of altitude error into modes JustRight and High.....	13
9 Two methods of fuzzy implication.....	15
10 Two definitions for the aggregation of rule consequents.....	16
11 Two methods of defuzzification – centroid and mean of maxima.....	17
12 Comparison of fuzzy and Bayesian controllers.....	22
13 The intersection of membership functions in a three-dimensional state space.....	24
14 The union of membership functions in a three-dimensional space.....	25
15 Effect on membership functions when setting a threshold.....	26
16 The modeled region in state space when using a threshold.....	26
17 The modeled region in state space when using a normalized sum.....	27
18 The effect of the inference method in the fuzzy regions of the state space.....	28
19 Footprint of fuzzy set when the input variables are correlated.....	31
20 Fuzzy sets of rule base composition example.....	32
21 Rule base composition is an approximation of a multidimensional relationship.....	33
22 A two-dimensional Gaussian membership function.....	35
23 Defining a one-dimensional trapezoidal membership function.....	35
24 Two-dimensional trapezoidal membership function.....	36
25 Prototype points defining a fuzzy partitioning.....	37
26 The effect of $\sigma$ on a two-dimensional fuzzy partitioning.....	38
27 The effect of $\sigma$ on one-dimensional fuzzy sets.....	38
28 Defining the overlap of a fuzzy partition.....	39
29 Example of three fuzzy sets defined on a two-dimensional space.....	40

FIGURE	Page
30 Example of four fuzzy sets defined on a two-dimensional space.....	41
31 General architecture for set composition model.....	42
32 General structure of the hierarchical model.....	43
33 System architecture for parallel competitive fuzzy systems.....	45
34 System architecture for parallel cooperative systems.....	47
35 The Engineering Flight Simulator projectors.....	50
36 The Engineering Flight Simulator cockpit.....	51
37 Plot of both pilot's and inferred flight mode.....	53
38 The relationship between mode certainty and decision confidence.....	55
39 Anomalous inputs caused oscillations in base-line FMI.....	57
40 Plot of base-line FMI output for a landing.....	58
41 Plot of re-tuned FMI output for the landing of Figure 40.....	59
42 Example of nervousness near mode transitions.....	60
43 Flight mode decision with filtering, for flight of Figure 42.....	60
44 Distance ratios for inferring initial approach.....	62
45 Plot of distance ratios used in upgraded FMI.....	64
46 Fuzzy regions defined by distance ratios in upgraded FMI.....	64
47 Fuzzy membership functions defined for the distance ratios.....	65
48 Flight mode interpretation of approach using distance ratios.....	66
49 HFMFs for an altitude of 500 feet above ground.....	69
50 HFMFs for an altitude of 1500 feet above ground.....	69
51 HFMFs for an altitude of 3000 feet above ground.....	70
52 HFMFs for an altitude of 4500 feet above ground.....	70
53 Performance of three input HFMF-based Flight Mode Interpreter.....	72
54 Using statistical means as prototype points.....	76
55 Partitioning a two-dimensional plane with two prototype points.....	82
56 A point, $x$ , on the boundary of the fuzzy overlap.....	83
57 Degree of membership of point in fuzzy region.....	85
58 Two overlapping fuzzy sets defined on two dimensions.....	85

**LIST OF TABLES**

TABLE		Page
1	Fuzzy rules for a simple example of rule base composition.....	32
2	Flight modes identified by the FMI.....	54
3	Inputs to the baseline FMI.....	54
4	Harral's one-dimensional fuzzy membership functions.....	56
5	Distances used in upgraded FMI.....	61
6	Distance ratios used in upgraded FMI.....	63
7	Fuzzy membership functions for the FMI upgrade.....	67
8	Prototype points for example HFMFs.....	71

## CHAPTER I

### INTRODUCTION

#### RESEARCH OVERVIEW

This dissertation reports research pushing forward the theory and application of fuzzy logic and fuzzy control. The push is in a theoretical area needed to apply fuzzy logic to the interpretation and management of systems of increasing complexity. The theoretical work contained herein is motivated by engineering development presently being sponsored in the automation of aircraft flight management. That ongoing development also proves to be an effective test-bed for the theoretical results.

Use of onboard flight direction avionics has been standard in the commercial aircraft industry for some time. In 1989, under NASA sponsorship [30], Texas A&M embarked on research to extend the flight director concept. The improvements involved incorporating recent results from the field of artificial intelligence and expert systems. The ability to maintain operation in the presence of uncertainty was of primary importance.

During the NASA-sponsored project of 1989-94, a fuzzy expert system was developed in simulation for interpretation and direction of the flight operations of a Boeing-737 jet transport aircraft. The project performed by Steve Lass during that time period was an onboard fuzzy flight manager for controlling the descent from altitude of the jet, under operational constraints [23]. The constraints essentially restricted the aircraft from entering certain regions of its operational state space. Such a flight manager would have been impossible, using crisp rules. As it was implemented, the fuzzy flight manager used forty rules to navigate around prohibited state space regions.

In the 1989-94 series, the application was to autonomous operation, closing the control loop back through the automatic flight control system. The expert system functioned to formulate commands to the auto-pilot which were then executed with pilot override capability. A follow-on NASA-sponsored commercialization project, commencing in 1994, focused on smaller aircraft, of the light-twin category, and refocused functionality from autonomous *metacontrol* to *pilot advising*.

Termed *ASTRA*©, the Automated Safety and Training Avionics project was first conceived by Dr. John Painter in 1988. The goals of the *ASTRA* project are twofold. First, the goal is to

---

This dissertation follows the format of *IEEE Transactions on Fuzzy Systems*.

improve the safety of general aviation aircraft. Second, the goal is to improve training of pilots for those aircraft. These goals are being accomplished through the incorporation of an onboard pilot advisory system using the latest in computer hardware and artificial intelligence techniques. The onboard system improves the pilot's situational awareness by monitoring the pilot's flight performance, informing the pilot of anomalies, and assisting in flight management.

## AIRCRAFT METACONTROL

At the heart of the development of the pilot advisory system is the idea of an aircraft metacontroller proposed by Painter [27] and demonstrated by Glass [14]. A metacontroller differs from a flight control system in that a metacontroller generates high-level messages for the automatic flight control system auto-pilot. The metacontroller relies on a computer-encoded knowledge base to give high level commands for flying the aircraft. Figure 1 is taken from [27] in which the original metacontroller and pilot advisor concept are presented. The idea of a metacontroller gives a convenient hierarchical structure to the task at hand.

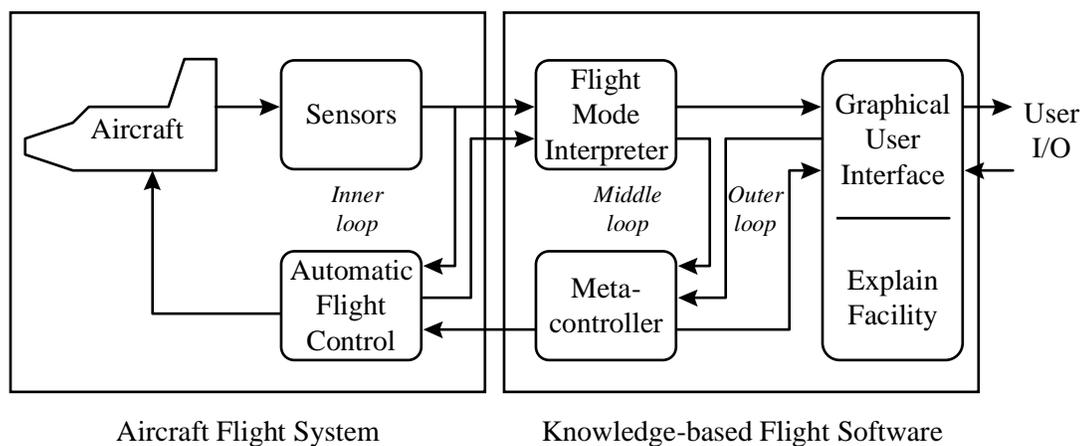


Figure 1. Original system architecture, including metacontroller.

The current state of the research does not pursue direct control of the aircraft, but rather seeks to aid the pilot through monitoring and advising [31]. Figure 2 is a modification of the original system architecture that reflects this change in functionality. Figure 2 first appears in [21]. The pilot advisor now uses encoded knowledge to act as an assistant to the pilot. The knowledge is encoded using fuzzy logic and expert system rules. The domain of the encoded knowledge is related

to safety, navigation, and general aircraft performance issues. Such a pilot advisory system is a valuable step toward increased automation in the cockpit.

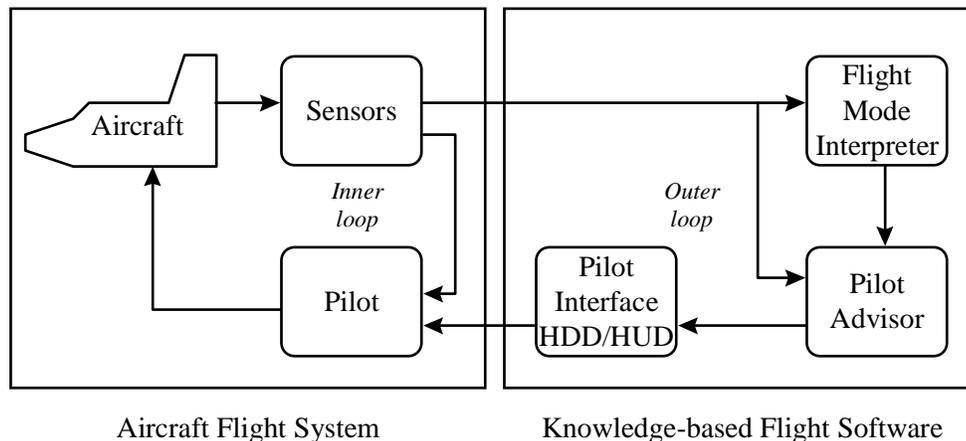


Figure 2. Updated system architecture, including pilot advisor.

There are four primary components to the current system architecture. The pilot and the aircraft are of course the central members of the overall system. The *Pilot Advisor* (PA) and the *Flight Mode Interpreter* (FMI) are the other two primary components. The PA serves the function of metacontroller. The PA and FMI are basically two software modules running in an on-board avionics computer. The Flight Mode Interpreter is a fuzzy logic system that classifies the current state of the aircraft into operational modes. The Pilot Advisor is an expert system that generates messages for the pilot based on the inferred flight mode, aircraft sensor readings, and information from the pilot. A special suite of sensors is necessary to interface the aircraft to the on-board computer. A *head-down display* (HDD) and a *head-up display* (HUD) provide an interface to the pilot.

The Pilot Advisor is an expert system, with expertise in aircraft limitations, navigation, flight procedures, etc. It relies on the Flight Mode Interpreter to indicate the current operational mode of the aircraft. Based on that mode, the Pilot Advisor “fires” a set of rules that check physical limitations of the aircraft, displays information relevant to the current mode on the HUD and the HDD, and assists the pilot in flight planning. The PA manages displays on the HUD and HDD to provide the clearest possible view of the situation and to support pilot activities. A goal of the Pilot Advisor is to decrease the pilot’s workload during the flight.

## FLIGHT MODE INTERPRETATION

The key to both the metacontroller and the pilot advisor system is the modeling of aircraft operations as predefined *operating modes*. These operating modes are inferred from measurements of standard flight variables, such as airspeed, altitude, etc. Specific operating modes then correspond to partitions of an operational state space, as shown in Figure 3. Flight-mode inference is possible by observing into which state space partition flight-sensor measurements fall. Given such an inference and a set of rules defining the operational modes as functions of flight variables, flight guidance may be formulated, to fine-tune the observed operational mode.

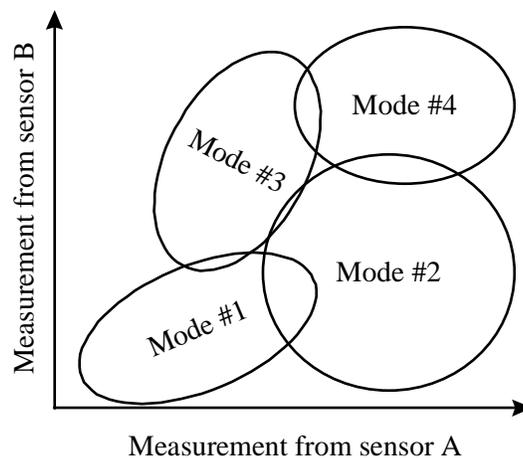


Figure 3. An aircraft's state space can be partitioned into operational modes.

The Flight Mode Interpreter is responsible for making a decision about the aircraft operating mode based on sensor information, navigational information, and mission planning. The FMI decreases the need for pilot input to the advisory system by automatically inferring the current stage of a flight. The advice, alarms, and symbology of the HUD and HDD are driven by the flight mode inferred by the Flight Mode Interpreter. There are four basic requirements of the FMI.

- It must provide the correct flight mode. The measure of the FMI's performance is based on how closely the FMI can match a pilot's intended mode.
- It must provide information about the certainty and confidence associated with the decision.

- It must be robust. That is, it must be able to make the mode decision even if not all the sensor readings point to the same mode.
- It must not be “nervous”. Quick oscillations between modes will translate into HUD symbology that blinks off and on, advice that is changing and perhaps even conflicting, and alarm messages that come and go.

## **FUZZINESS AND UNCERTAINTY**

The overlap between the modes of Figure 3 is intentional and reflects the uncertainty that exists when defining flight modes. Uncertainty results from the fact that the transitions from one mode to the next are gradual, not discrete. Other sources of uncertainty include differences in flying technique between pilots and difference in flight conditions from day to day. These uncertainties are what motivate a fuzzy solution to the problem of flight mode interpretation. Crisp, discrete boundaries can not be drawn between flight modes.

The concept of a fuzzy set was introduced by Lotfi Zadeh in 1965 [42]. The stated purpose was to deal with “classes” that have no “sharply defined criteria of class membership.” Fuzzy sets allow the construction of system models when the sets that comprise the system are not clearly defined. Such is the case for the operational flight modes of an aircraft. For the *ASTRA* project, fuzzy sets provide a way to partition the operating space into fuzzy, sometimes ambiguous modes.

Fuzzy systems are primarily useful to the engineer as a means for encoding human knowledge and expertise into systems. Fuzzy logic lends itself well to automating human decision processes, when the decision rules can be stated in terms of if/then rules. Fuzzy logic allows for the presence of uncertainty when making decisions. Previous fuzzy control applications have been successful for systems having relatively few measured states and relatively few control rules. One example is train positioning during station arrival, for the Sendai subway in Japan [34]. By way of comparison, flight management is orders of magnitude more complex.

The recent surge of fuzzy applications has demonstrated the above strength. It has also revealed some fundamental weaknesses. In particular, the current state-of-the-art is impractical for complex systems. It is the complexity of multivariate flight mode interpretation that motivates much of the research documented in this dissertation. Specifically, multidimensional fuzzy membership functions provide a way to partition a multivariate state space into fuzzy regions. Use of such functions significantly simplifies fuzzy implementation for complex systems. For complex

control and decision problems with many inputs, fuzzy systems require a huge rule base. This rule base increases exponentially with the number of inputs. Large rule bases are difficult to encode and even more difficult to verify. A very important result from Lass's project [23] was that by shifting from one-dimensional to two-dimensional fuzzy membership functions, the 40-rule set was reduced to 2 rules, a 20-fold saving in complexity.

Another weakness in the current state-of-the-art for fuzzy systems engineering is the cumbersome and inconsistent manner in which correlation between input variables is modeled. In the example of flight mode interpretation, correlation exists between input variables when defining the flight modes. The diagonal orientation of modes 1 and 3 in Figure 3 are the result of correlation. Multidimensional fuzzy membership functions address the issue of rule base size and correlation.

## **OTHER RESEARCH EFFORTS**

Texas A&M University is not alone in recognizing the potential benefits of incorporating advanced computing techniques into the cockpit. The Advanced General Aviation Transport Experiments (AGATE) is a consortium composed of NASA, the FAA, industry and university representatives [8]. Created in 1994 and given an eight year time-frame, the goal of the AGATE consortium is to revitalize the general aviation industry. Many in the AGATE consortium have a vision for a Small Aviation Transportation System (SATS) which would serve as an alternative to short-range automobile trips for both personal and business transportation.

Several key capabilities are motivating an upgrade to the status-quo of avionics. The global positioning system (GPS) will continue to revolutionize and ease the task of on-board navigation. Indeed, GPS is an essential element of the *ASTRA* program. On-board traffic avoidance systems and data link capabilities will also increase the possibility for a new generation of avionics. On-board radar imaging and weather awareness technologies will allow for safety to be further increased.

A research effort similar to *ASTRA* is also being conducted by Search Technology, Inc. in Norcross, Georgia [36]. The goal of their Hazard Monitor software is to prevent "the negative consequences of hazardous situations that arise in the cockpit." The target aircraft for the Hazard Monitor are larger, commercial and transport aircraft. Like *ASTRA*, the Search Technology program does not seek to control the aircraft directly, but rather notifies the pilot of potential problems.

As more and more information becomes available in the cockpit, the need for on-board data fusion capabilities will become essential. It will also enable a greater degree of situational awareness. The on-going smart-cockpit research at Texas A&M University is poised to help usher in a new generation of avionics.

## **DISSERTATION OVERVIEW**

The following chapters develop new techniques for utilizing fuzzy logic in complex systems, like smart-cockpit avionics. Chapter II contains a unified presentation of the theory behind fuzzy logic, including a clear demonstration of the relationship between fuzzy logic and Bayesian probability. This relationship motivates the development of hypertrapezoidal fuzzy membership functions in Chapter III. This new technique for defining and using multidimensional fuzzy relationships is an important extension to fuzzy logic with application in fuzzy control and classification. Chapter III also includes a discussion of the various system architectures that can assist the engineer in designing fuzzy systems. Chapter IV is a comprehensive collection of results obtained from the author's research applied to the domain of flight mode interpretation. The chapter includes results from applying hypertrapezoidal fuzzy membership functions to the flight mode interpretation process, the use of mode filtering techniques, and other upgrades to the *ASTRA* project. Conclusions and recommendations for further work are listed in Chapter V.

## CHAPTER II

### FUZZY DECISION THEORY

#### FUZZY LOGIC

##### *Fuzzy Sets*

The concept of a fuzzy set was introduced by Lotfi Zadeh in 1965 [42]. According to Zadeh, the stated purpose of fuzzy sets is to deal with “classes” that have no “sharply defined criteria of class membership.” A fuzzy set is completely defined by its fuzzy membership function,  $\mu(x)$ , which gives the degree of membership of an element,  $x$ , in a fuzzy set. The classic example is that of the set of tall people. The height of a person will indicate whether or not a person is tall, but the boundary between tall people and short people can not be drawn at some exact height. Fuzzy sets allow the construction of system models when the sets that comprise the model are not clearly defined.

The membership function indicates the degree of membership of a crisp value in a fuzzy set. A variety of basic shapes can be used to design the fuzzy membership functions. The trapezoidal fuzzy membership function is the most popular. Trapezoids are easily specified and calculated. The Gaussian probability function is useful in problems requiring adaptation because it is everywhere differentiable. Usually, the choice is based more on personal preference than any mathematical justification. Figure 4 shows the common basic shapes used in fuzzy systems.

One motivation for the use of fuzzy sets is in modeling human expertise. Generally, a human’s perception of a system is not based on precise mathematical models. We tend to understand systems on more of a heuristic level. A computer’s strength, on the other hand, is in precise numerical or repetitive tasks. Fuzzy logic is a way of mapping human knowledge into a form useable by a computer. The last ten years have seen an explosion in the use of fuzzy logic for solving many practical problems.

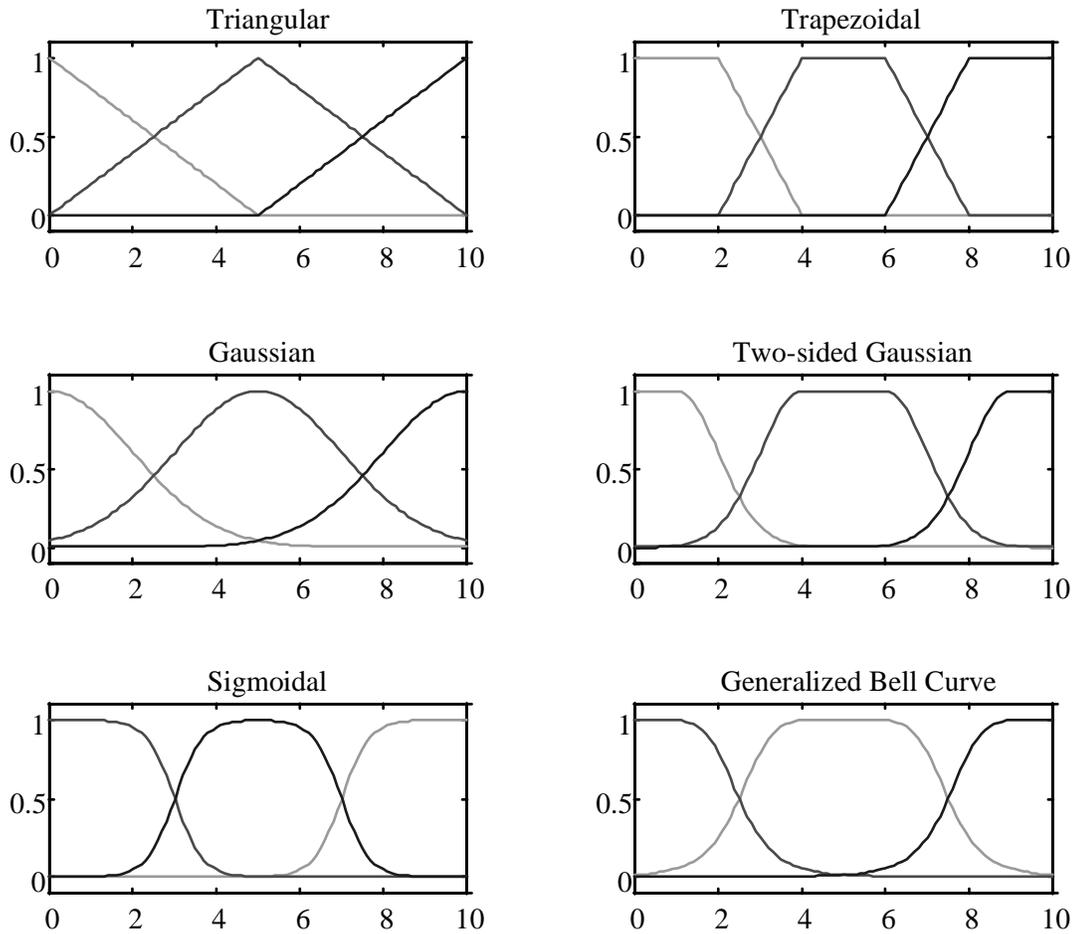


Figure 4. Common shapes for fuzzy membership functions.

### ***Fuzzy Set Operations***

Once fuzzy sets have been defined, the next step is to perform operations on those fuzzy sets. As of yet, there is not a consensus in the fuzzy community regarding how fuzzy set operations should be defined. The more common are known as the soft and hard connectives. If the fuzzy membership functions are viewed as Bayesian conditional probabilities, the soft fuzzy connectives provide consistency [28]. Figure 5 shows plots of both the soft and hard connectives for two sets defined in the same domain.

The first fuzzy set operation to define is intersection, or the AND operator. For fuzzy sets  $A$  and  $B$ , with degrees of membership  $\mu_A(x)$  and  $\mu_B(y)$ , the intersection of  $A$  and  $B$  is defined by  $\mu_{A \cap B}(x, y)$ , where

$$\begin{aligned} \text{Soft: } \mu_{A \cap B}(x, y) &= \mu_A(x) \cdot \mu_B(y) \\ \text{Hard: } \mu_{A \cap B}(x, y) &= \min(\mu_A(x), \mu_B(y)) \end{aligned} \quad (1)$$

The second fuzzy set operation to define is the union, or the OR operator.  $\mu_{A \cup B}(x, y)$  is defined by

$$\begin{aligned} \text{Soft: } \mu_{A \cup B}(x, y) &= \mu_A(x) + \mu_B(y) - \mu_A(x) \cdot \mu_B(y) \\ \text{Hard: } \mu_{A \cup B}(x, y) &= \max(\mu_A(x), \mu_B(y)) \end{aligned} \quad (2)$$

The third fuzzy set operation to define is complement, or the NOT operator.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (3)$$

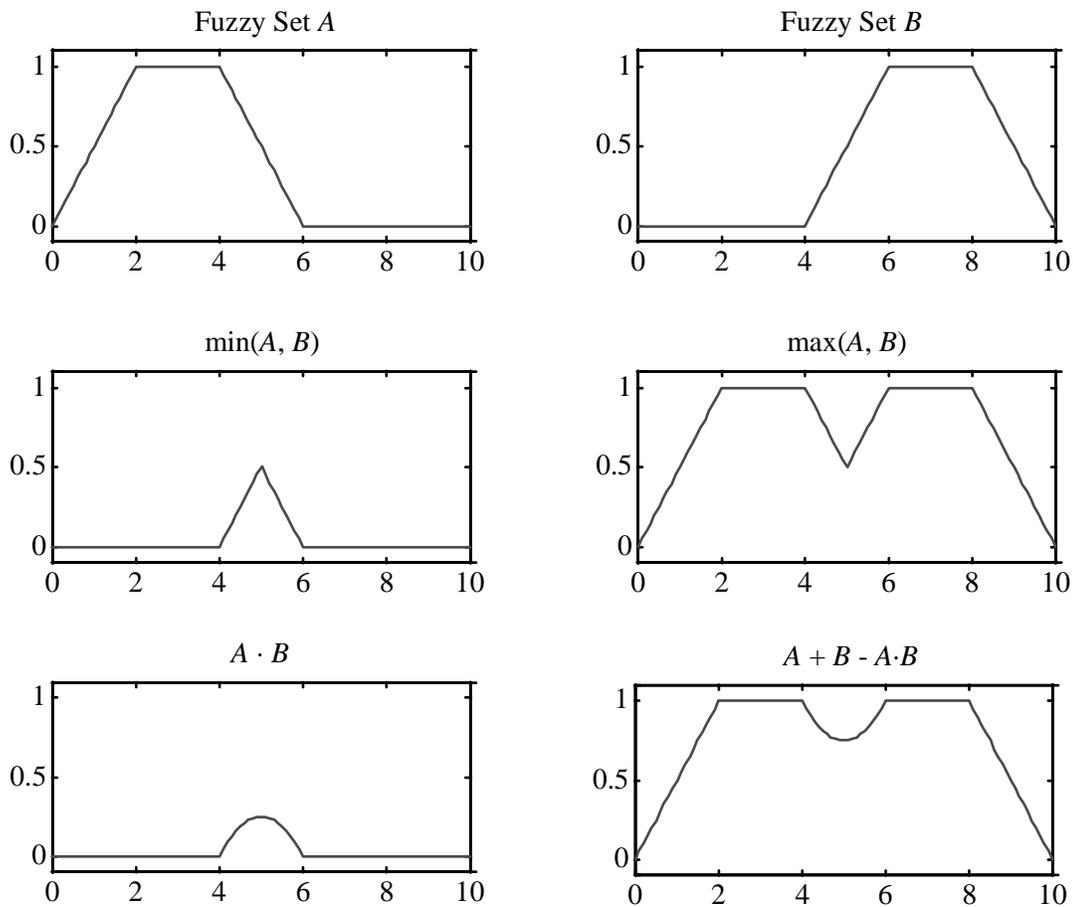


Figure 5. Hard and soft fuzzy connectives for sets in the same domain.

While Figure 5 shows the fuzzy operations being performed on two variables in the same domain, generally set operations are performed on sets of different domains. Sets might be defined on the domains of altitude and airspeed to determine the flight mode of an aircraft. In this case, the operations are defined in the rule base. Figure 6 gives more insight into the use of the three basic fuzzy set operations for operations in multiple domains.

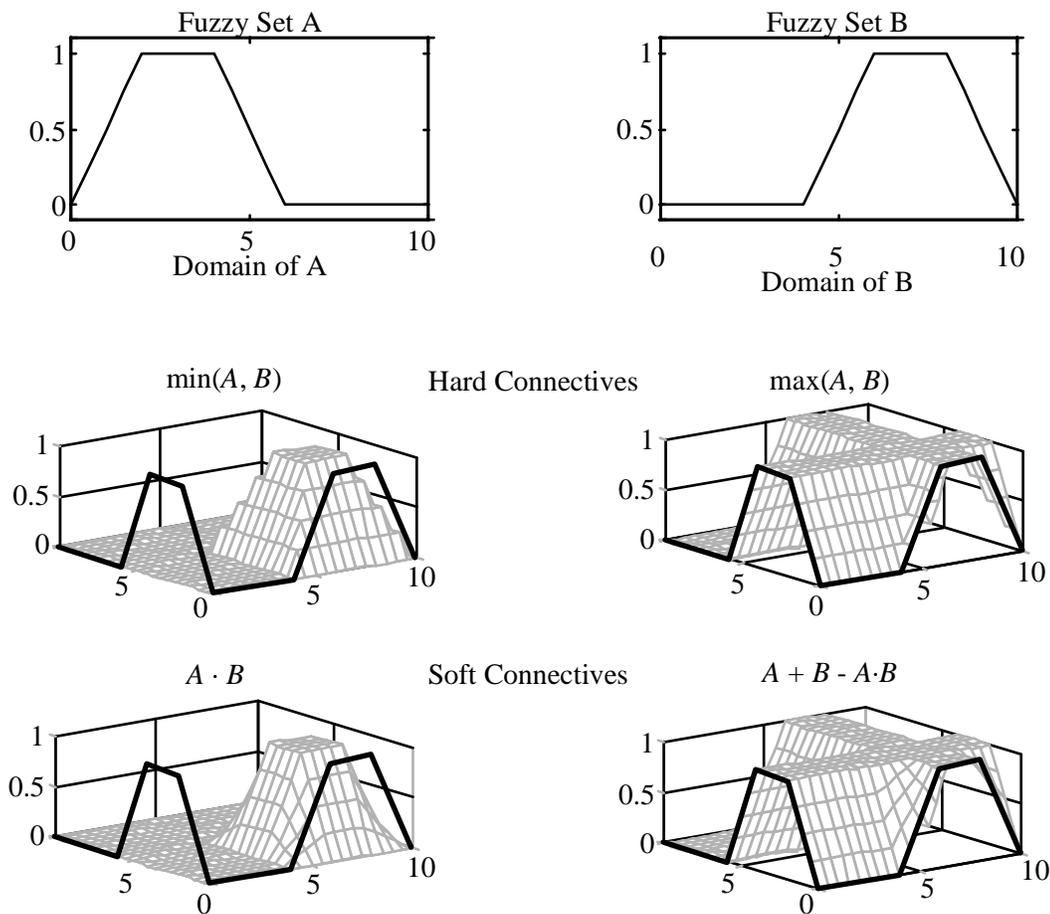


Figure 6. Hard and soft fuzzy connectives for sets in different domains.

## FUZZY SYSTEMS

A fuzzy system uses fuzzy sets to perform a mapping of input variables to output variables. They are helpful in situations which require the embedding of human expertise in a control or advisory computer. Traditional control is based on mathematical models of the system being

controlled. Fuzzy systems generally do not rely on a mathematical model of the system, but rather on the knowledge of an expert familiar with the system.

Fuzzy systems have been used in a variety of applications. Applications include tracking, tuning, classification, voice recognition, financial predictions, automotive transmissions, washing machines, image stabilization in video cameras, medical diagnosis, and most recently for flight mode analysis of aircraft [27]. Fuzzy systems are a cost effective method for designing nonlinear systems based on an heuristic understanding of the desired behavior. Fuzzy systems are being used to perform tasks that are beyond the scope of traditional control techniques.

A fuzzy system can be broken into three parts -- fuzzifier, rule base, and defuzzifier. Figure 7 shows a typical fuzzy system. The rule base is an essential element of any fuzzy system. The fuzzifier and defuzzifier may not be needed in all situations.

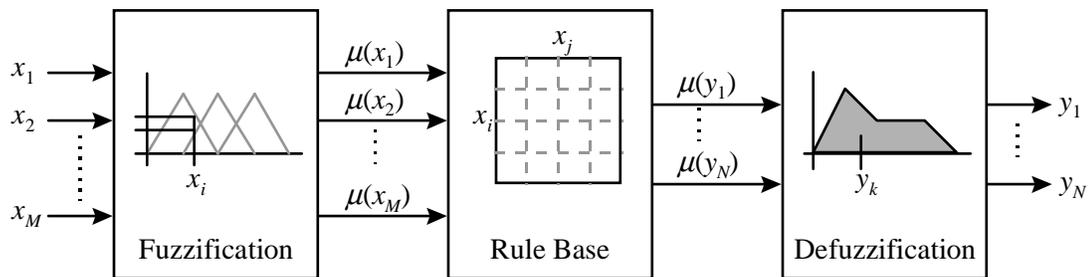


Figure 7. General structure of a fuzzy system.

### ***Fuzzification***

A fuzzifier is needed in cases where the input to the system is in the form of numerical data. This would be the case when the inputs are coming from sensors which are measuring physical quantities in the system. In some cases, the input to the fuzzy system may not come in the form of numerical data, but as fuzzy sets. This would be the case for a medical diagnosis program when a doctor might qualify the redness of a rash on a patient.

Similarly, not all situations require a defuzzifier. In flight mode analysis, for example, the output of the system is the operational mode of the aircraft. No conversion to the real number line is needed. On the other hand, if the mode of operation of an aircraft was being used to drive inputs to that aircraft, the final fuzzy sets would have to be defuzzified to generate actual throttle or control surface positions. Consider a greatly simplified example of an auto-pilot. Each of three

components of a fuzzy system would be needed to maintain altitude, heading, and airspeed of an aircraft.

A fuzzy-based auto-pilot would require sensor readings of the variables which define the aircraft's state. These might include, among others, the altitude, heading, and indicated airspeed. The desired altitude, heading, and indicated airspeed would also be inputs, as well as the current position of the throttle and control surfaces. Each of these input variables must first be mapped to a degree of membership in their respective fuzzy sets. In other words, for the input vector,  $X$ , the membership functions  $\mu_i(X)$  must be evaluated. The evaluation of  $\mu_i(X)$  is called fuzzification.

Consider the example shown in Figure 8. Five fuzzy sets are defined in the domain of altitude error – *VeryLow*, *Low*, *JustRight*, *High*, and *VeryHigh*. If the aircraft's current altitude is 65 feet above the desired altitude, the degree of membership in the sets *JustRight* and *High* are 0.67 and 0.33, respectively. From the Bayesian perspective, the membership functions correspond to subjective, conditional probabilities. Then the probability of the aircraft being in the mode, *JustRight* is 0.67, or  $P(\text{JustRight} \mid \epsilon_{\text{alt}}) = 0.67$ . At the same time,  $P(\text{High} \mid \epsilon_{\text{alt}}) = 0.33$ . These are measures of how closely the given altitude matches each mode.

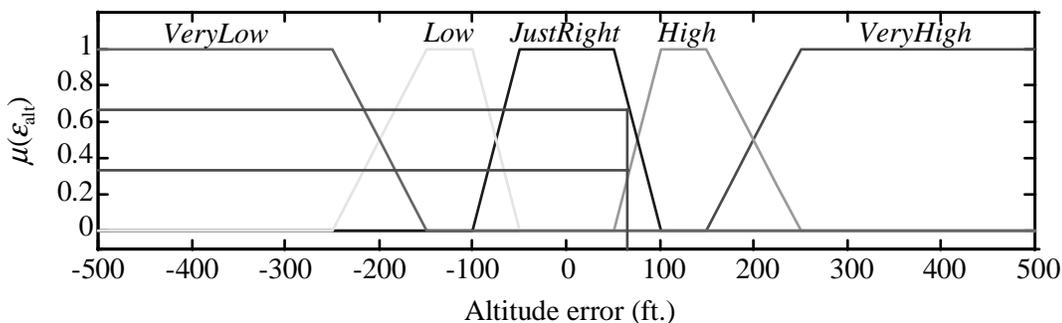


Figure 8. Example of fuzzification of altitude error into modes *JustRight* and *High*

### ***Fuzzy Rule Base***

Fuzzification prepares crisp numerical data for input to the rule base. The rule base is the most important part of any fuzzy system. The rule base encodes the knowledge of the expert into if/then rules of the form

*IF condition THEN result*

where the *condition* is called the antecedent and the *result* is called the consequent. Both the antecedent and the consequent are logical propositions of the form

$$domain_i \text{ is fuzzy set }_j$$

where *fuzzy set*<sub>*j*</sub> is defined in the domain *domain*<sub>*i*</sub> using one of the membership functions discussed earlier. In the aircraft auto-pilot example, a sample rule could be

$$\text{IF } Altitude \text{ is } High, \text{ THEN } ElevatorAngle \text{ is } NegativeSmall .$$

Generally, the antecedent contains conditions involving more than one domain, such that the rules are of the form

$$\text{IF } domain_i \text{ is fuzzy set }_j \text{ AND } domain_m \text{ is fuzzy set }_n, \text{ THEN } result .$$

In this case, the AND operator is used to combine two conditional statements into one. The method of the combination is according to the definition of the fuzzy AND operator as defined in equation (1). When more than one condition is placed in the antecedent of an if/then rule, the overall effect is to generate a new, multidimensional membership function in the product space of the propositions' domains. If the inference is performed using the probabilistic intersection of the inputs' sets, then that multidimensional region in state space is defined by

$$\mu(x_1, x_2, \dots, x_N) = \prod_{i=1}^N \mu_i(x_i) \quad (4)$$

The number of rules needed in a fuzzy system depends on the complexity of the desired mapping, the number of inputs, and the number of fuzzy sets defined on each input domain. The size of the rule base is a major problem for complex fuzzy systems. For a small two-input, one output system, with the two input domains partitioned into five fuzzy sets, the number of rules needed to completely cover the input space is 25. If rules are needed for every combination of sets defined in the domains of the inputs, the number of rules in the rule base grows exponentially according to the following equation:

$$\text{Number of rules} = M \prod_{i=1}^N n_i \quad (5)$$

where  $M$ , the number of outputs  
 $N$ , the number of inputs  
 $n_i$ , the number of sets in each domain

The complexity of the rule base is a major area of research for fuzzy systems.

### **Implication**

Once the antecedent is evaluated, its “truth value”, or “degree of certainty”, is associated with the proposition of the consequent. If the fuzzy system includes defuzzification, the method of implication must be considered. Implication is a function of the antecedent’s “truth value” and the output’s membership function. Consider the rule IF  $x$  is  $A$  THEN  $y$  is  $B$  which has been evaluated and found to have a “truth value” of  $\beta = \mu_A(x)$ . The implication can be expressed as  $I(\beta, \mu_B(y))$ . The implication function is generally defined in one of two ways.

$$\begin{aligned} \text{Scaling: } \mu'_B(y) &= I(\beta, \mu_B(y)) = \beta \cdot \mu_B(y) \\ \text{Clipping: } \mu'_B(y) &= I(\beta, \mu_B(y)) = \min(\beta, \mu_B(y)) \end{aligned} \quad (6)$$

The product version of the implication maintains consistency with a probabilistic view of fuzzy logic. Figure 9 shows the effect of the two different methods of implication on the output membership functions.

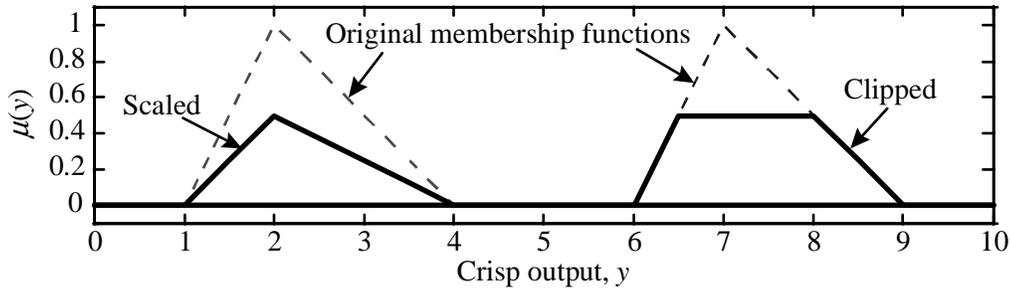


Figure 9. Two methods of fuzzy implication.

### **Aggregation**

Another function that must be performed in the rule base is aggregation. Aggregation determines how to combine the outputs of different rules which refer to the same fuzzy sets in their consequence. For example, suppose that the following two rules have antecedents that evaluate to  $\beta_1 = \mu_A(x)$  and  $\beta_2 = \mu_B(x)$ :

IF  $x$  is  $A$ , THEN  $y$  is  $C$   
IF  $x$  is  $B$ , THEN  $y$  is  $D$

What then is the membership function for the output,  $y$ ? Again, two methods are commonly used for aggregation. One results from a probabilistic view of fuzzy logic. The other is based on the clipping max and min operators. For the two consequents  $C$  and  $D$ , which have “truth values” (through an implication operation) of  $\mu_C(y)$  and  $\mu_D(y)$ , the aggregation can be defined as

$$\begin{aligned} A(\mu_C(y), \mu_D(y)) &= \mu_C(y) + \mu_D(y) - \mu_C(y) \cdot \mu_D(y) \\ A(\mu_C(y), \mu_D(y)) &= \max(\mu_C(y), \mu_D(y)) \end{aligned} \quad (7)$$

and is illustrated in Figure 10.

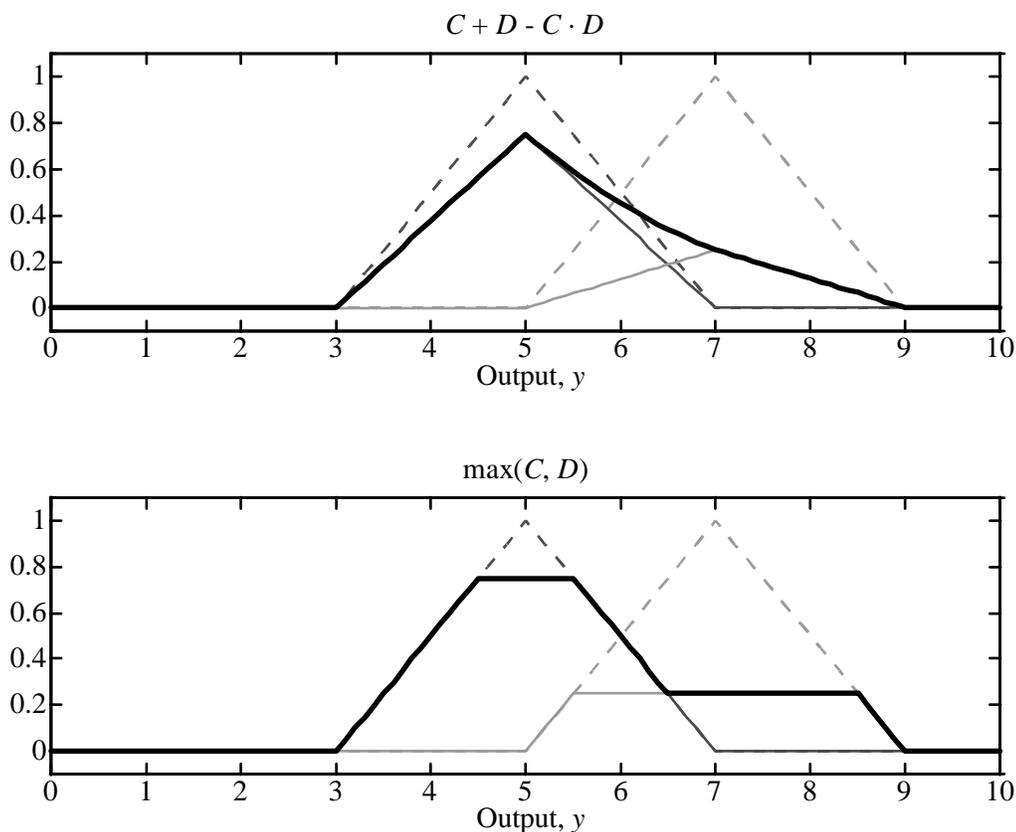


Figure 10. Two definitions for the aggregation of rule consequents.

### ***Defuzzification***

If the output of the fuzzy system must be a crisp value, one final step is needed – defuzzification. Like so many areas in fuzzy logic system design, the best choice for the

defuzzification method is not universally agreed upon by fuzzy system engineers. The two most common methods are centroid and mean of maxima.

The centroid method of defuzzification is calculated by

$$\text{centroid}(C') = \frac{\int_y y \cdot \mu_{C'}(y) dy}{\int_y \mu_{C'}(y) dy} \tag{8}$$

The mean of maxima method determines the average of the crisp points that maximize the output membership function. Mathematically, the mean of maxima method can be stated

$$\text{MOM}(C') = \int_{Y'} y \cdot \mu_{C'}(y) dy$$

$$\text{where } Y' = \left\{ y: \mu_{C'}(y) = \max_y(\mu_{C'}(y)) \right\} \tag{9}$$

Figure 11 shows examples of these two types of defuzzification.

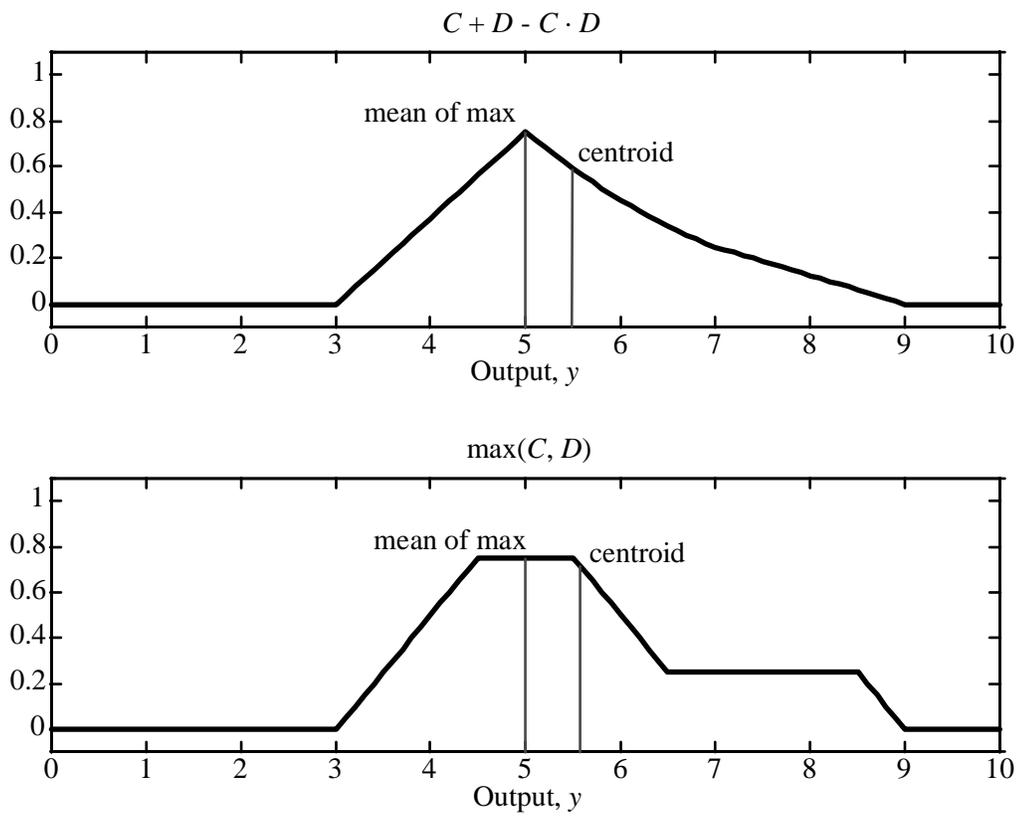


Figure 11. Two methods of defuzzification – centroid and mean of maxima.

## BAYESIAN PROBABILITY

Zadeh's philosophy of fuzzy sets is by far the most popular perspective in the fuzzy logic community. However, there is a growing popularity for the perspective that views fuzzy membership functions as conditional probability functions. Painter [28] has shown that a common implementation of fuzzy logic can be formulated in Bayes notation. He further concludes that "fuzzy control may be easily understood, and therefore pragmatically applied, in terms of decision and averaging concepts familiar in the Bayesian signal processing and control world."

At the heart of the fuzziness vs. probability issue is the use of the Bayesian subjective interpretation of probability, as opposed to the frequentist interpretation of probability based on sample sets. Instead of stating that a specific man of height  $x$  is *tall* to the degree of 0.75, a Bayesian statistician would state the  $P(A|x) = 0.75$ . Here  $A$  is the proposition, "the man is tall."

In light of this argument, many statisticians and some engineers are of the opinion that fuzzy set theory is nothing more than an ad hoc excursion away from well formulated theories of uncertainty [24]. Peter Cheeseman, a researcher at NASA Ames Research Center is among the more vocal. See [5] and [6]. In [6], Cheeseman states that "FST is not a theoretical advance." He suggests that the error in fuzzy set theory is in insisting on a dichotomy of sets when such a dichotomy is meaningless. Cheeseman further explains,

The amount of liquid in a cup, a person's height, the fullness of someone's beard, and so forth are all continuously varying quantities, so to try to describe them with a binary (true/false) predicate is an error. A cup can be stated to have any degree of fullness (fraction of volume occupied by liquid) as a simple assertion of degree, without having to invent a whole new concept (FST) to do so.

Fuzzy set theory may not be the theoretical advancement some would like to believe. It is an engineering advancement, however. It works, and many engineers have found it to be a fast and easy way to implement knowledge-based systems. The questions of its originality or uniqueness are moot when the issues are design time and applicability. The exponential growth in the number of uses of fuzzy logic in real industrial applications is testimony to fuzzy logic's value to the engineering community.

While philosophical debates are not what most practitioners need, for theorists the Bayesian interpretation offers an important guide to understanding and extending fuzzy set theory. The rich and proven background of Bayesian statistics is a measure by which to judge many of the ad-hoc

techniques that abound in fuzzy logic literature. The view of membership functions as Bayesian conditional probabilities have guided the work set forth in this dissertation.

### THE BAYES/FUZZY ISOMORPHISM

The mathematical similarities between Bayesian probability and fuzzy logic is well illustrated with an example from control theory. Consider a set of mutually exclusive events,  $E_1, E_2, \dots, E_N$  which forms a partitioning of the space of all possible events. That is,

$$\begin{aligned} E_1 \cup E_2 \cup \dots \cup E_N &= S \\ P(S) &= 1 \\ E_1 \cap E_2 \cap \dots \cap E_N &= \phi \\ P(\phi) &= 0 \end{aligned} \quad (10)$$

Each event has an associated a priori probability,  $P(E_i)$ . Due to the fact that the events are mutually exclusive and form a complete partitioning of the event space, the probabilities must sum to unity.

$$P(E_1) + P(E_2) + \dots + P(E_N) = P(S) = 1 \quad (11)$$

The variable  $x$  is a value of the random variable  $X$  and is taken to be a sensor variable useful for making decisions on the occurrence of the events  $E_1, \dots, E_N$ . Conditional probability density functions of the form  $f_X(x|E_i)$  exist which model the probability of  $x$  given the occurrence of an event,  $E_i$ .

The variable  $z$  is a value of a random variable  $Z$  describing a control action to be taken. The control action is chosen based on the determination of event  $E_i$ . The conditional probability density function for the control variable  $z$ , given observation  $x$ , is

$$f_{Z|X}(z|x) = \frac{\sum_i f_{Z|X}(z|x, E_i) \cdot f_X(x|E_i) \cdot P(E_i)}{\sum_j f_X(x|E_j) \cdot P(E_j)} \quad (12)$$

which is shown from first principles in Appendix A. The unique control value  $\hat{z}$  is obtained by taking the conditional mean.

$$\hat{z} = E\{z|x\} = \int_{-\infty}^{\infty} z \cdot f_{Z|X}(z|x) dz \quad (13)$$

In order to facilitate the comparison of probabilistic control and fuzzy control, equation (12) can be rewritten as

$$f_{z|x}(z|x) = \sum_i f_{z|x}(z|x, E_i) \cdot G_i(x) \quad (14)$$

$$\begin{aligned} \text{where } G_i(x) &= \frac{f_x(x|E_i) \cdot P(E_i)}{\sum_j f_x(x|E_j) \cdot P(E_j)} \\ &= \frac{f_x(x|E_i) \cdot P(E_i)}{f_x(x)} \end{aligned} \quad (15)$$

Notice that (15) takes the form  $\frac{a}{a+b}$ , which yields the following characteristics.

$$\begin{aligned} G_i(x) &\in [0, 1] \\ \sum_i G_i(x) &= 1 \end{aligned} \quad (16)$$

These characteristics suggest that  $G_i(x)$  may be a probability. Indeed, according to Bayes' Theorem, equation (15) can be written

$$\frac{f_x(x|E_i) \cdot P(E_i)}{f_x(x)} = P(E_i|x) \quad (17)$$

showing that  $G_i(x)$  is the conditional posteriori probability of  $E_i$  given  $x$ .

$$G_i(x) = P(E_i|x) \quad (18)$$

This is the conditional probability function which is useful for implementing "minimum probability of error" decision strategies [16]. However, here a hard decision is not made. The a posteriori probabilities are used to weight the conditional control densities, to yield an ensemble-averaged control density.

$$f_{z|x}(z|x) = \sum_i f_{zx}(z|x, E_i) \cdot P(E_i|x) \quad (19)$$

Computing  $f_{zx}(z|x, E_i)$  in equation (19) is not attractive, since in the simplest case,  $f_{zx}(z, x, E_i)$  is a three-dimensional density. If  $x$  is a vector of sensor inputs, the problem becomes even more complex. A more desirable strategy for computing the control density is one based on  $f_z(z|E_i)$ .

What assumptions must be made to ignore the conditioning on  $x$  in  $f_{z|x}(z|x, E_i)$ ? Since  $E_i$  is an event whose occurrence is modeled on the values of  $x$ , is the conditioning on  $x$  redundant? The question is,

$$f_{z|x}(z|x, E_i) \stackrel{?}{=} f_z(z|E_i). \quad (20)$$

From first principles it can be seen that this is the same as asking,

$$f_{zx}(z, x|E_i) \stackrel{?}{=} f_z(z|E_i) \cdot f_x(x|E_i) \quad (21)$$

or in English, “Given  $E_i$ , are  $z$  and  $x$  independent?” The answer is the following:

Given  $E_i$ , the a priori decision density,  $f_x(x|E_i)$ , is uniquely determined without recourse to  $z$ . It is modeled from the decision problem on  $x$  and  $E_i$ . Likewise, given  $E_i$ , the control density,  $f_z(z|E_i)$ , is uniquely determined without recourse to  $x$ . It is also a model, from the control problem on  $z$  and  $E_i$ . Therefore, since the two conditional densities are uniquely determinable without recourse to the other’s variable, the variables are conditionally independent. Their independence flows from the separateness of the decision and control problems. The advantage of recognizing this independence is a greatly simplified control density calculation.

$$f_{z|x}(z|x) = \sum_i f_z(z|E_i) \cdot P(E_i|x) \quad (22)$$

The preceding mathematical development was completed entirely based on first principles of probability theory. However, the similarities between this control method and a fuzzy logic controller should be readily apparent to anyone familiar with the field of fuzzy control. For example, the events  $E_1, \dots, E_N$  correspond to the input fuzzy sets. The calculation of  $P(E_i|x)$  may be interpreted as a “fuzzification stage.” The conditional densities  $f_z(z|E_i)$  are similar to output membership functions and the calculation of the expected value,  $E\{z|x\}$ , is isomorphic to what is known as the “centroid method.” Figure 12 shows a comparison of the two types of controllers.

While the fuzzy controller relies on subjective membership functions and heuristic rules, the Bayes controller relies on statistical densities and conditional probabilities, drawn from the well-founded and consistent Bayes Boolean algebra.

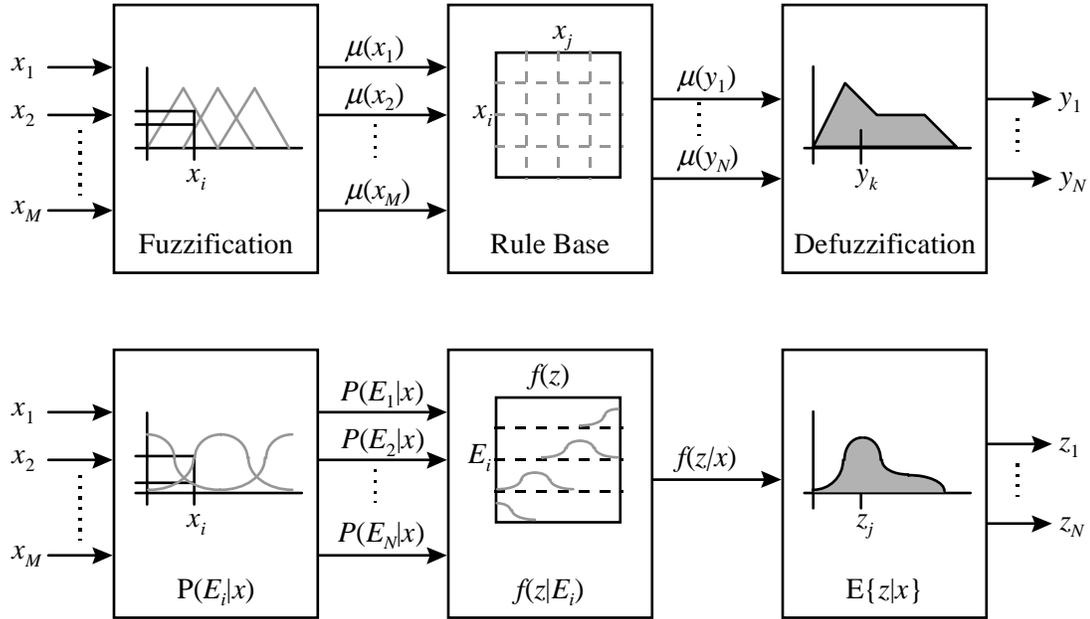


Figure 12. Comparison of fuzzy and Bayesian controllers.

## DEALING WITH UNCERTAINTY

Uncertainty is inherent in the real-world problems that engineers face. Fuzzy decision theory attempts to tackle the problem of reasoning under uncertainty. Designing systems that can accomplish a control objective or provide meaningful support to its users becomes a greater challenge when sensors can be noisy or can malfunction, when operational modes are ambiguous, or when the intent of the user must be determined. The previous sections provided the mathematical basis for fuzzy decision theory. Here, various techniques for dealing with uncertainty are discussed in the context of fuzzy decision theory.

On the most basic level, uncertainty in fuzzy systems is modeled by the fuzzy regions that exist between the fuzzy sets defined by a rule base. These fuzzy regions model the gradual transitions that exist between what would otherwise be crisp sets. Fuzzy logic has demonstrated its usefulness in dealing with uncertainty by partitioning an operational space into sets with gradual transitions.

On a higher level, uncertainty can also be modeled by modifying the interpretation of the rules of a fuzzy rule base. Such is the case for the Flight Mode Interpreter of the *ASTRA* project, which is discussed in Chapter IV. For this complex problem, several mechanisms for dealing with

uncertainty in the rule base have been considered. For example, consider the problem of defining a region in a state space of high dimensionality. The fuzzy rule is of the form

IF  $domain_i$  is *fuzzy set* <sub>$i$</sub>  AND....AND  $domain_N$  is *fuzzy set* <sub>$N$</sub> , THEN *result* .

Mathematically, this rule is evaluated using equation (23):

$$\mu(x_1, x_2, \dots, x_N) = \prod_{i=1}^N \mu_i(x_i) \quad (23)$$

The antecedent defines a fuzzy set in the state space of  $N$  domains. The result can be a conclusion, or decision, or control action. Suppose, however, that the variable of domain  $i$  does not fall into the region covered by fuzzy set  $i$ . In this case, the consequence will *not* be concluded, or decided, or acted upon because one of the conditions of the antecedent is not met and the product of equation (23) evaluates to zero. This is acceptable if the state space is entirely covered by other rules in the fuzzy rule base. However, in systems with high dimensionality, covering the entire state space is impractical.

In the flight mode interpretation problem, several inputs are used to interpret the current operational mode of the aircraft. If one of the inputs does not match a particular mode, but all the other inputs do, the Flight Mode Interpreter must still be able to make a decision for that mode. The one input that does not match may reflect differences in pilot technique, a pilot's error, a malfunctioning sensor, or poorly defined membership functions. The term "anomaly" has been useful in describing these situations.

Figure 13 shows a three-dimensional state space and a region defined by three one-dimensional membership functions. For simplicity the one-dimensional membership functions are crisp. Because they are crisp, the diagram is general for both fuzzy min and probabilistic product. If any of the three variables fall outside the membership functions of its domain, the system is deemed completely out of the partition shown. In other words, the possibility of an anomaly is not considered.

In real world applications, like the Flight Mode Interpreter, the possibility for anomalous conditions are real and problematic. Several ways of preventing anomalous inputs from zeroing the inferences have been considered during the development of the Flight Mode Interpreter. The following sections explain the benefits and drawbacks of each.

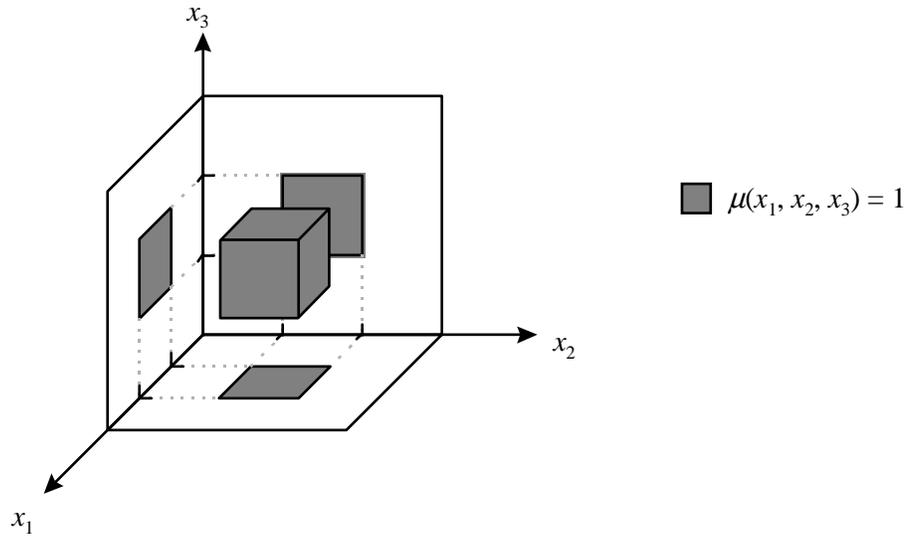


Figure 13. The intersection of membership functions in a three-dimensional state space.

### ***Probabilistic Union***

The first proposed method for dealing with anomalous inputs in fuzzy systems is to use probabilistic union (or fuzzy max) when evaluating the rules. This is equivalent to interpreting the rules as

IF  $domain_i$  is fuzzy set $_i$  OR...OR  $domain_m$  is fuzzy set $_m$ , THEN result .

In this case, if *any* of the domains match their respective fuzzy set, the result is concluded. Mathematically, the inference can be expressed as

$$\mu(x_1, x_2, \dots, x_N) = 1 - \prod_{i=1}^N (1 - \mu_i(x_i)) \quad (24)$$

The partition that this method builds is quite different from probabilistic intersection of Figure 13. Figure 14 shows the state space partition resulting from interpreting the rules using probabilistic union. While this scheme allows for the presence of anomalies during inference, it overextends the region being modeled and is therefore not an acceptable method.

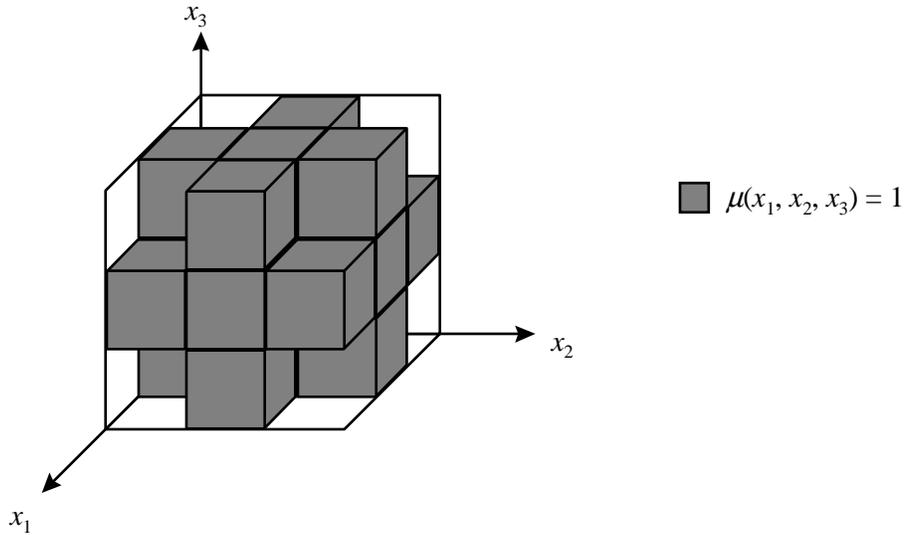


Figure 14. The union of membership functions in a three-dimensional space.

### ***Anomaly Threshold***

The method originally proposed and implemented by Economides [9] to deal with anomalies is the setting of a threshold of membership values on all the inputs. That is, if the degree of membership of a variable in a particular domain's fuzzy set is below a certain threshold,  $\tau$ , do not allow that variable to unduly eliminate the decision for that mode. Rather, reduce the degree of membership of that inference by multiplying by the threshold. Mathematically, the rule of inference changes from equation (23) to

$$\mu(x_1, x_2, \dots, x_N) = \prod_{i=1}^N \max(\mu_i(x_i), \tau). \quad (25)$$

This is equivalent to limiting the minimum value of a membership function to  $\tau$ , as shown in Figure 15. The effect on the modeled state space is shown in Figure 16. Notice that the state space is completely covered such that any input would have at least a degree of membership of  $\tau^3$  in the modeled region.

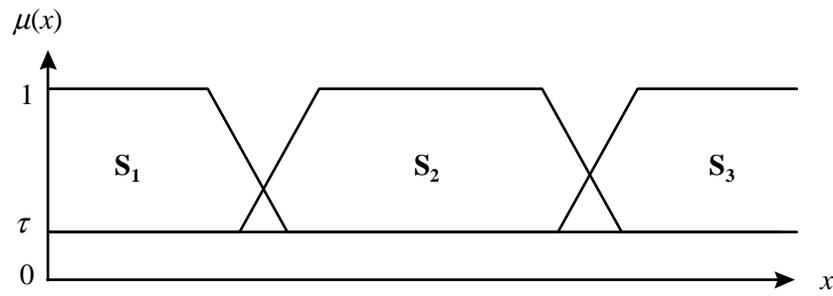


Figure 15. Effect on membership functions when setting a threshold.

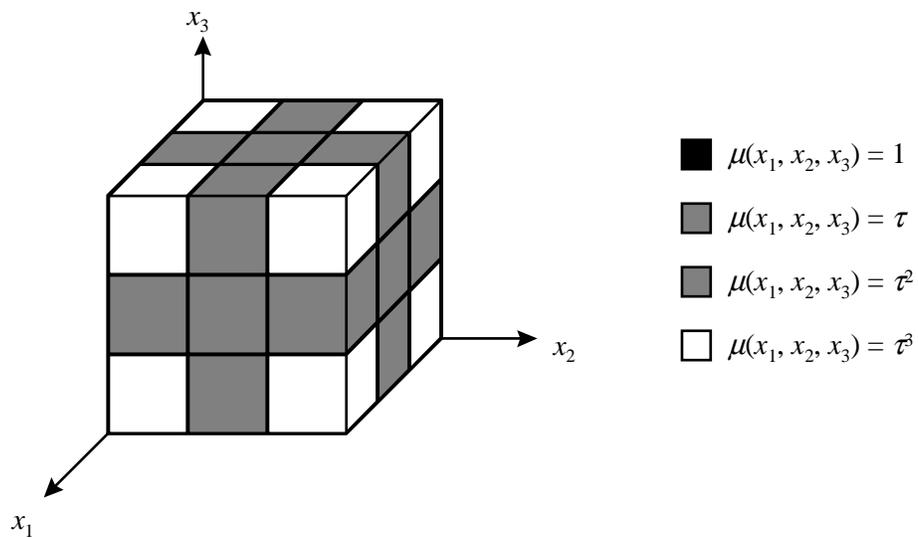


Figure 16. The modeled region in state space when using a threshold.

### *Normalized Sum*

The author proposes a third method for addressing the issue of anomalous inputs. The use of a normalized sum is an intuitive method for calculating the certainty associated with a fuzzy decision. The normalized sum is calculated as shown in equation (26).

$$\mu(x_1, x_2, \dots, x_N) = \frac{1}{N} \sum_{i=1}^N \mu_i(x_i) \quad (26)$$

This method is not based on the probabilistic union of the fuzzy sets. Rather, the summation reflects an increasing certainty as more evidence supporting a particular decision is evaluated. The normalized sum more closely resembles the anomaly threshold method, as shown in Figure 17.

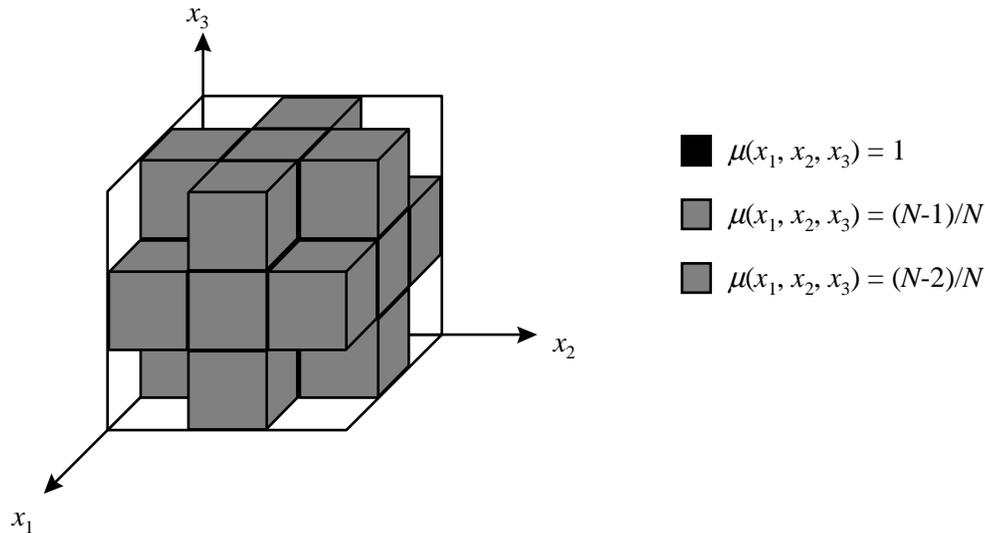


Figure 17. The modeled region in state space when using a normalized sum.

Figure 13 - Figure 17 illustrate the multidimensional regions generated by modeling uncertainty in the fuzzy rules using the probabilistic intersection, the probabilistic union, an anomaly threshold, and a normalized sum. For simplicity of illustration, the regions were generated assuming non-fuzzy intervals on the three input domains. Another important comparison reveals differences when the inputs are fuzzy. Figure 18 shows four plots of  $\mu(x_1, x_2)$  as functions of  $\mu(x_1)$  and  $\mu(x_2)$  using the four different methods outlined above.

The first plot, labeled “probabilistic intersection,” reveals the basic problem of ignoring the possibility of anomalies. If either of the membership functions  $\mu(x_1)$  or  $\mu(x_2)$  evaluate to zero, the entire inference evaluates to zero. Compare this to the “probabilistic union” in which if either of the membership functions  $\mu(x_1)$  or  $\mu(x_2)$  evaluate to one, the entire inference evaluates to one. Neither are an acceptable inference method if anomalies are anticipated, especially as the number of inputs increases. Setting a threshold on the degree of memberships is an improvement. However, notice the regions in which  $\mu_i(x_i) < \tau$ . In these regions, a change in input does not always translate into a

change in the certainty of the inference. The inference is constant even as the degree of certainty on the input increases or decreases.

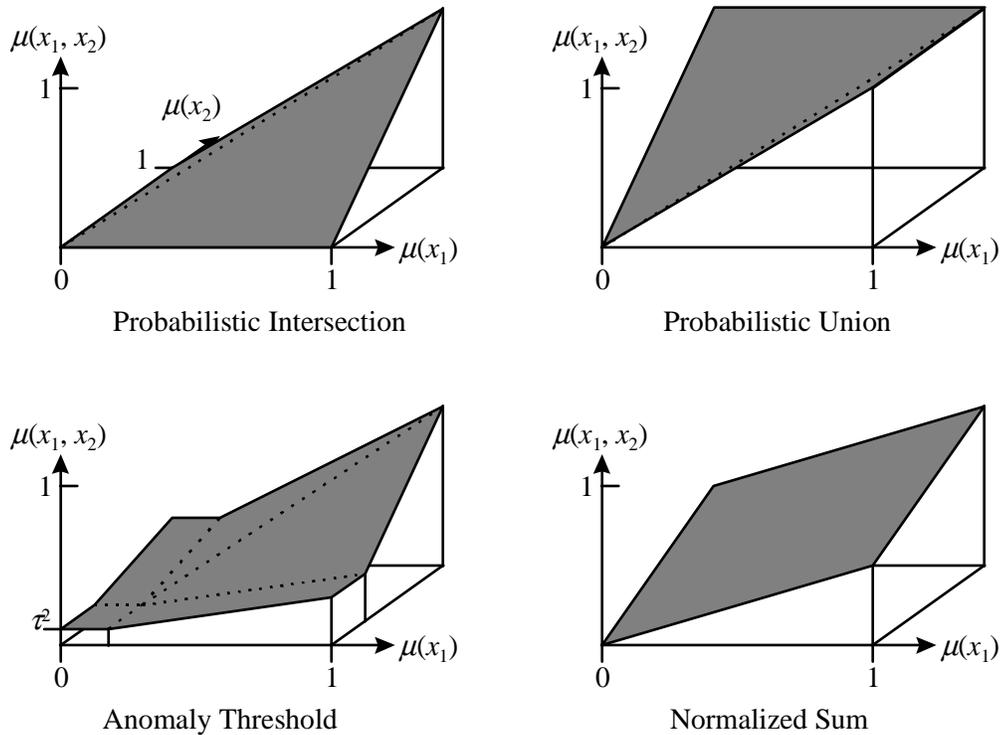


Figure 18. The effect of the inference method in the fuzzy regions of the state space.

The fourth plot of Figure 18, labeled “normalized sum,” does not exhibit any of these negative characteristics. This method of inference prevents anomalies from “zeroing out” an inference, unlike a straight probabilistic intersection. Notice how the edges of the surface always decrease for a decreasing  $\mu_i(x_i)$ . That is, monotonicity is preserved. In fact, for all decreasing  $\mu_i(x_i)$ , the inference decreases. For these reasons, the normalized sum is the superior inference method when anomalous inputs are of concern.

### Summary

This chapter included a unified approach to fuzzy decision theory from the perspective of Bayesian probability. The parallels between fuzzy set theory and Bayesian probability theory were discussed and demonstrated with an example. Finally, the three variations the standard inference mechanism were compared on the basis of their robustness to anomalous inputs. The theory

presented in this chapter has proven invaluable in the implementation of the Flight Mode Interpreter. Chapter IV includes an in-depth look at the Flight Mode Interpreter in the context of the fuzzy decision theory which was presented in this chapter. The following chapter explores the issue of dimensionality in fuzzy systems.

## CHAPTER III

### DIMENSIONALITY IN FUZZY SYSTEMS

#### ONE-DIMENSIONAL FUZZY SYSTEMS

The current state of the art in fuzzy system design is the use of one-dimensional fuzzy sets. Each input variable to the fuzzy system is partitioned into fuzzy sets. The fuzzy set definitions on one input are independent of the value of other inputs. A fuzzy rule base combines these one-dimensional fuzzy sets on the input space, and maps them to the system's output space. As the number of inputs increase, the size of the rule base grows exponentially. The use of fuzzy inference for flight mode interpretation has revealed that this standard fuzzy logic approach is insufficient for application in complex systems. To address some fundamental shortcomings in the current state-of-the-art, the author has developed the *hypertrapezoidal fuzzy membership function* (HFMF). This chapter explains the motivation and theory of this new technique. Additionally, system architectures are explored as a method for managing complex fuzzy systems.

#### *Correlation*

Correlation between input variables of a fuzzy system can lead to complications in current fuzzy techniques based on one-dimensional membership functions. By "correlation" is meant the condition that a fuzzy set describing a system state is represented by an irregular, smoothly connected region in a multivariable state space. The "footprint" of such a mode on the  $x$ - $y$  plane could look something like the solid ellipse in Figure 19. One-dimensional membership functions cannot by themselves represent such a relationship. The current practice approximates a smooth representation by composition of two or more single-variable regions. Such a composition is shown in dashed lines on Figure 19. An  $N$ -dimensional fuzzy membership function scheme should be able to account for correlation between the domains that define a set.

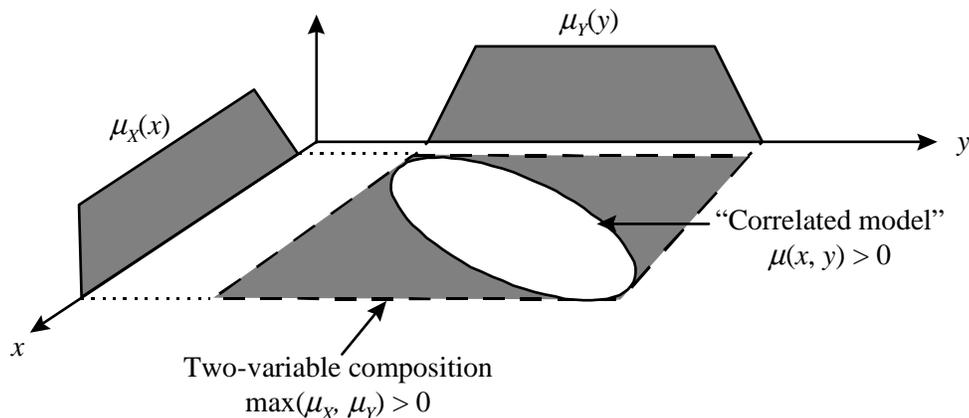


Figure 19. Footprint of fuzzy set when the input variables are correlated.

### ***Rule Base Composition***

The most common method of defining an N-dimensional fuzzy relationship between the variables of a system is using rule base composition. While the N-dimensional fuzzy membership function is not explicitly defined, the effect of fuzzy implication is a mapping from the N-dimensional space onto a degree of membership in a composite set. One-dimensional membership functions are defined on each domain and the multidimensional composition is accomplished through the fuzzy if/then rules.

The obvious advantage that rule base composition has over explicit definition of multidimensional sets is the simplicity with which the one-dimensional membership functions are defined. The visualization and specification in multidimensional space becomes very difficult for  $N > 3$ . The disadvantage then is that only certain basic shapes can result from rule base composition. Namely, the sets defined by if/then rules have rectangular footprints, as illustrated in Figure 19.

There is another disadvantage to rule base composition that may not be immediately noticeable. There is the possibility of unintended “valleys” forming on the state space of the inputs. Consider a simple system with inputs  $x_1$  and  $x_2$  and output  $y$ . Let  $x_1$  and  $x_2$  be partitioned into three fuzzy sets each as shown in Figure 20. Nine rules are required to completely cover the input space. Table 1 lists the rules.

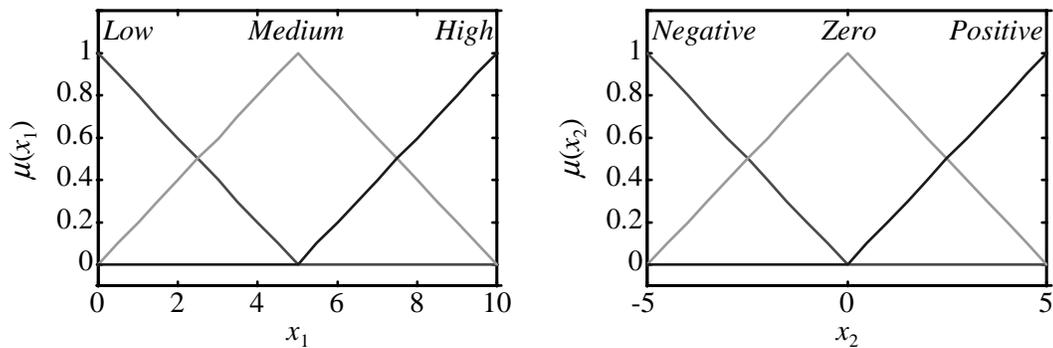


Figure 20. Fuzzy sets of rule base composition example.

Table 1.

Fuzzy rules for a simple example of rule base composition.

IF $x_1$ and $x_2$ THEN $y$	$x_2$ is <i>NEGATIVE</i>	$x_2$ is <i>ZERO</i>	$x_2$ is <i>POSITIVE</i>
$x_1$ is <i>LOW</i>	$y$ is <i>SMALL</i>	$y$ is <i>MEDIUM</i>	$y$ is <i>MEDIUM</i>
$x_1$ is <i>MEDIUM</i>	$y$ is <i>MEDIUM</i>	$y$ is <i>LARGE</i>	$y$ is <i>LARGE</i>
$x_1$ is <i>HIGH</i>	$y$ is <i>SMALL</i>	$y$ is <i>MEDIUM</i>	$y$ is <i>MEDIUM</i>

Of interest is the degree of membership in the output set *LARGE*, as a function of the inputs  $x_1$  and  $x_2$  as shown in Figure 21. Notice that as  $x_1$  is held constant at 5 (approximately *MEDIUM*) and  $x_2$  varies from 0 to 5 (from *ZERO* to *POSITIVE*), the degree of membership in the set *LARGE* “dips”. From the rule base in Table 1, there is no reason for the degree of membership in *LARGE* to decrease in that range. In effect, the desired multidimensional relationship has been inadequately approximated by the rule base composition. It should be noted that the connectives and aggregation methods used in Figure 21 are consistent with the Bayesian perspective of fuzzy logic. For the min/max operators, the “valley” is even more exaggerated.

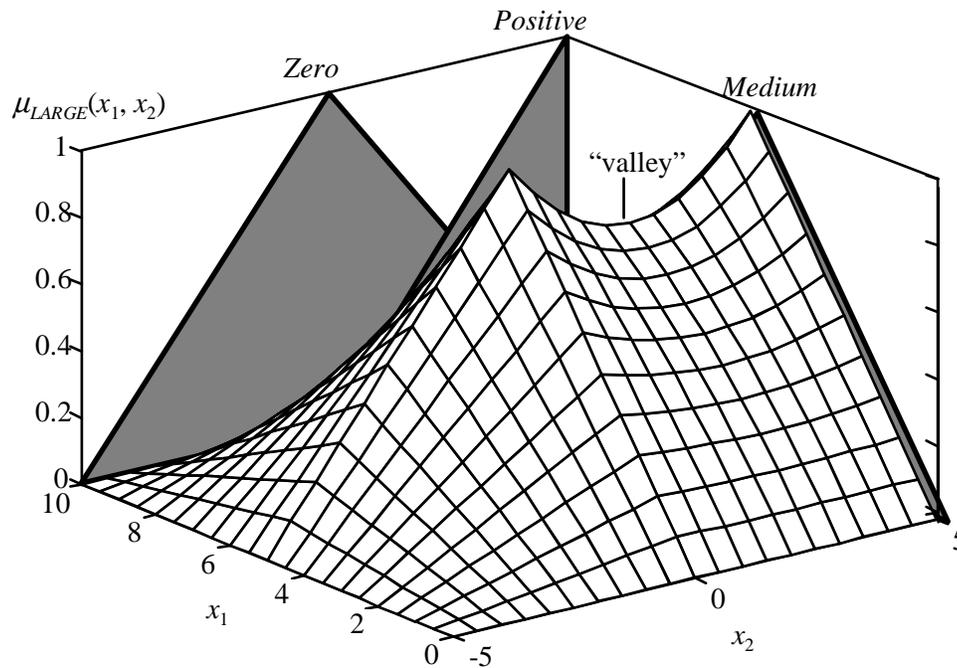


Figure 21. Rule base composition is an approximation of a multidimensional relationship.

### ***Conditional Membership Functions***

For systems with correlation between two variables,  $x_1$  and  $x_2$ , the one-dimensional membership functions describing a fuzzy set  $A$  are of the form  $\mu_A(x_1|x_2)$  and  $\mu_A(x_2|x_1)$ . Correlation of membership can be approximated by designing conditional membership functions. This is a brute force method of specifying  $\mu_A(x_1 | x_2 = X_k)$  for  $k = 1, 2, \dots, K$ . This requires  $K$  one-dimensional fuzzy set definitions for each set or mode. In the case that  $X_j < X^* < X_{j+1}$ , an interpolation must be done to approximate  $\mu_A(x_1 | x_2 = X^*)$ .

This method has been successfully used in a 737 auto-pilot simulation [23]. In [23], Lass's simulated auto-pilot required that the aircraft remain inside a fuzzy operational mode during its descent. The mode was defined on the state space of indicated airspeed and altitude. The desired mode of operation formed a "tunnel" through state space that could not be defined by one-dimensional membership functions. Lass determined that a rule base composition of the desirable operation region would also be impractical. Consequently, Lass defined conditional fuzzy membership functions of altitude for certain intervals of airspeed.

Conditional membership functions can account for correlation in the state space, but lacks efficiency, is arduous to tune, and is not extendible to  $N$  dimensions. For each “corner” in the state space, a pair of membership functions must be designed. With each modification in one plane, the parameters for the other planes must be adjusted. In effect, the tent-shaped membership functions are defined piece by piece, which does not generalize to a state space of more than two variables. A method is needed for designing  $N$ -dimensional membership functions.

### PDF-BASED FUZZY MEMBERSHIP FUNCTIONS

All the previous work in the area of multidimensional membership functions encountered by the author is based on the Gaussian probability density function. The Gaussian PDF is easily extendible to  $N$  dimensions. It can be expressed with one mathematical expression and is differentiable over the entire surface. The general Gaussian membership function is shown in equation (27). The trapezoid, on the other hand, requires a piece-wise definition and is not everywhere differentiable.

$$\mu_A(x) = a_A \exp \left[ -\frac{1}{2} \left( \frac{x - \bar{x}_A}{\sigma} \right)^2 \right] \quad (27)$$

The extension to  $N$  dimensions is accomplished by simply allowing  $x$  in equation (27) to be a vector. Multidimensional Gaussian membership functions have proved especially useful in the areas of clustering [3] and training [40].

One disadvantage to the multidimensional Gaussian membership function is the fact that the membership function evaluates to unity at only a single point in state space. It is the Gaussian equivalent of the triangular membership function. Foster and Khambhampati [10] address this issue by extending the unity point of the Gaussian density along a vector in state space. Figure 22 shows a Gaussian membership function defined on a two-dimensional space, with a vector from (4, 6) to (6, 4) defining the top ridge.

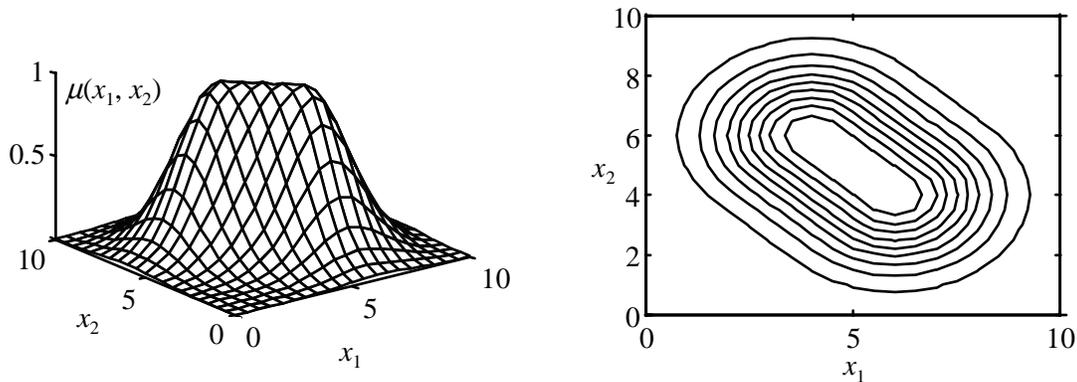


Figure 22. A two-dimensional Gaussian membership function.

## HYPERTRAPEZOIDAL FUZZY MEMBERSHIP FUNCTIONS

### *One-dimensional Trapezoids*

An important consideration in the development of N-dimensional membership functions is that they be specified with only a few parameters. The standard method for defining one-dimensional trapezoidal membership functions is with four points –  $a$ ,  $b$ ,  $c$ , and  $d$ , as shown in Figure 23. This method, however, is impractical for defining membership functions on multiple dimensions.

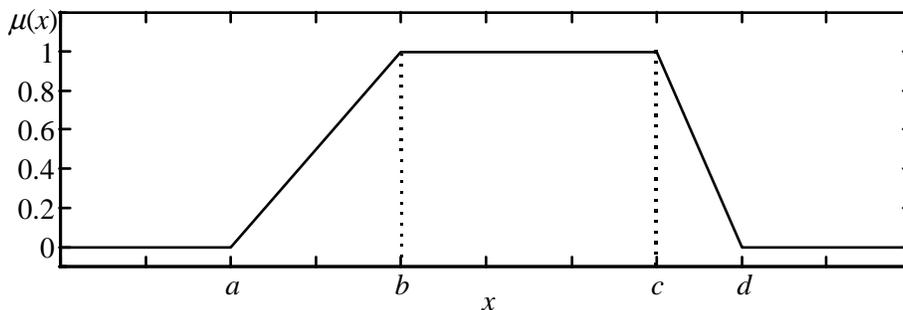


Figure 23. Defining a one-dimensional trapezoidal membership function.

The extension of the trapezoidal membership function into a two-dimensional space would require at least eight points, as shown in Figure 24.

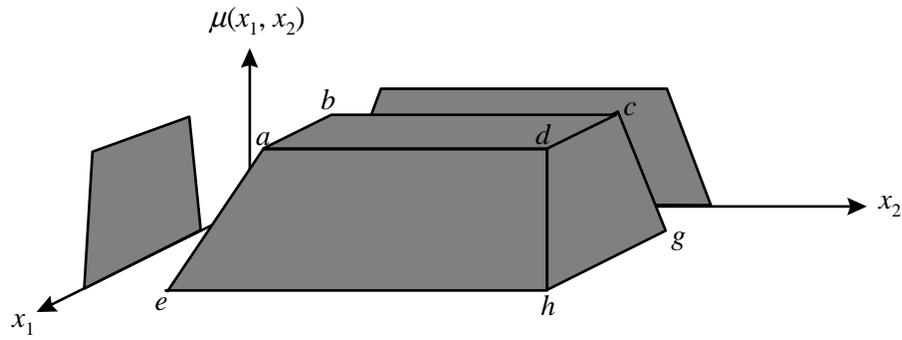


Figure 24. Two-dimensional trapezoidal membership function.

### ***Fuzzy Partitionings***

Another important consideration is that the multidimensional fuzzy sets should enforce the alternate fuzzy logic connectives, originally compared by Bellman and Zadeh [2]. These are the usual connectives of the Bayes version of fuzzy logic [28], wherein the membership values sum to unity. That is, for membership functions  $\mu_i(x)$ ,

$$\sum_i \mu_i(x) = 1 \quad \forall x \quad (28)$$

Membership functions defined in such a manner are referred to as a *fuzzy partitioning*. Fuzzy membership functions based on Gaussian probability density functions can easily be extended to  $N$  dimensions. However, they do not exhibit the desirable property of equation (28). Trapezoidal membership functions, on the other hand, can be defined with the design constraint of equation (28).

### ***Derivation of Hypertrapezoidal Fuzzy Membership Functions***

Based on the requirements outlined above, the author developed a new mechanism for specifying and calculating multidimensional fuzzy membership functions [22]. Termed *hypertrapezoidal fuzzy membership functions*, this new development is a major advancement in the practical application of fuzzy logic to engineering problems.

As an alternative to trying to define all the corners of  $N$ -dimensional fuzzy sets, consider the use of a single point in the state space as the defining parameter of an  $N$ -dimensional fuzzy set. Each fuzzy set in a fuzzy partitioning would then have an associated  $N$ -dimensional vector which is a typical value for that set. The author chose to call such an  $N$ -dimensional vector the *prototype*

point. The prototype point,  $\lambda_i$ , for a fuzzy set,  $S_i$ , with a membership function,  $\mu_i(x)$ , satisfies the following equations.

$$\begin{aligned}\mu_i(\lambda_i) &= 1 \\ \mu_j(\lambda_i) &= 0 \quad j \neq i\end{aligned}\quad (29)$$

Figure 25 shows a simple example of a fuzzy partitioning in two dimensions using three prototype points to define three fuzzy sets.

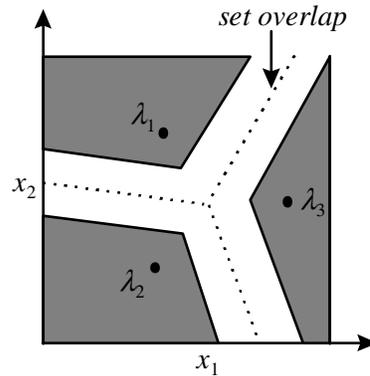


Figure 25. Prototype points defining a fuzzy partitioning.

A measured value,  $x$ , which is an N-dimensional point in the state space of a fuzzy partitioning, has a degree of membership in a fuzzy set based on its Euclidean distance from the prototype point for that set. For example, if  $x = \lambda_1$ , then  $\mu_1(x) = 1$ ,  $\mu_2(x) = 0$ , and  $\mu_3(x) = 0$ . As another example, if  $x$  is equidistant from all three prototype points, then  $\mu_1(x) = 0.333$ ,  $\mu_2(x) = 0.333$ , and  $\mu_3(x) = 0.333$ . This is the basis of hypertrapezoidal fuzzy membership functions and has proven to be quite useful in inferring operational flight modes of an aircraft.

One additional parameter is needed for defining an N-dimensional fuzzy partitioning. The *crispness factor* determines how much overlap exists between the sets of two adjacent prototype points. The author chose to define the range of the crispness factor,  $\sigma$ , to be  $[0, 1]$ . For  $\sigma = 1$ , no overlap exists between the sets, and the partitioning reduces to a minimum distance classifier. Figure 26 shows the resulting partitions of the above example for the two extremes  $\sigma = 0$  and  $\sigma = 1$ .

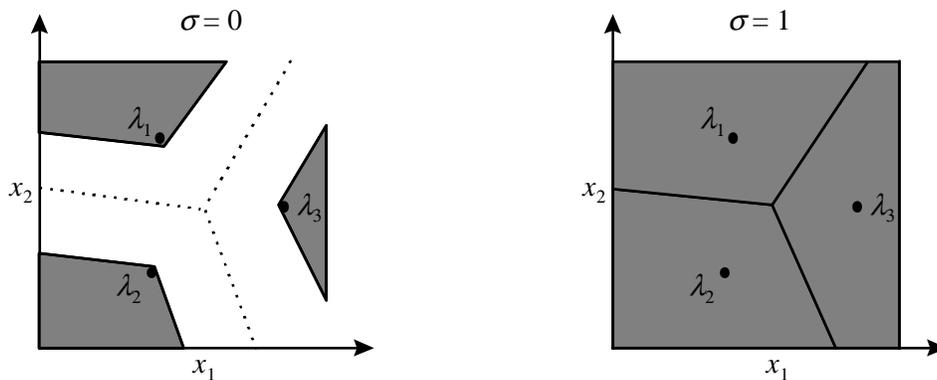


Figure 26. The effect of  $\sigma$  on a two-dimensional fuzzy partitioning.

Such a scheme for defining fuzzy sets can also be used to define standard one-dimensional fuzzy partitions. Figure 27 illustrates how varying  $\sigma$  in a one-dimensional partition evolves the membership functions from triangular fuzzy sets, through trapezoidal fuzzy sets, and finally to crisp, non-fuzzy sets.

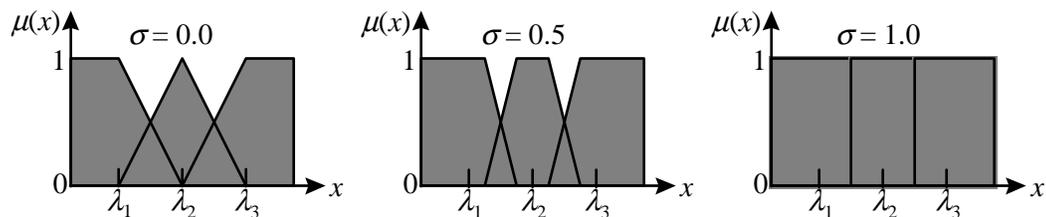


Figure 27. The effect of  $\sigma$  on one-dimensional fuzzy sets.

The author chose to define the crispness factor according to equation (30) and Figure 28.

$$\sigma = \frac{2\alpha}{d} \quad (30)$$

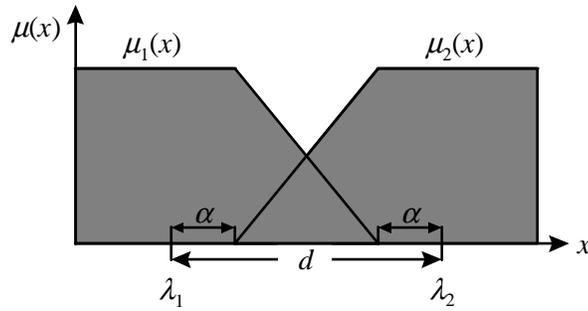


Figure 28. Defining the overlap of a fuzzy partition.

Given a sensor measurement,  $x$ , the HFMFs can now be calculated using standard trigonometry. First, a distance measure,  $\rho_{ij}$ , is calculated for each pair of prototype points, as shown in equation (31). Here,  $d(x,y)$  is the Euclidean distance between  $x$  and  $y$ .

$$\rho_{ij}(x) = \frac{d^2(x, \lambda_i) - d^2(x, \lambda_j)}{d^2(\lambda_i, \lambda_j)} \quad (31)$$

Then the pair-wise membership functions are calculated for each pair of prototype points, as shown in equation (32). Here,  $\vec{v}_{ji}$  is a vector from  $\lambda_j$  to  $\lambda_i$ ,  $\vec{v}_{jx}$  is a vector from  $\lambda_j$  to  $x$ , and  $\vec{v}_{ji} \cdot \vec{v}_{jx}$  is the dot product of the two vectors.

$$\mu_{ij}(x) = \left\{ \begin{array}{ll} 0; & \rho_{ij}(x) \geq 1 - \sigma \\ 1; & \rho_{ij}(x) \leq \sigma - 1 \\ \frac{\vec{v}_{ji} \cdot \vec{v}_{jx} - \frac{\sigma}{2} \cdot d^2(\lambda_j, \lambda_i)}{(1 - \sigma) \cdot d^2(\lambda_j, \lambda_i)}; & \text{otherwise} \end{array} \right\} \quad (32)$$

Finally, the degree of membership,  $\mu_i(x)$ , of measured input,  $x$ , can be determined in one of two ways. The first is based on product inference and is shown in equation (33). The second is based on fuzzy-min inference and is shown in equation (34). Both are normalized such that equation (28) is satisfied. Here,  $M$  is the number of fuzzy sets in the partition.

$$\mu_i(x) = \frac{\prod_{j=1 \neq i}^M \mu_{i|j}(x)}{\sum_{k=1}^M \left( \prod_{j=1 \neq k}^M \mu_{k|j}(x) \right)} \quad (33)$$

$$\mu_i(x) = \frac{\min_j(\mu_{i|j}(x))}{\sum_{k=1}^M \left( \min_j(\mu_{k|j}(x)) \right)} \quad (34)$$

Notice that equations (30) - (34) are general for  $N$  dimensions, including  $N=1$ . These four equations allow for the design of  $N$ -dimensional membership functions using only  $N+1$  parameters. Additionally, the desirable property of equation (28) is enforced.

### *Examples of Hypertrapezoidal Fuzzy Membership Functions*

The following diagrams are examples of fuzzy membership functions designed using the described technique of equations (30) - (34). All the examples use the product inference of equation (33). Figure 29 shows an example of three fuzzy sets defined on two domains. The definition of the three sets is accomplished with the following parameters:  $\lambda_1 = (9, 1)$ ,  $\lambda_2 = (5, 5)$ ,  $\lambda_3 = (1, 9)$ , and  $\sigma = 0.5$ .

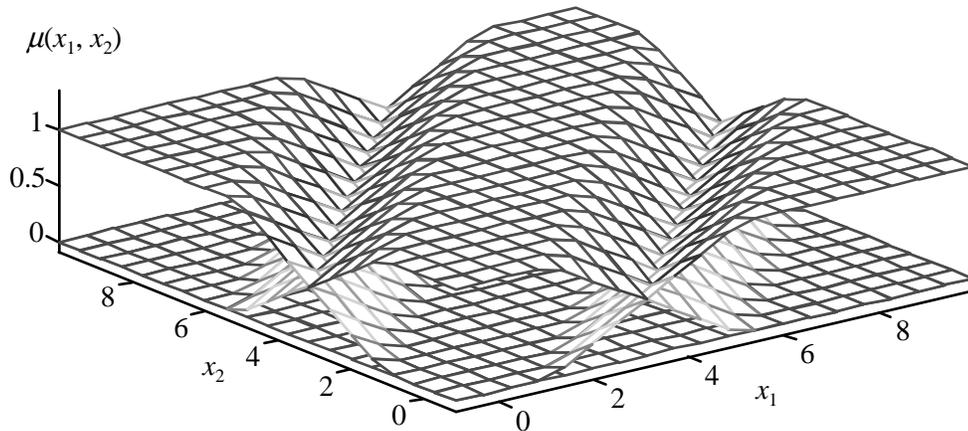


Figure 29. Example of three fuzzy sets defined on a two-dimensional space.

A rule base operating on one-dimensional sets could only approximate the correlation represented in Figure 29. In this case, a transformation of the axes could also compensate for the

correlation. While simple, the example shows how the prototype points and crispness factor define a fuzzy partition. A more complex example is shown in Figure 30, for which a transformation of axes could not compensate for the correlation between the inputs.

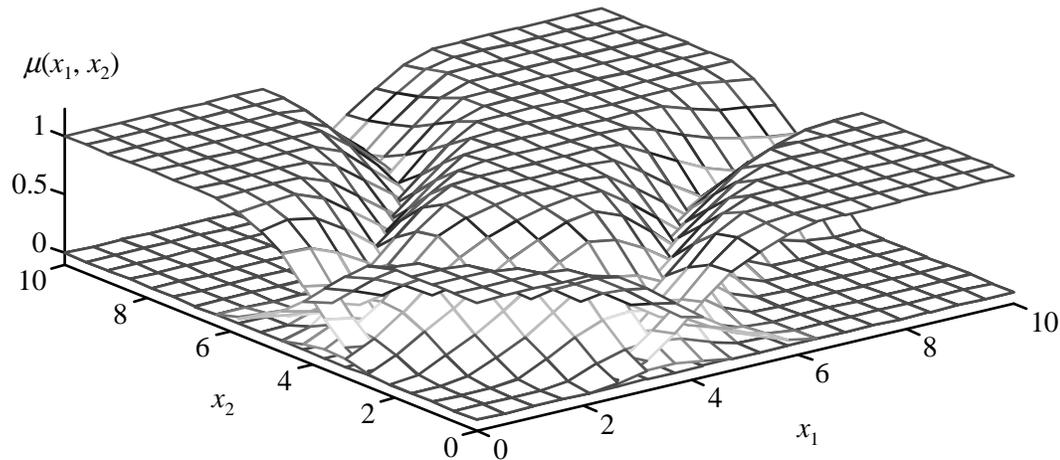


Figure 30. Example of four fuzzy sets defined on a two-dimensional space.

Hypertrapezoidal fuzzy membership functions have proven to be a valuable asset for the flight mode interpretation problem. Detailed examples of multidimensional fuzzy sets designed for flight mode interpretation are shown in Chapter IV. The remainder of this chapter explores other methods for dealing with dimensionality in complex systems.

## **FUZZY SYSTEM ARCHITECTURES**

The following sections document several architectures that can be used to design complex fuzzy systems involving high dimensionality. Practical examples are given in which these architectures have been successfully utilized in industry.

### ***Multilevel Rule Bases***

As mentioned earlier, a major obstacle for designing complex fuzzy systems is the size of the rule base. Multidimensional membership functions decrease rule base size by moving some of the dimensionality out of the rule base and into the fuzzy set definitions. Another technique is to divide the inference stage of a fuzzy system into two or more stages. Multilevel rule bases divide the inference stage either for the purpose of intermediary set composition or for an hierarchical inference strategy.

The *set composition model* of a multilevel rule base is useful for separating a heuristic system identification stage from the control stage of a fuzzy rule base. The first rule base in the set composition model is responsible for determining the qualitative state of the system being observed or controlled. The second rule base uses the inferred system state to determine the desired action or control. Figure 31 shows the general structure for the set composition model of a multilevel rule base.

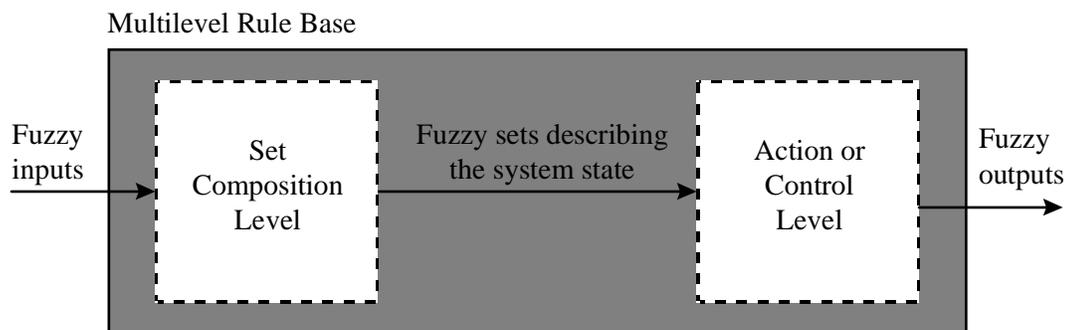


Figure 31. General architecture for set composition model.

This approach for dividing the fuzzy rules into two manageable modules has proven useful in the area of general aviation pilot advisement. The *ASTRA* project seeks to improve situational awareness of the pilots in small aircraft. Improved situational awareness should lead to increased safety in high workload situations. Feedback to the pilot about the qualitative state of the aircraft can be provided through the use of a multilevel fuzzy rule base.

The *ASTRA* fuzzy inference scheme is a good example of the set composition model for a multilevel rule base. The fuzzy inputs to the *ASTRA* inference module are the degrees of membership in fuzzy modes defined on the state space of several flight variables. These variables include indicated airspeed, angle of attack, altitude, etc. The output of the first rule base level is a qualitative characterization of the operational mode of the aircraft. The operational mode of the aircraft is the intermediary set which indicates not only the stage of the flight (i.e. *cruise*, *initial approach*, etc.) but also any anomalies in the aircraft state (i.e. incorrect flap setting, unusual altitude or speed, etc.). Thus, situation recognition is separated from, and subsequently becomes the input to, generating the advice to the pilot.

The *hierarchical model* of a multilevel rule base is useful for separating different levels of goals. For example, the high-level planning that goes into determining the desired position for a

robotic arm could be separated from the mid-level goal of determining a feasible trajectory to get it there. A third level could even calculate the necessary control inputs to the joint motors, a low-level task. As shown in Figure 32, the hierarchical model divides the overall inference process into manageable sub-goals.

The large number of applications of “supervisory servo control” would fall into this category. See for example [25] and [32]. Supervisor servo control is a two-level hierarchical control architecture. The low-level functional block is a traditional numerical controller. The high-level functional block is a rule-based controller that compensates for nonlinearities in the system, or different operational modes of the system. The supervisory level may adjust parameters in the low-level controller, or manipulate reference points.

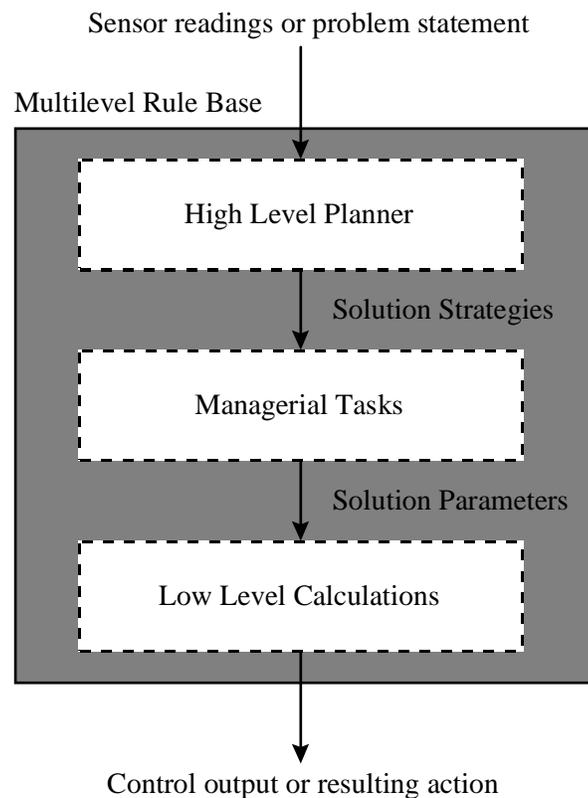


Figure 32. General structure of the hierarchical model.

A hierarchical architecture leads to advantages in both the area of training and hybrid combinations of intelligent control techniques. Separating the sub-goals of a knowledge-based system allows the designer to tailor the subsystems to better solve the unique problem of each level.

Nowhere has the hierarchical model demonstrated its usefulness more than in the area of robotics. As an example, Fukuda and Shibata propose a hierarchical control system for intelligent robotics and mechatronics [11]. The proposed system has three levels. The highest level is a fuzzy inference mechanism that “manipulates symbols to reason logically for control strategies.” The second level uses a fuzzy neural network to produce control references based on the high level control strategy. The final level is a neural network servo controller which provides inputs to the robot motors based on the computed control references of level two. The authors show that the hierarchical nature of such a system architecture allows hybrid combinations of AI, fuzzy, neural, and genetic techniques to take advantage of their individual strengths.

An useful observation was made by Cleveland and Meystel [7] in their design of a hierarchical fuzzy system for an industrial sprayer. They comment that, at lower levels, a higher resolution of data is needed. These different requirements for data resolution translated into different sampling rates. The highest level in their architecture performed heuristic searches at the rate of 1 Hz, while a PID controller in the lowest level operated at 1000 Hz. Intermediary fuzzy logic controllers had a sampling rate of 100 Hz. Such an arrangement also prevents the supervisory levels from responding to high frequency sensor noise.

### ***Parallel Fuzzy Systems***

Like multilevel rule bases, parallel fuzzy systems can be subdivided into two groups – the *competitive model* and the *cooperative model*. Both offer a way to divide a potentially unmanageable rule base into manageable sub-components.

The *competitive model* of parallel fuzzy systems relies on two or more fuzzy systems operating in the same domain, as illustrated in Figure 33. They would generally have the same inputs, yet produce different outputs. Their different outputs might reflect opposing goals or different assumptions used to build their respective rule bases. A means of resolving the conflicting recommendations of the parallel expert systems is needed. Sometimes the respective outputs can simply be averaged. In other applications, a more sophisticated mechanism for choosing or resolving the competitive results may be required. The competitive model of parallel expert systems has been used extensively at the Knowledge Based Signal Processing Lab of Texas A&M University.

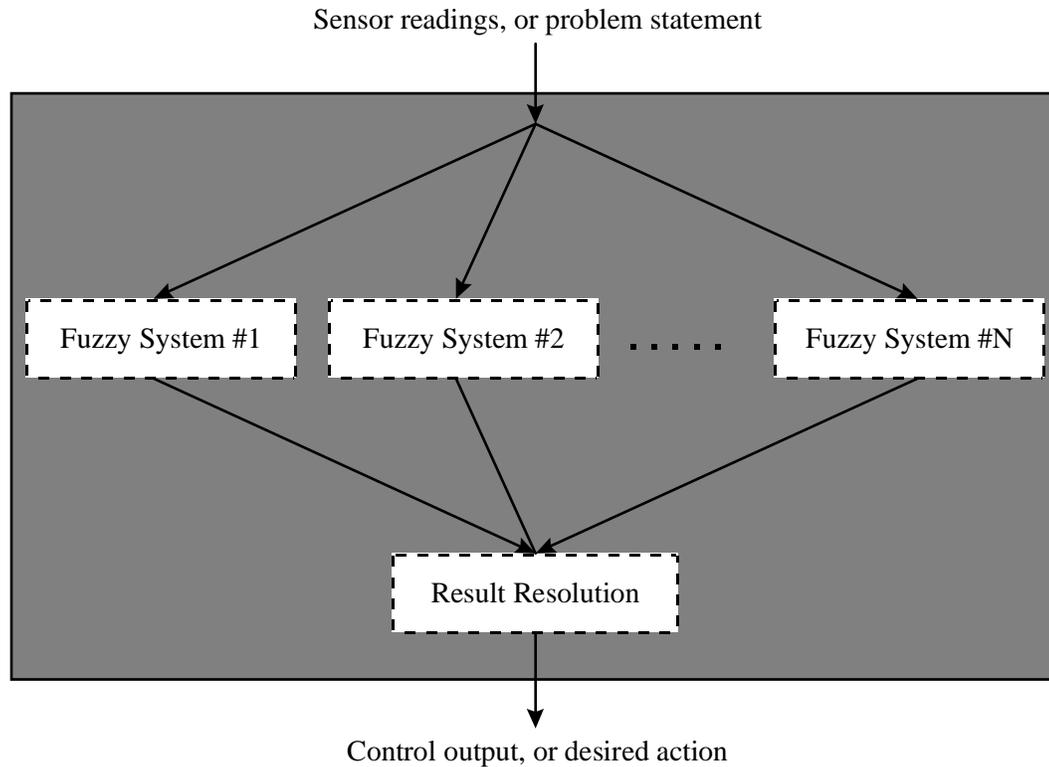


Figure 33. System architecture for parallel competitive fuzzy systems.

For example, Jowers [19] incorporated what he called Local Area Experts (LAE) into an intelligent signal processing tool for real-time diagnoses. In his words,

...the purpose here is not to develop a single, real-time, all encompassing pattern recognition expert, but rather, to develop local experts, all of which simultaneously process data and submit opinions to the LAE manager. The manager then makes a selection... The partitioning of the problem domain minimizes the overhead of each LAE and allows for faster processing of the data at the local level. This design also takes advantage of benefits proffered by the 'parallel processing community'.

Economides [9] applied this approach to aircraft flight control. He created several parallel Flight Mode Experts. Each expert was essentially a fuzzy system designed to recognize a particular operational mode of an aircraft. The experts reported to an Interpreter which made a decision about the aircraft's current state. Economides extended Jowers work by including the concept of confidence. While each Flight Mode Expert reported to an Interpreter its belief that the aircraft was

operating in its respective mode, the Interpreter also generated a confidence factor based on that information.

Other examples of competitive fuzzy systems can be found in [13] and [15]. Halgamuge, et. al. apply the use of two competing fuzzy systems to the nonlinear test problem of backing up a truck and trailer [15]. One of the fuzzy systems is an expert on driving the truck when the angle between the truck and trailer is small. The other fuzzy system is an expert on driving the truck when the angle is large. Both systems calculate a steering wheel input and the final result is computed based on fuzzy sets defined for the angle between the truck and trailer. The authors point to the advantage of the parallel fuzzy systems. “If different strategies are implemented in one single fuzzy controller, validation, maintenance and error correction become almost impossible.”

Control of an inverted pendulum, known as the “cart and pole problem,” has become the standard test bed for fuzzy system algorithms. In [18], Katai, et. al. simulate the control of an inverted pendulum using two, parallel, competitive fuzzy systems. The decomposition of the system is accomplished by decoupling the goal of controlling the pendulum from the goal of controlling the cart. The resulting fuzzy sets from the two parallel rule bases are combined by the maximum aggregation operation. Defuzzification using the centroid method results in the force which should be applied to the cart. The authors show the advantage of segmenting the fuzzy inference stage for genetic algorithm training.

*Cooperative parallel fuzzy systems*, illustrated in Figure 34, distinguish themselves from their competitive counterparts by the fact that each fuzzy system is responsible for a different sub-problem. They may or may not have the same inputs, but their outputs are in different domains. Each rule base is designed independently to make inferences about its own area of specialty. The outputs of the individual expert systems do not need to be resolved, as is the case for the competitive model. This model lends itself well to a distributed computing environment in which the overall problem can be partitioned into separable sub-problems. A mechanism may be needed to allow the parallel systems to share their respective inferred knowledge.

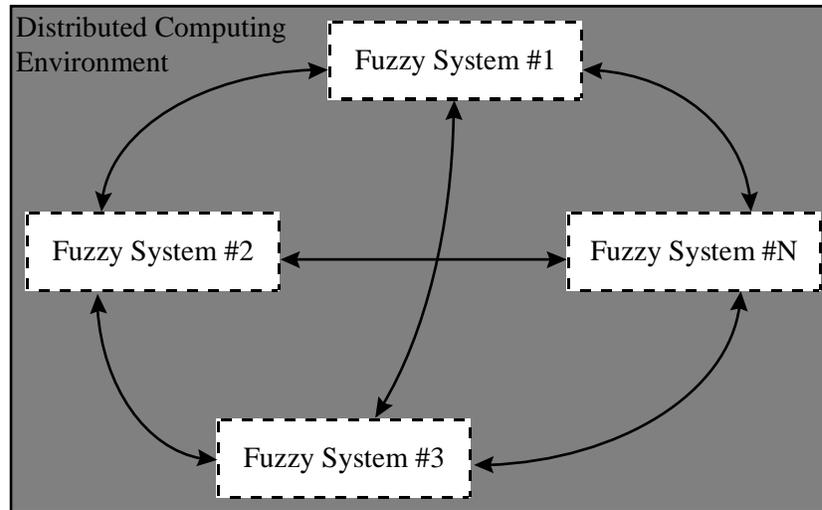


Figure 34. System architecture for parallel cooperative systems.

The advantage of the cooperative model is the distributed nature of the architecture. The parallel systems can be developed by several individuals with expertise in the respective problem domains. Furthermore, the parallel fuzzy systems can be run on separate, and even remote machines. The emerging field of mobile software agents could find significant application in networks using such an architecture.

In [12], Ghabri and Ladet describe a set of fuzzy controllers that could be classified into this category of parallel cooperative systems. The application domain is an automated manufacturing and production system in which each component is controlled by a fuzzy system. Because of interdependence between the various systems, the researchers developed a mechanism through which the parallel fuzzy systems were able to utilize information from each other. They named their architecture Fuzzy Logic Based Distributed Control Systems with Interaction.

### **Summary**

The “curse of dimensionality” is beginning to plague the field of applied fuzzy logic. One solution to this curse is the separation of the problem into more manageable and more easily maintained sub-systems. The author has outlined some of the fuzzy system architectures that have emerged, including architectures motivated by the *ASTRA* project. More importantly, the author presented a revolutionary method for modeling correlated, multidimensional fuzzy sets. The next chapter will show how hypertrapezoidal fuzzy membership functions are being used to perform flight mode interpretation.



## CHAPTER IV

### MULTIDIMENSIONAL FLIGHT MODE INTERPRETATION

#### AUTOMATED SAFETY AND TRAINING AVIONICS

The theory and design principles presented in this dissertation are directly motivated by an on-going research effort at Texas A&M University. The Automated Safety and Training Avionics (*ASTRA*) program is a human-centered design project [33] with the goal of improving the safety and training of pilots of general aviation aircraft. The *ASTRA* system monitors and advises the pilot in procedures and navigation. In order to monitor and advise, the on-board avionics must maintain a qualitative understanding of the operational state of the aircraft. This operational state is called the *flight mode*. The avionics system that determines the flight mode is the *Flight Mode Interpreter* (FMI). According to one of the principle system designers of *ASTRA*, the FMI is the “heart” of the system [39].

The FMI is the heart of the *ASTRA* system for two reasons. First, one of the primary features of the *ASTRA* system is *automatic mode switching*. Automatic mode switching is what enables the *ASTRA* system to dynamically and automatically reconfigure the pilot interfaces. The pilot interfaces include the *head-down display* (HDD) and the *head-up display* (HUD). According to Trang, U.S. Army test pilot and a system designer of *ASTRA*, automatic mode switching can increase situational awareness and reduce pilot errors [39]. The second reason that the FMI can be considered the heart of the *ASTRA* system is that the advice generated by the system for the pilot is mode-dependent. The *Pilot Advisor* software module is responsible for generating advice based on the current flight mode. If the FMI can not determine the appropriate operational mode of the aircraft, the *ASTRA* system is useless.

Not only is the Flight Mode Interpreter an essential element of the *ASTRA* system, it is also a significant technological advancement for the avionics industry. It performs a pattern recognition function which is not implemented in any commercially available avionics system. In 1995, NASA selected *GAPATS* (General Aviation Pilot Advisory and Training System), an implementation of *ASTRA*, for commercialization funding [29] and the State of Texas included *ASTRA* in the State of Texas Advanced Technology Program [41]. The current phase of the *ASTRA* program includes flight tests scheduled for the fourth quarter of 1997.

## THE ENGINEERING FLIGHT SIMULATOR

The development and evaluation of the *ASTRA* system is being performed in the Texas A&M Engineering Flight Simulator (EFS). The EFS, maintained by the Department of Aerospace Engineering, is a fixed-based simulator powered by a Silicon Graphics Reality 2 Graphics Workstation, a R4400 processor, and a three-screen projection system, as shown in Figure 35. A refurbished T-37 cockpit serves as the cockpit of the simulator, as shown in Figure 36. Two computer monitors inside the cockpit provide configurable displays for instrument and head-down displays. The cockpit includes both force and displacement sticks to further facilitate the simulation of various aircraft. The simulator also consists of a network of PCs which receive flight data from the Silicon Graphics computer over an ethernet. The PCs are responsible for driving the instrument displays, head-down displays and head-up displays.

The test vehicle for the *ASTRA* project is a Rockwell Commander 700 twin engine aircraft owned by Texas A&M. One of the aircraft models in the simulator is the Rockwell Commander 700. A system like *ASTRA* could not be developed without a development environment like the EFS. An on-going goal of the *ASTRA* team is that the *ASTRA* system functions identically whether connected to the simulator, or installed in the actual aircraft. Careful planning and modular designs will continue to allow the EFS to be the ideal development and testing platform for advanced cockpit avionics.

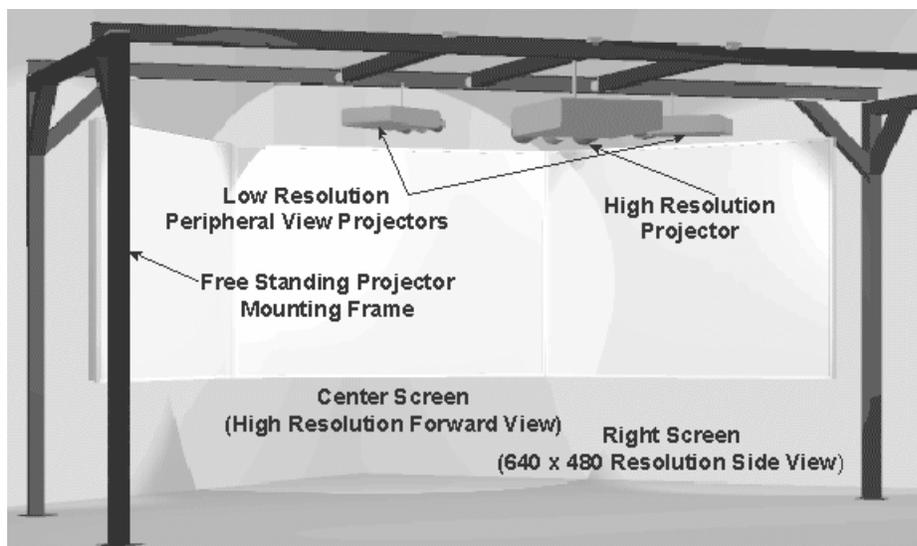


Figure 35. The Engineering Flight Simulator projectors.



Figure 36. The Engineering Flight Simulator cockpit.

## THE FLIGHT MODES

As mentioned in Chapter I and emphasized in the previous section, the Flight Mode Interpreter plays an essential role in the task of improving the pilot's situational awareness. The reason for this is obvious. In order to improve the pilot's situational awareness, the computer must maintain its own situational awareness. Indeed, one of the names used for the Flight Mode Interpreter is "Situation Recognizer." What situations then should this "Situation Recognizer" recognize?

At this stage in the development of *ASTRA*, the flight modes being identified are nominal flight procedures. For example, the Flight Mode Interpreter currently identifies when a pilot is flying a final approach, but does not attempt to identify when the aircraft is nearing a stall. A stall could occur during a final approach and cause an emergency situation. In the current *ASTRA* design, non-nominal conditions are identified by the Pilot Advisor module, which constantly monitors for hazardous situations like stalls, regardless of the FMI indicated flight mode. Some non-nominal conditions are mode-dependent, in which case the Pilot Advisor uses the FMI indicated flight mode to analyze the flight conditions.

The separation of the responsibility for flight mode identification and for anomaly detection was an important decision in the development of *ASTRA*. As will be discussed later, it resulted in a new design philosophy for the fuzzy membership functions of the Flight Mode Interpreter. It is also consistent with the experiences of other researchers of knowledge-based decision aids. The Hazards Monitoring research of Search Technologies, Inc. maintains linked "situation nodes" in their pilot assistant software [35]. Each situation node has an associated list of "expectations." When a situation is identified, the list of expectations associated with that situation are confirmed. If an expectation is not met, an appropriate message is issued to the pilot.

There are then two primary considerations when choosing which flight modes should be identified. First, what operational procedures will likely benefit from having the Pilot Advisor monitor the pilot's actions? Secondly, at which stages of the flight could the pilot's performance be enhanced or his work load be reduced by the system? The flight modes used in the current *ASTRA* effort all meet these criteria. However, as will be discussed in Chapter V, there is a need for one or two additional flight modes. These additional flight modes will augment the usefulness demonstrated by the current *ASTRA* design.

Why use fuzzy sets to model the flight modes? After all, the flight mode decision is a crisp decision. The automatic mode switching of the flight displays requires a crisp specification of the flight mode. If the aircraft state configuration indicates a degree of membership in mode *A*, and a degree of membership in mode *B*, the pilot would not want to see a cluttered overlay of two different displays. While the flight mode decision must be a crisp all-or-nothing decision, the flight modes do overlap in the state space and fuzzy sets are an excellent model of the ambiguity in defining the flight modes. However, there is another motivation for using fuzzy models of the flight modes. The degree of memberships in the fuzzy flight modes can be interpreted as a measure of certainty and used to derive confidences of the flight mode decisions. These certainties and confidence factors allow for filtering the mode decision, an important capability, as will be shown.

## QUANTIFYING RESULTS

The measure of the FMI's performance is how closely its mode decision matches the intended mode of the pilot. During testing of the FMI, the pilot indicates what mode best characterizes the current situation. The FMI should come reasonably close to selecting the same mode based solely on sensor data. The plot of Figure 37 is a good example of the FMI's ability to emulate what the pilot considers to be the modes for an entire flight from takeoff to touchdown. This plot was generated using simulation data from the A&M Engineering Flight Simulator utilizing the Commander 700 model.

Notice that the computer's inference may slightly lead the pilot's stated mode (e.g., the *climbout* to *cruise* transition). At other times the computer may lag in the inference (e.g., the *cruise* to *initapp* transition). However, the FMI's output follows closely enough to allow the Pilot Advisor to give meaningful and timely messages through all seven stages of the flight.

The ability to understand and quantify the performance of the Flight Mode Interpreter was significantly enhanced by the author's development of a MATLAB™ toolbox for flight mode

interpretation. The analysis enabled by the toolbox is the focus of this chapter. Appendix B describes the FMI MATLAB™ Toolbox.

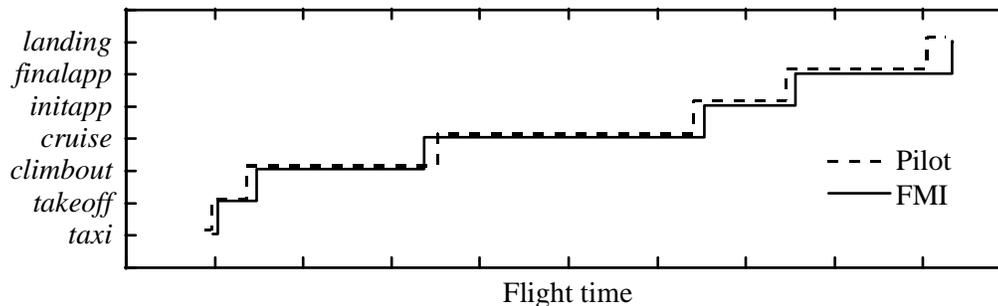


Figure 37. Plot of both pilot's and inferred flight mode.

### ONE-DIMENSIONAL FMI BASELINE

The Flight Mode Interpreter of the *ASTRA* system classifies the operating condition of the aircraft into one of seven predefined modes. The seven operational modes are shown in Table 2. These were the seven modes used in Harral's Flight Mode Interpreter [17]. The FMI performs the flight mode classification based on input variables that are provided by the sensor suite of the aircraft. Table 3 lists the sensed values originally used by Harral in his prototype FMI. These eight variables form an eight-dimensional space of aircraft operation. At any given time in a flight, the aircraft's state can be quantified as a single point in that eight-dimensional space. The job of the FMI is to partition that space into seven different operational modes. Harral's FMI served as the baseline Flight Mode Interpreter for the upgrades and new developments detailed in this chapter.

Table 2.  
Flight modes identified by the FMI.

<b>OPERATIONAL FLIGHT MODES</b>
<i>taxi</i>
<i>takeoff</i>
<i>climbout</i>
<i>cruise</i>
<i>initial approach</i>
<i>final approach</i>
<i>landing</i>

Table 3.  
Inputs to the baseline FMI.

<b>FMI INPUTS</b>
engine power
angle of attack
roll
landing gear setting
flap setting
indicated airspeed
altitude
rate of climb

### ***Certainty and Confidence***

The Flight Mode Interpreter not only produces a qualitative description of the current state of the aircraft, but also provides two measures of that description – the certainty and the confidence. Both measures are in the range of [0, 1]. The certainty is simply the degree of membership that a state vector has in the multidimensional fuzzy set for each mode. The multidimensional fuzzy

modes can be implemented by a composition of one-dimensional fuzzy sets defined in the domains of each input variable. For example, to determine if the airplane is currently in the *takeoff* mode, the sensor readings for the altitude, thrust, rate of climb, etc. are independently calculated using the corresponding *takeoff* fuzzy sets. The certainty is the Bayesian probability of a mode, given the sensor readings. The one-dimensional fuzzy sets implemented in the first prototype are shown in Table 4.

Another calculation has proven useful in understanding the *confidence* associated with a particular flight mode decision. The confidence is calculated based on the flight modes with the highest and second highest certainty values. If  $C_1$  is the certainty of the chosen mode, and  $C_2$  is the next highest calculated certainty, the confidence of a decision is defined as

$$\text{Confidence}(C_1, C_2) = \frac{C_1 - C_2}{C_1} \quad (35)$$

The relationship between the mode certainties and the decision confidence is shown in Figure 38. According to the Bayes isomorphism, the confidence of a decision is related to the odds of making the wrong decision, as shown in Appendix A. The mode certainty is used to make the flight mode decision and, as will be shown in the following section, can be used to filter the mode decision.

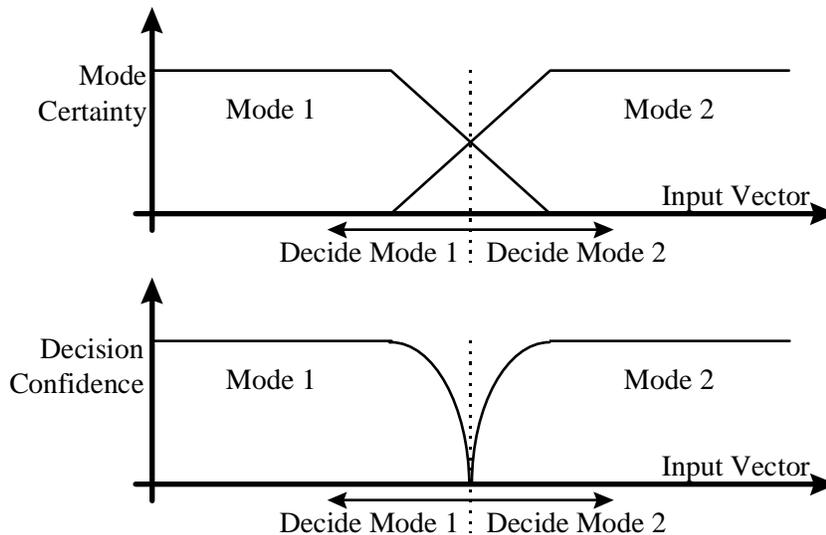


Figure 38. The relationship between mode certainty and decision confidence.

As the airplane transitions from one mode to another, the certainty values will inevitably decrease for the current mode and increase for the next mode. The FMI chooses the mode with the highest certainty. The confidence of that decision decreases as the transition is entered, and then increases as the aircraft enters the next mode of operation.

Table 4.

Harral's one-dimensional fuzzy membership functions.

	<i>taxi</i>	<i>takeoff</i>	<i>climbout</i>	<i>cruise</i>	<i>initapp</i>	<i>finalapp</i>	<i>landing</i>
power [%]	N/A						
alpha [degrees]							
roll [degrees]			N/A		N/A		
landing gear					N/A		
flaps [degrees]							
airspeed [knots]							
altitude [feet]							
rate of climb [fpm]							

## ONE-DIMENSIONAL FMI UPGRADE

The base-line FMI implemented by Harral proved that flight mode interpretation through observation of aircraft configuration was a real possibility. Harral's work not only proved the technology, but also provided a glimpse of the usefulness of such a system. His carefully planned designs and impeccable programming skills have allowed the *ASTRA* program to continue to mature. The author has implemented several important modifications to the one-dimensional version of the FMI during this maturing process. These upgrades to the one-dimensional FMI significantly improved its performance.

### *Anomalies*

Chapter II, page 22 includes a section dealing with uncertainty in fuzzy systems. One of the challenges in the flight mode interpretation problem is how to handle anomalous conditions that do not match any of the modeled flight modes. The method by which the base-line FMI treated anomalies caused the flight mode decision to occasionally oscillate between flight modes. Figure 39 shows such a situation in the transition from *final approach* to *landing*.

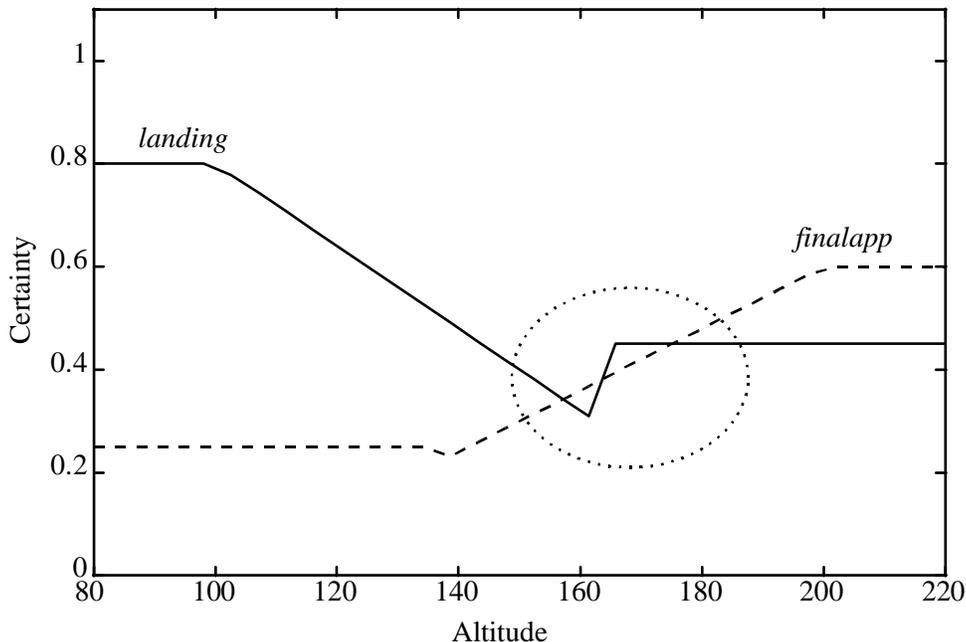


Figure 39. Anomalous inputs caused oscillations in base-line FMI.

In shown in Figure 39, as the altitude decreases, the FMI's decision oscillates between *final approach* and *landing* because of the way the anomalous input (in this case, altitude) is included in the certainty calculation. By changing the certainty calculations to match equation (25), of page 25, the oscillations in the flight mode decision caused by anomalous inputs were eliminated.

### ***FMI Tuning***

The base-line FMI lacked a convenient method of measuring its performance and of understanding the effect various parameters have on the flight mode decision. The author needed a development tool to allow graphical analysis of the FMI performance. Appendix B describes a suite of MATLAB™ functions that enable such an analysis. The toolbox includes a function which plots the flight mode decision for previously recorded flight data. Figure 40 is a plot of the inferred FMI flight mode and the pilot specified flight mode for a landing recorded in the EFS. Clearly, the FMI failed to adequately distinguish between *takeoff* and *initapp* flight modes. Such plots of the flight mode decision revealed the need to tune the fuzzy membership functions of Table 4.

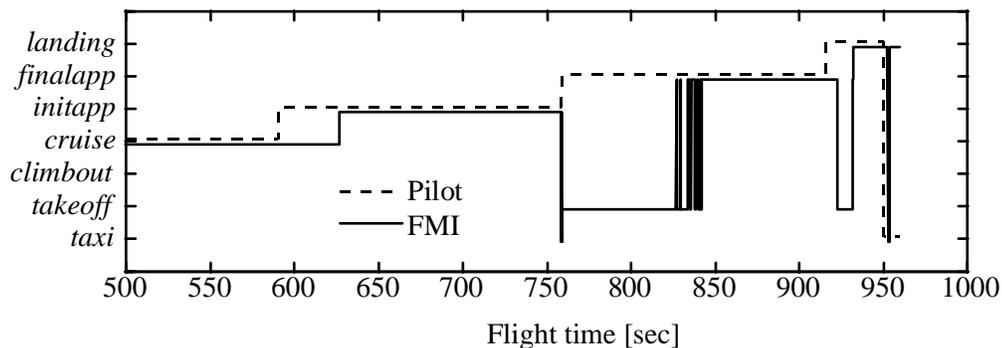


Figure 40. Plot of base-line FMI output for a landing.

While tuning the membership functions for the Flight Mode Interpreter, the author recognized an improved philosophy for their design. The base-line FMI was designed by considering in what range a variable *should be* for a given mode. For example, what *should* the engine power be during an initial approach? This design method was motivated by the fact that the base-line FMI not only generated a flight mode decisions, but also identified “alarms.” However, in the *ASTRA* system, the Pilot Advisor module is now responsible for identifying the alarms, and so the functionality of the FMI can now be limited to that of identifying the most probable flight mode.

From a pilot's perspective, this refocusing of the functionality of the FMI results in more meaningful alarm messages, as discussed in [9].

The design philosophy of the upgraded one-dimensional FMI is based on the range in which a variable *could be*, given a mode. What *could* the engine power be during an initial approach? Designing the fuzzy membership functions to meet the *could be* criteria resulted in a widening of the membership functions, and more importantly, a more robust Flight Mode Interpreter. Figure 41 is the flight mode plotted for the tuned membership functions of the flight of Figure 40. While still not perfect, there is obvious improvement.

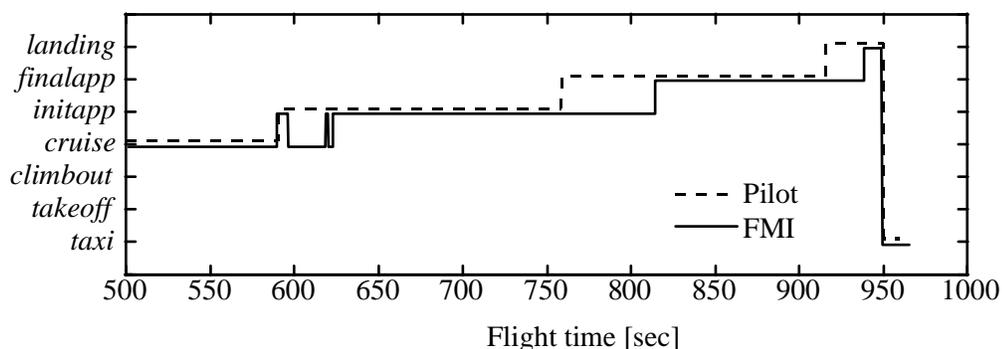


Figure 41. Plot of re-tuned FMI output for the landing of Figure 40.

The new design philosophy also motivated the *ASTRA* team to reexamine the inputs to the FMI. Of particular concern was whether the gear and flap positions should be used in making the flight mode decision. The pilot *could* forget to extend the gear before final approach and confuse the FMI into thinking it was still in initial approach. This problem was confirmed for both gear and flaps in the EFS with the tuned FMI. Consequently, gear and flap positions are no longer inputs to the Flight Mode Interpreter. Finally, through analysis using the author's FMI analysis toolbox, the angle-of-attack was found to not be a useful indicator of flight mode and is no longer used by the FMI.

### **Decision Filtering**

The plot of Figure 41 is an example of the improvement that was achieved by merely adjusting the one-dimensional fuzzy membership functions of the FMI. It also shows one of the major short-comings of the base-line FMI. The flight mode plot at the transition between *cruise* and *initapp* illustrates what the *ASTRA* team has termed "nervousness." In the *ASTRA* system,

these types of oscillations translate into HUD symbology that comes and goes, and HDD messages that flicker off and on. The transitions from one mode to the next were especially susceptible to nervousness. The FMI needed some form of filtering on the decision.

The upgraded Flight Mode Interpreter includes the ability to filter the certainty values for the individual modes. The filter is a low-pass infinite impulse response filter. Filtering the certainty values helps eliminate the nervousness that the *ASTRA* team was observing near the transitions of the modes. Figure 42 is an example of nervousness at the transition between *climbout* and *cruise*. Figure 43 is a plot of the FMI output for the same flight, but with filtering enabled. Obviously, the filtering significantly improves the flight mode decision.

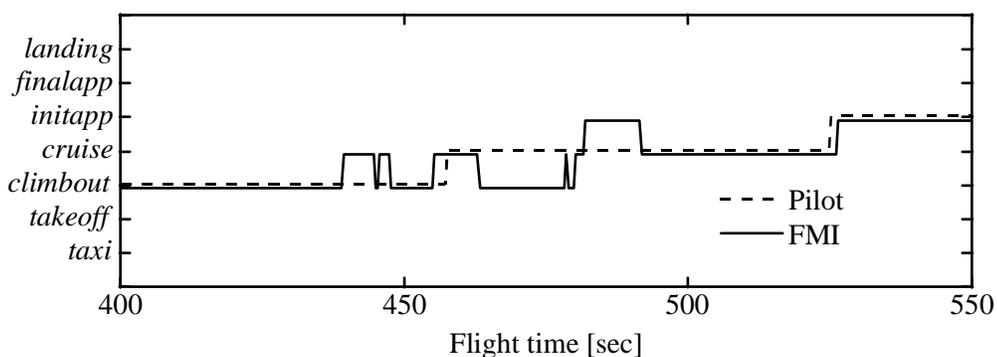


Figure 42. Example of nervousness near mode transitions.

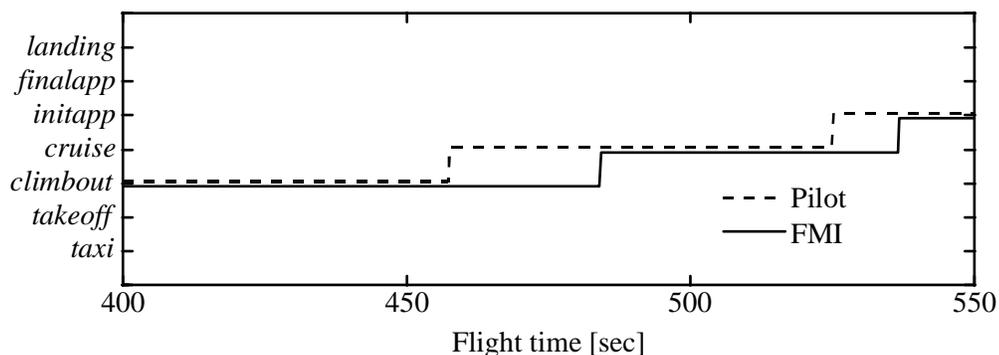


Figure 43. Flight mode decision with filtering, for flight of Figure 42.

The IIR filtering currently implemented in the FMI is the simple, single pole filter of equation (36).  $\mu_A(k)$  are the “raw” certainty values and  $\mu_A'(k)$  are the filtered certainty values. Of

utmost concern when filtering the flight mode decision is the resulting time delay introduced by the filter. Therefore, the coefficients have been parameterized to the time delay,  $t_d$ , of the filter and the sampling frequency,  $f_s$ . Here, the time delay is the time it takes a unit step response to reach 0.9. Because the IIR filter is filtering the certainty values, rather than the decision, the time delay,  $t_d$ , is the approximate worst-case delay introduced by the IIR filter. Generally, the decision delay introduced by the filter will be less than  $t_d$ .

$$\begin{aligned}\mu'_A(k) &= a_0 \cdot \mu_A(k) + b_1 \cdot \mu'_A(k-1) \\ a_0 &\cong \frac{2.25}{f_s \cdot t_d} \\ b_1 &= 1 - a_0\end{aligned}\tag{36}$$

### ***Distance Inputs***

The *ASTRA* team found the flight mode interpretation problem particularly challenging during the approach and landing phase of a flight. The FMI was more susceptible to nervousness and even misclassification during the phase of the flight when an on-board advisory system needs to be most useful. In order to improve the FMI performance during *initial approach*, *final approach* and *landing*, Trang proposed including the distances to various fixes of the approach as state variables for the Flight Mode Interpreter [39]. The distances are listed in Table 5 and are made available to the *ASTRA* system by the Navigation module.

Table 5.  
Distances used in upgraded FMI.

<b>Distances</b>	<b>Definition</b>
aircraft to final approach fix	$d_{AC-F}$
aircraft to the missed approach point	$d_{AC-M}$
initial approach fix to final approach fix	$d_{IF}$
final approach fix to missed approach point	$d_{FM}$
missed approach point to airport	$d_{MA}$

An example of a condition listed in Trang's thesis for inferring *initial approach* is  $(d_{AC-M} \geq d_{FM}) \cap (d_{AC-F} \leq d_{IF})$ . One complication with implementing this condition as a fuzzy rule in the

Flight Mode Interpreter is that such rules will be different for each airport. The fuzzy membership function for  $d_{IF}$  would be airport-dependent. The author was motivated to formulate the rules specified by Trang into a general, airport-independent format. Such a formulation would allow one set of membership functions to be used for any airport.

The author generalized the distance rules found in Trang's thesis for any airport by forming four ratios of the distances listed in Table 5. Consider for example  $(d_{AC-M} \geq d_{FM}) \cap (d_{AC-F} \leq d_{IF})$ . This condition can be rewritten as

$$\left( \frac{d_{AC-M}}{d_{FM}} \geq 1 \right) \cap \left( \frac{d_{AC-F}}{d_{IF}} \leq 1 \right).$$

In effect, the two ratios define two circles in the geography of the approach. The mode *initial approach* can be inferred when the airplane is "outside" one of the circles and "inside" the other circle. Figure 44 illustrates the ratios defining *initial approach*.

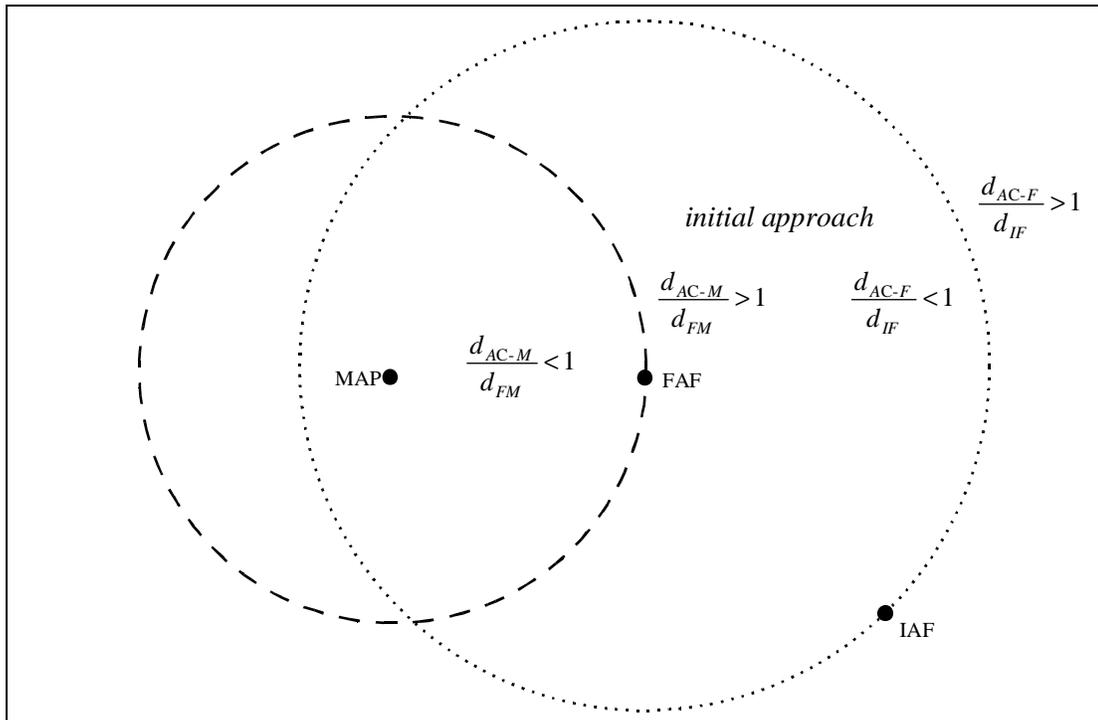


Figure 44. Distance ratios for inferring initial approach.

Fuzzy membership functions can now be defined on the domains of these ratios. For example, two fuzzy membership functions would be required for the rule of Figure 44. One would

be an “open-right” membership function for the  $> 1$  ratio. The other would be an “open-left” membership function for the  $< 1$  ratio. The fuzzy membership functions provide a smooth transition between modes for what would otherwise be hard-decision rules.

To summarize, the distance rules listed in Trang’s thesis can be formulated using the distance ratios of Table 6. For the case where the IAF and the FAF are collocated, Trang proposed using  $d_{IF} = 10$  NM. In effect, these ratios define the geographical regions illustrated in Figure 45. These regions can be combined to infer *initial approach*, *final approach*, and *landing*, as shown in Figure 46. Finally, the fuzzy membership functions of Figure 47 smooth the transitions between modes.

Table 6.

Distance ratios used in upgraded FMI.

<b>Ratio Name</b>	<b>Ratio</b>
distance ratio #1	$\frac{d_{AC-F}}{d_{IF}}$
distance ratio #2	$\frac{d_{AC-F}}{d_{FM}}$
distance ratio #3	$\frac{d_{AC-M}}{d_{FM}}$
distance ratio #4	$\frac{d_{AC-M}}{d_{MA}}$

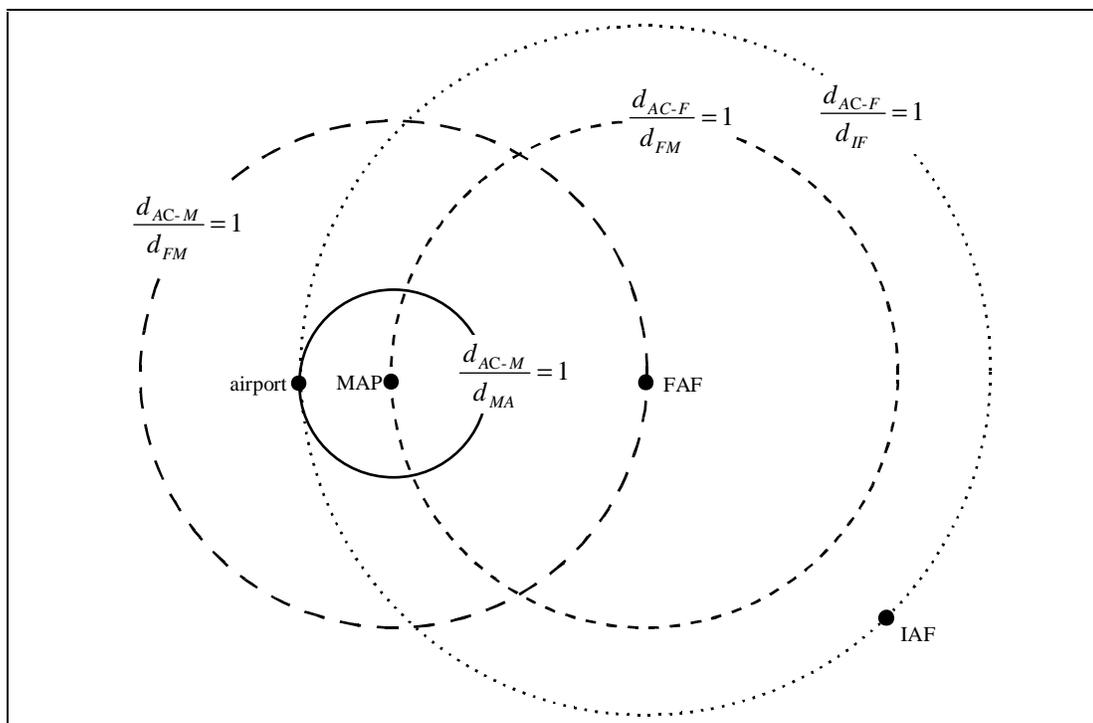


Figure 45. Plot of distance ratios used in upgraded FMI.

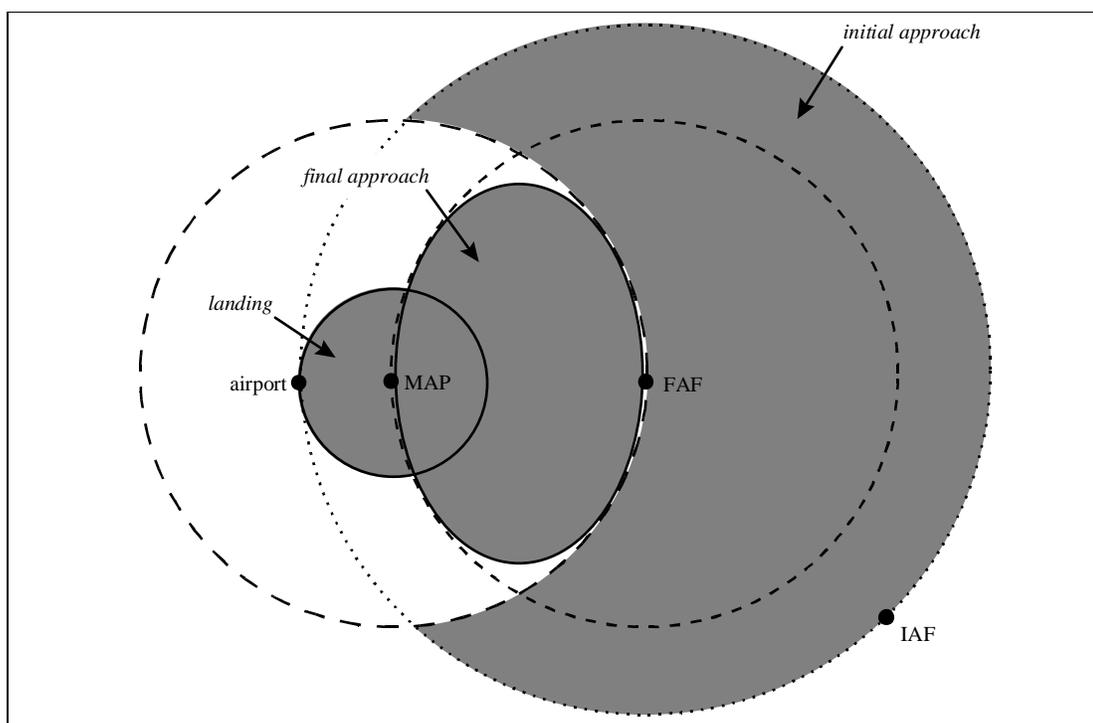


Figure 46. Fuzzy regions defined by distance ratios in upgraded FMI.

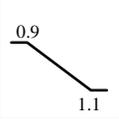
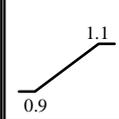
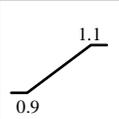
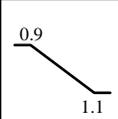
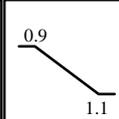
	<i>initapp</i>	<i>finalapp</i>	<i>landing</i>
$\frac{d_{AC-F}}{d_{IF}}$		N/A	N/A
$\frac{d_{AC-F}}{d_{FM}}$	N/A		
$\frac{d_{AC-M}}{d_{FM}}$			N/A
$\frac{d_{AC-M}}{d_{MA}}$	N/A	N/A	

Figure 47. Fuzzy membership functions defined for the distance ratios.

The author has confirmed the usefulness of the distance rules originally proposed by Trang. Figure 48 is a plot of an EFS simulated approach into the Waco's TSTC airport. The flight modes shown were inferred based solely on the fuzzy rule base of Figure 47. Based on this and similar results, distance rules have been incorporated into the upgraded FMI.

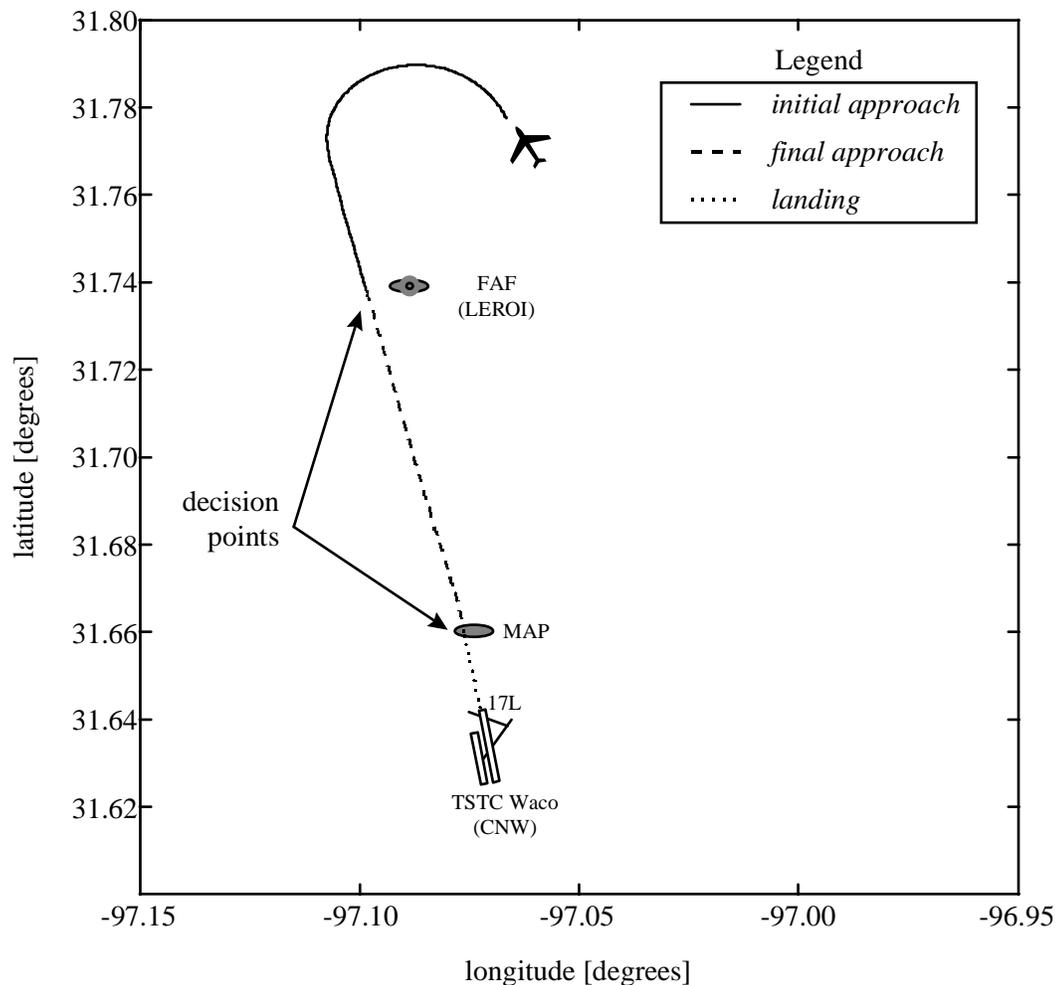


Figure 48. Flight mode interpretation of approach using distance ratios.

### ***Summary of FMI Upgrade***

The base-line Flight Mode Interpreter provided a glimpse of the usefulness of an avionics system which maintains a qualitative assessment of the current flight procedure. However, it also revealed the challenge of the flight mode interpretation problem. Two modifications and two augmentations to the FMI have improved performance: a modified method for certainty calculations when anomalies are detected, a new approach to the design of the fuzzy membership functions, flight mode decision filtering, and the inclusion of distance inputs. The membership functions for the upgraded FMI are shown in Table 7. The next section presents results of a Flight Mode Interpreter based on the author's invention: hypertrapezoidal fuzzy membership functions.

Table 7.  
Fuzzy membership functions for the FMI upgrade.

	<i>taxi</i>	<i>takeoff</i>	<i>climbout</i>	<i>cruise</i>	<i>initapp</i>	<i>finalapp</i>	<i>landing</i>
power [%]							
roll [degrees]							
airspeed [knots]							
altitude [feet]							
rate of climb [fpm]							
$\frac{d_{AC-F}}{d_{IF}}$	N/A	N/A	N/A	N/A		N/A	N/A
$\frac{d_{AC-F}}{d_{FM}}$	N/A	N/A	N/A	N/A	N/A		
$\frac{d_{AC-M}}{d_{FM}}$	N/A	N/A	N/A	N/A			N/A
$\frac{d_{AC-M}}{d_{MA}}$	N/A	N/A	N/A	N/A	N/A	N/A	

## MULTIDIMENSIONAL FMI RESULTS

The development of hypertrapezoidal fuzzy membership functions (HFMFs) introduced in Chapter III, was motivated by the flight mode interpretation problem. The one-dimensional baseline flight mode interpreter uses rule base composition to model the flight modes. That is,  $N$  one-

dimensional functions are multiplied together on the Cartesian product space formed from the  $N$  data variables. For a two-dimensional function, composed from two one-dimensional functions, the fuzzy set's footprint in the plane is rectangular. In the Bayes isomorphism, this corresponds to combining two probability functions as though the variables were independent, and hence uncorrelated. The problem is that for aircraft flight modes, that is a very unrealistic assumption. For example, airspeed and altitude are indeed correlated and the resulting subset shapes are anything but hypercubes.

### ***HFMF Examples***

Examples of hypertrapezoidal fuzzy membership functions designed for the *ASTRA* Flight Mode Interpreter are shown in Figure 49 through Figure 52. The example HFMFs are three-dimensional fuzzy sets defined in the state space of altitude, indicated airspeed, and climb rate. The HFMFs partition the state space into operational flight modes. The flight modes modeled in this example are *takeoff*, *climbout*, *cruise*, *initapp* (initial approach), *finalapp* (final approach), and *landing*.

Unfortunately, fuzzy sets defined on three dimensions can not be illustrated on paper since the degree of membership,  $\mu(x)$ , must be plotted on the third axis of a three-dimensional plot. Therefore, only two inputs can be plotted at a time. In the following examples, the HFMFs are projected onto the axes of indicated airspeed (IAS) and climb rate (ROC). The third input, altitude above ground, is varied from 500 feet in Figure 49 to 4500 feet in Figure 52. Imagine that the aircraft has just taken off in Figure 49, and is gaining altitude in Figure 50 through Figure 52.

Notice in Figure 49 that at a lower altitude, given a positive rate of climb, the flight modes *takeoff* and *climbout* fill the state space. Similarly, a negative rate of climb would indicate that the current flight mode could be characterized by either *finalapp* or *landing*, depending on the indicated airspeed.

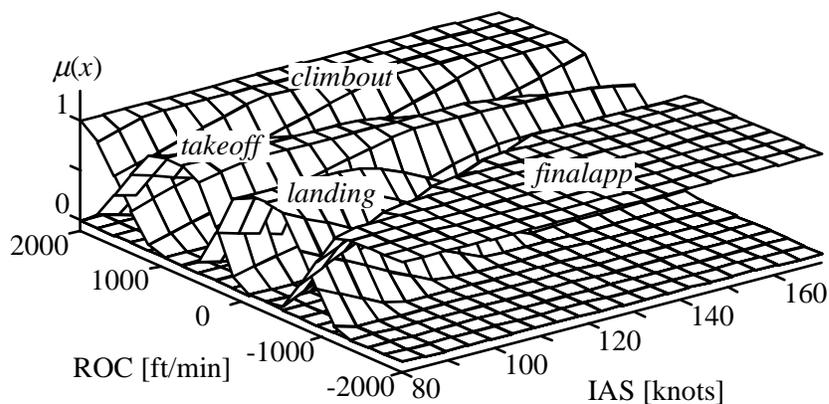


Figure 49. HFMFs for an altitude of 500 feet above ground.

As the altitude increases (see Figure 50), the flight modes *landing* and *takeoff* disappear from the state space and are replaced by *climbout*, *initapp*, and *finalapp*.

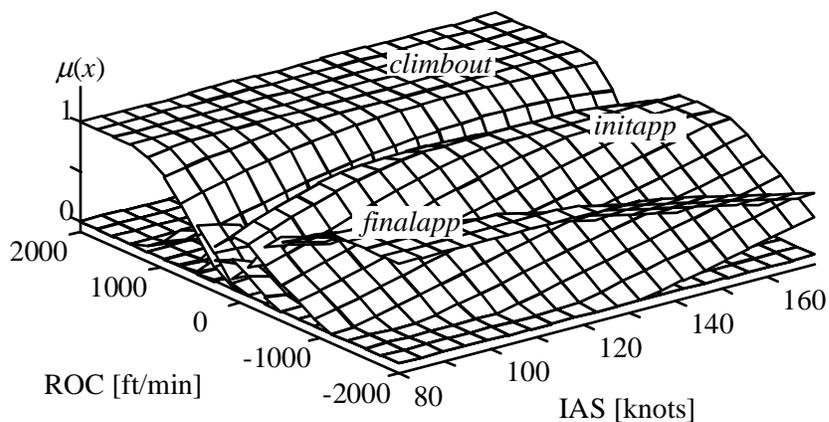


Figure 50. HFMFs for an altitude of 1500 feet above ground.

Figure 51 is a plot of the sample HFMFs at an altitude of 3000 feet. Notice that as the rate of climb approaches zero and as the airspeed increases, the inferred mode becomes *cruise*.

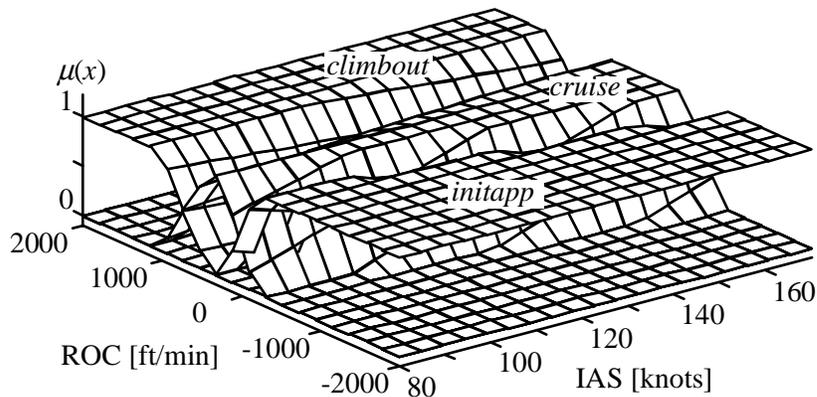


Figure 51. HFMFs for an altitude of 3000 feet above ground.

Figure 52 shows that as the altitude increases (in this case to 4500 feet), the fuzzy set modeling *cruise* expands to include more of the state space.

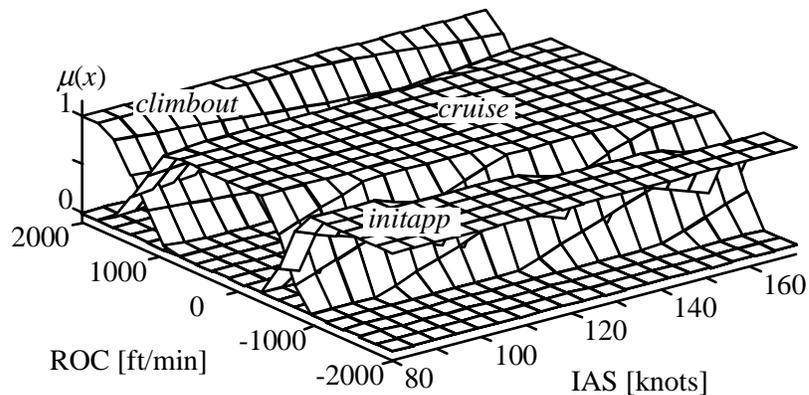


Figure 52. HFMFs for an altitude of 4500 feet above ground.

HFMFs are a powerful, yet simple mechanism for partitioning a state space into fuzzy regions. An important point to draw from Figure 49 through Figure 52 is that one set of prototype points can be used to build multidimensional fuzzy sets which model correlated regions in state space. These plots, and the design that went into them, were accomplished with the author's HFMF MATLAB™ Toolbox, which is documented in Appendix C. The prototype points used for this

sample partitioning of a three-dimensional state space are shown in Table 8. These prototype points were derived using training data from 26 test flights by three different pilots.

Table 8.  
Prototype points for example HFMFs.

	<b>airspeed [knots]</b>	<b>altitude [feet]</b>	<b>rate of climb [fpm]</b>
<i>takeoff</i>	93	85	685
<i>climbout</i>	119	772	1485
	132	2111	1285
<i>cruise</i>	145	2675	-30
	145	2675	30
<i>initapp</i>	130	2360	-648
	120	1870	-766
<i>finalapp</i>	105	950	-1037
	100	526	-905
<i>landing</i>	70	43	-324

Figure 53 is a flight mode plot of an EFS test flight using the hypertrapezoidal FMI and the prototype points of Table 8. Notice that this sample run of the FMI uses only three inputs. Because only three inputs were used, the hypertrapezoidal FMI was especially susceptible to noise in the rate of climb input. The one-dimensional counterpart currently uses nine inputs. With nine inputs driving the fuzzy inference, the one-dimensional FMI is less susceptible to noise.

Consequently, the rate of climb generated by the EFS model required filtering to remove rapid fluctuations. Such filtering of the EFS raw data is justified by the fact that a standard cockpit mounted vertical speed indicator (VSI) includes a low-pass filter with a time delay of six to nine seconds [1]. Therefore, the example shows that given properly filtered inputs, flight mode interpretation can be accomplished with hypertrapezoidal fuzzy membership functions and only three variables.

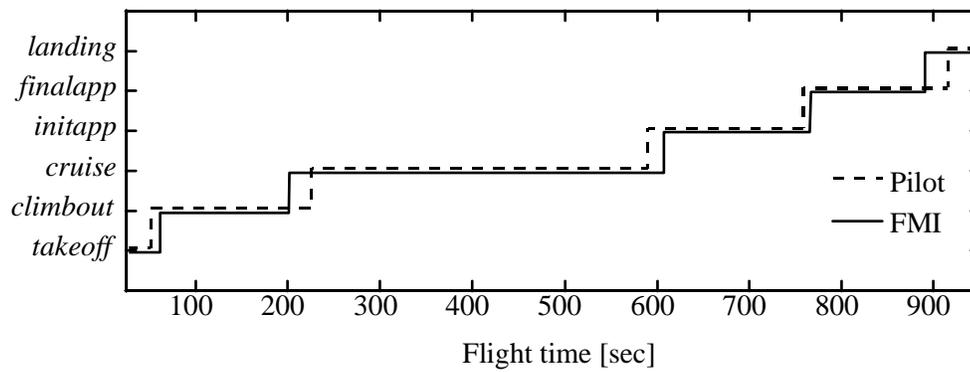


Figure 53. Performance of three input HFMF-based Flight Mode Interpreter.

### *Summary of Hypertrapezoidal FMI*

Hypertrapezoidal fuzzy membership functions are an advancement in the state of the art for fuzzy systems. Motivated by the flight mode interpretation problem, HFMFs are a fuzzy partitioning of a state space using relatively few parameters. The plots in this section show the potential that HFMFs provide for multidimensional problems. The implementation of the hypertrapezoidal FMI and the prototype points of Table 8 were achieved using some preliminary efforts at automatic training. The next chapter suggests that the real potential of HFMFs lies in the automatic training made possible by the unique manner in which they are specified.

## **CHAPTER V**

### **CONCLUSIONS AND RECOMMENDATIONS**

This chapter begins with a summary of the innovations presented in this dissertation. The innovations include both theoretical developments and practical applications. But innovative work seldom answers all the questions, or addresses all the challenges. The section titled “Recommendations for Future Work” lists some of the unanswered questions and a few of the unmet challenges. This chapter should be more than the closing of a dissertation. It could be the prelude to a few yet unwritten.

#### **INNOVATIONS**

This dissertation contains several important innovations related to dimensionality in fuzzy systems. These innovations are motivated by real-world challenges, guided by time-honored scientific principles, and proven through practical application.

##### ***Fuzzy Logic – A Bayesian Science***

The author showed that fuzzy logic can be formulated using first principles of Bayesian probability. While not discounting the contribution mainstream fuzzy logic theorists have made to the field of engineering, an understanding of the isomorphism that exists between fuzzy logic and Bayesian decision theory helps focus theoretical development. Specifically, the interpretation of degrees of membership as conditional probabilities led to a system for fuzzy logic which is consistent with “traditional” approaches to classification, estimation, detection and control.

##### ***Uncertainty in Fuzzy Systems***

The author also addressed the issue of uncertainty in fuzzy systems. How does one deal with anomalous inputs to a system like the *ASTRA* Flight Mode Interpreter? The author explained the effect of various forms of inference, including probabilistic intersection, probabilistic union, anomaly threshold, and normalized sum. Product inference is consistent with a Bayesian approach, but fails to accommodate anomalous inputs. Product inference can be used with a threshold for anomalous inputs, but this approach introduces a non-monotonic characteristic to the inference. Normalized sum is the preferred method for inference when anomalous inputs are a concern.

### ***Hypertrapezoidal Fuzzy Membership Functions***

The most exciting innovation presented in this dissertation is the hypertrapezoidal fuzzy membership function. HFMFs are a mechanism for specifying N-dimensional fuzzy membership functions with relatively few parameters. Currently, the state-of-the-practice in fuzzy engineering is rule-base composition of one-dimensional membership functions. HFMFs address the issue of correlation between system variables, and the exponential growth in the rule-base. The author expects HFMFs to gain wide acceptability as results of their application continue to be publicized.

### ***Fuzzy System Architectures***

The author explained that requirements for many complex fuzzy systems of high dimensionality can be met by partitioning the task responsibility into more manageable sub-components. Nearly all fuzzy system architectures can be categorized as multilevel or parallel. Multilevel fuzzy systems include hierarchical and set-composition models. Parallel fuzzy systems are either competitive or cooperative. Modularity in architecture, coupled with modularity in interface, has proven to be a valuable asset to the *ASTRA* development team.

### ***Flight Mode Interpretation***

The real-world challenge driving theoretical innovation has also served as the practical application. Flight mode interpretation is an innovation in its own right. Yet to be implemented in any commercial avionics system, automatic determination of flight procedure is an important capability when enhancing situational awareness in the cockpit. The *ASTRA* Flight Mode Interpreter is capable of reliably configuring pilot displays with timely information. It also enables procedure-specific advice.

## **RECOMMENDATIONS FOR FUTURE WORK**

No dissertation would be complete without a discussion of the possibility of follow-up research. In fact, several of the topics in this dissertation address "Future Work" suggested by former graduates. As mentioned in a previous chapter, the most promising area of future work is automatic training of hypertrapezoidal fuzzy membership functions.

### ***Additional Flight Modes***

In the Fall of 1997, flight tests are to be conducted on the current *ASTRA* system. It is hoped that these tests will validate the usefulness of the Flight Mode Interpreter in enhancing the

situational awareness of the pilot. The next phase of *ASTRA* must incorporate additional FMI flight modes. A holding procedure, for example, is an excellent opportunity for cockpit automation to assist the pilot. Identifying *holding*, independent of pilot input, may not be useful with the current FMI inputs. It may not be useful because the pilot would be well into the execution of the procedure before the FMI could identify a *holding* mode. A *holding* mode would require either additional inputs to the FMI, or a closer coupling of the FMI and Navigation module.

### ***Non-nominal Flight Mode Interpretation***

The current Flight Mode Interpreter only considers nominal flight modes – *taxi, takeoff, climbout, cruise, initial approach, final approach, and landing*. As *ASTRA* matures, there will be a need to identify non-nominal flight modes. *Stall*, for example, would be an excellent non-nominal flight mode to identify. Perhaps two parallel Flight Mode Interpreters could be used to divide the responsibility of identifying nominal and non-nominal conditions. The nominal FMI would continue to drive the head-up and head-down displays. The non-nominal FMI would continuously monitor for potentially hazardous situations.

### ***Expanded Situational Awareness***

Part of the responsibility of a non-nominal FMI could be to monitor for hazardous weather patterns or unexpected air traffic conditions. Such enhanced monitoring will be possible through proposed upgrades to the national aviation system. These upgrades include a significant number of ground-to-air and even air-to-air data links. The *ASTRA* program is partly motivated by the large amount of information available to the pilot. The amount of available information is about to grow to an unprecedented level. Texas A&M and the *ASTRA* program are in an excellent position to find ways of fusing that information into enhanced situational awareness, and reduce information overload for the pilot.

### ***HFMF Systems***

There are several unanswered questions regarding the application of hypertrapezoidal fuzzy membership functions. One area which was not explored during the project work of this dissertation is the use of one-dimensional and multidimensional membership functions in a single fuzzy system. How does one build a rule-base for both standard membership functions and hypertrapezoidal membership functions? Can HFMFs be used as output sets of a fuzzy system? How are HFMFs “defuzzified”? HFMFs will continue to be a rewarding research area.

### *Analog HFMF Device*

Hypertrapezoidal fuzzy membership functions are computationally intensive. There are many potential applications for an analog circuit which could “calculate” HFMFs. The author believes that the HFMF equations could be implemented with comparators and amplifiers. A chip with the capability to classify signals into fuzzy sets could be used in speech recognition, adaptive control, or similar embedded systems requiring pattern recognition capability. A reliable training procedure for HFMFs would further enhance the usefulness of such a chip.

### *HFMF Training, Statistical*

Automatic training of hypertrapezoidal fuzzy membership functions has the most potential for rewarding research. Specified with relatively few parameters, HFMFs will be an excellent mechanism for fuzzy systems which require learning – both off-line training and on-line adaptation. The most obvious approach for training HFMFs would be to interpret the prototype points as statistical means. The problem with this straight-forward approach is illustrated in Figure 54.

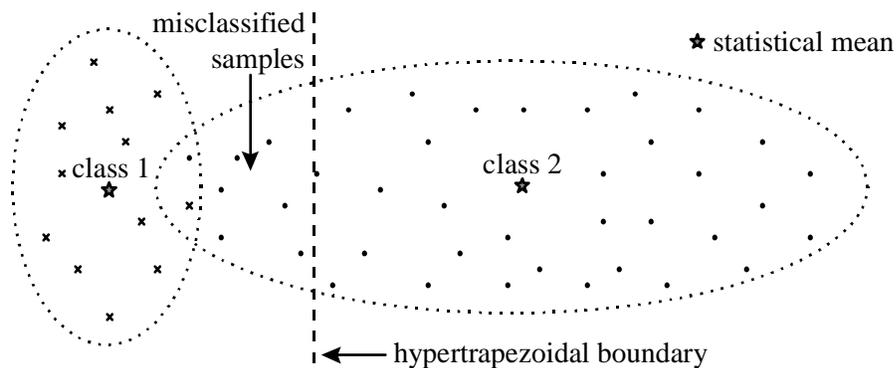


Figure 54. Using statistical means as prototype points.

Using the mean as the prototype point can introduce a bias in the HFMF decision boundary. One approach to correcting this bias would be to adjust the prototype points based on the statistical variance of the two classes. Another option would be to project the prototype points an equal distance away from the overlap. In preliminary attempts at HFMF training, the author used a  $k^{\text{th}}$ -nearest neighbor algorithm to estimate the mean of the overlap. Prototype points were then placed equidistant from the overlap, in the direction of their respective means. Such statistical-based placement of prototype points could serve as the basis of HFMF training.

### ***HFMF Training, Neural***

Neural networks are another potential method for automating the design of HFMFs. In particular, the author suggests a study of *adaptive resonance theory* networks [4]. ART networks may not be directly applicable to HFMF training, because they are used to cluster training data which does not already include pre-defined class associations. However, they are based on Euclidean distance classification, and include a mechanism for creating additional neural nodes as needed. The *vigilance parameter* of ART networks may provide insight into how to add prototype points to an HFMF system. Such a study could be performed and compared with a neural network version of the FMI, as implemented by Nguyen [26]. A neuro-fuzzy system is another candidate for implementing automatic training of an FMI [20].

### ***HFMF Training, Genetic***

Genetic algorithms is a set of techniques for searching a state space for an optimal solution. Harral, in his development of the base-line Flight Mode Interpreter, explored the use of genetic algorithms for tuning one-dimensional fuzzy membership functions [17]. To extend Harral's work, a researcher should not allow the genetic algorithm too much freedom in the design of the membership functions. HFMF prototype points could be directly encoded into genetic chromosomes. The genetic algorithm would repeatedly adjust the prototype points, attempting to correctly classify as many of the training samples as possible.

### ***In Closing***

These are a few of the possible directions for future research. But it is the author's intention to provide more than a starting point for future research. Hopefully, the ideas and experiences outlined in this dissertation will benefit engineering practitioners who are building systems today.

## REFERENCES

- [1] *Aviation Fundamentals*. Englewood, CO: Jeppesen Sanderson, Inc., 1991.
- [2] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Management Science*, vol. 17, no. 4, pp. 141-164, December 1970.
- [3] H. R. Berenji and P. S. Khedkar, "Clustering in product space for fuzzy inference," in *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 1993, pp. 1402 - 1407.
- [4] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, no. 37, pp. 54-115, 1987.
- [5] P. Cheeseman, "Probability versus fuzzy reasoning," in *Uncertainty in Artificial Intelligence*, L. N. Kanal and L. F. Lemmer, Eds. Amsterdam: Elsevier, 1986, pp. 85-102.
- [6] P. Cheeseman, "Discussion: fuzzy thinking," *Technometrics*, vol. 37, no. 3, pp. 282-283, August 1995.
- [7] B. Cleveland and A. Meystel, "Dynamic predictive planning + fuzzy logic control = intelligent control," in *Proceedings of the 1990 International Symposium on Intelligent Control*, 1990, pp. 1217-1224.
- [8] J. Cox, "What is this thing called...AGATE?" *Sport Aviation*, vol. 46, no. 3, pp. 40-45, March 1996.
- [9] G. T. Economides, "An object-oriented, abductive interpreter for aircraft flight control," M.S. Thesis, Department of Electrical Engineering, Texas A&M University, College Station, TX, August 1993.
- [10] G. T. Foster and C. Khambhampati, "Optimal set placement and multi-dimensional fuzzy sets for fuzzy logic controllers," in *Proceedings of the International Conference on Control'94*, March 1994, pp. 658 - 663.
- [11] T. Fukuda and T. Shibata, "Hierarchical control system in intelligent robotics and mechatronics," in *Proceedings of the 1993 International Conference on Industrial Electronics, Control, and Instrumentation*, March 1993, pp. 33 - 38.
- [12] M.-K. Ghabri and P. Ladet, "Application of fuzzy control to a high-speed discrete event system," in *Proceedings of the 1995 IEEE/IAS International Conference on Industrial Automation and Control*, 1995, pp. 89-94.
- [13] O. Ghanayem and L. Reznik, "Hierarchical versus adaptive fuzzy logic controller: design and performance," in *Proceedings of the 1994 IEEE Conference on Intelligent Information Systems*, 1994, pp. 224 - 228.
- [14] E. A. Glass and J. H. Painter, "Object-oriented design of discrete fuzzy control for aircraft flaps," in *Proceedings of the 2nd International Workshop on Industrial Fuzzy Control and Intelligent Systems*, College Station, TX, December 1992, pp. 108-112.
- [15] S. K. Halgamuge, T. A. Runkler, and M. Glesner, "A hierarchical hybrid fuzzy controller for realtime reverse driving support of vehicles with long trailers," in *Proceedings of the 1994 Fuzzy Systems International Conference*, 1994, pp. 1207-1210.

- [16] J. C. Hancock and P. A. Wintz, *Signal Detection Theory*. New York: McGraw-Hill Book Co., 1966.
- [17] V. Q. Harral, "Genetic algorithms for tuning fuzzy membership functions in flight control software," M.S. Thesis, Department of Electrical Engineering, Texas A&M University, College Station, TX, December 1995.
- [18] O. Katai, M. Ida, T. Sawaragi, S. Iwai, S. Kohno, and T. Kataoka, "Constraint-oriented fuzzy control schemes for cart-pole systems by goal decoupling and genetic algorithms," *ifuzzy Control Systems*, A. Kandel and G. Langholz, Eds. Boca Raton, FL: CRC Press, 1994, pp. 181-195.
- [19] S. J. Jowers, "Symbolic diagnoses for intelligent control of real-time systems," M.S. Thesis, Department of Electrical Engineering, Texas A&M University, College Station, TX, May 1989.
- [20] W. E. Kelly, R. Chaloo, R. McLauchlan, and S. I. Omar, "Neuro-fuzzy control of a robotic arm," in *Intelligent Engineering Systems Through Artificial Neural Networks, Volume 6*, C. H. Dagli, M. Akay, C. L. P. Chen, B. R. Fenandez, and J. Ghosh, Eds. New York: ASME Press, 1996, pp. 837-842.
- [21] W. E. Kelly and J. H. Painter, "Soft computing in the general aviation cockpit," in *Proceedings of the First On-line Workshop on Soft Computing*, Nagoya University, Japan, August 1996, pp. 151-156.
- [22] W. E. Kelly and J. H. Painter, "Hypertrapezoidal fuzzy membership functions," in *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, New Orleans, LA, September 1996, pp. 1279-1284.
- [23] S. L. Lass, "An object-oriented, knowledge-based aircraft approach controller," M.S. Thesis, Department of Electrical Engineering, Texas A&M University, College Station, TX, May 1995.
- [24] M. Laviolette, J. W. Seaman, J. D. Barrett, and W. H. Woodall, "A probabilistic and statistical view of fuzzy methods," *Technometrics*, vol. 37, no. 3, pp. 249-261, August 1995.
- [25] D. A. Linkins and M. F. Abbod, "Generic system architecture for supervisory fuzzy control," *Intelligent Systems Engineering Journal*, pp. 181-193, Winter 1994.
- [26] T. H. Nguyen and D. T. Ward, "A neural-network based inference engine for a general aviation pilot advisor," Paper #97-0221 of the *AIAA 35<sup>th</sup> Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 1997.
- [27] J. H. Painter, "Knowledge-based control," in *Proceedings of the 1992 IEEE Symposium on Computer Aided Control System Design*, Napa, CA, March 1992, pp. 138-147.
- [28] J. H. Painter, "Fuzzy decision and control, the Bayes context" in *Proceedings of the 32nd IEEE Conference on Decision and Control*, San Antonio, TX, December 1993.
- [29] J. H. Painter, D. T. Ward, and J. W. Crump, "General aviation pilot advisory and training system (GAPATS)", Proposal for Phase II, STTR/SBIR Contract, NASA Langley Research Center, Hampton, VA, 1994.

- [30] J. H. Painter, E. Glass, G. Economides, and P. Russell, "Knowledge-based processing for aircraft flight control," Final Report for NASA Grant NAG-1-1066 FDP, NASA Langley Research Center, Hampton, VA, 1994.
- [31] J. H. Painter, D. T. Ward, J. W. Crump, and W. W. Nash, "General aviation pilot advisory and training system," Final Report for Phase-1, Contract NAS1-20290, NASA Langley Research Center, Hampton, VA, 1995.
- [32] M. H. Perng and H. H. Chang, "Intelligent supervision of servo control," in *Proceedings of the IEE Conference on Control Theory & Applications*, November 1993, pp. 405 - 412.
- [33] W. B. Rouse, *Design for Success – A Human-Centered Approach to Designing Successful Products and Systems*. New York: John Wiley and Sons, Inc., 1991.
- [34] D. G. Schwartz and G. J. Klir, "Fuzzy logic flowers in Japan," *IEEE Spectrum*, pp. 22-24, July 1992.
- [35] M. D. Skidmore, J. P. Zenyuh, R. L. Small, T. J. Quinn, and A. D. Moyers, "Dual use applications of hazard monitoring: commercial and military aviation and beyond," in *Proceedings of the IEEE/AESS/SAE National Aerospace & Electronics Conference*, Dayton, OH, May 1995.
- [36] M. D. Skidmore, J. P. Zenyuh, R. L. Small, and T. J. Quinn, "A prototype flight management software error monitor," Final Report for NASA Contract NAS1-20210, NASA Langley Research Center, Hampton, VA, September 1995.
- [37] *Student Edition of MATLAB*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1995.
- [38] S. L. Tanimoto, *The Elements of Artificial Intelligence*. Rockville, MD: Computer Science Press, Inc., 1987.
- [39] J. A. Trang, "Automated safety and training avionics for general aviation aircraft," M.S. Thesis, Department of Electrical Engineering, Texas A&M University, College Station, TX, May 1997.
- [40] L. X. Wang, *Adaptive Fuzzy Systems and Control*. Englewood Cliffs, NJ: PTR Prentice Hall, 1994.
- [41] D. T. Ward and J. H. Painter, "Smart cockpit computing technology," Proposal for the State of Texas Advanced Technology Program, Austin, TX, Task #-999903-108, 1995.
- [42] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 - 353, June 1965.

**APPENDIX A**  
**PROOFS AND DERIVATIONS**

**CONTROL DENSITY EQUATION**

The control density of equation (12), page 19, can be found from first principle as follows.

$$\begin{aligned}
 F_{ZX}(z,x) &= P\{(Z \leq z) \cap (X \leq x)\} \\
 &= P\{(Z \leq z) \cap (X \leq x) \cap S\} \\
 &= P\left\{(Z \leq z) \cap (X \leq x) \cap \bigcup_i E_i\right\} \\
 &= P\left\{\bigcup_i (Z \leq z) \cap (X \leq x) \cap E_i\right\} \\
 &= \sum_i P\{(Z \leq z) \cap (X \leq x) \cap E_i\} - P\left\{\bigcap_i (Z \leq z) \cap (X \leq x) \cap E_i\right\} \\
 &= \sum_i P\{(Z \leq z) \cap (X \leq x) \cap E_i\} - P\left\{(Z \leq z) \cap (X \leq x) \cap \bigcap_i E_i\right\}
 \end{aligned}$$

Because events  $E_i$  are defined to be mutually exclusive, the probability of the intersection of all the events is zero.

$$F_{ZX}(z,x) = \sum_i P\{(Z \leq z) \cap (X \leq x) \cap E_i\}$$

By the definition of a conditional probability,

$$\begin{aligned}
 F_{ZX}(z,x) &= \sum_i P\{(Z \leq z) | (X \leq x) \cap E_i\} \cdot P((X \leq x) \cap E_i) \\
 &= \sum_i P\{(Z \leq z) | (X \leq x) \cap E_i\} \cdot P((X \leq x) | E_i) \cdot P(E_i) \\
 &= \sum_i F_{ZX}(z | (X \leq x) \cap E_i) \cdot F_X(x | E_i) \cdot P(E_i)
 \end{aligned}$$

Then by taking the partial derivative  $\frac{\partial^2}{\partial z \partial x}$ ,

$$f_{ZX}(z,x) = \sum_i f_{ZX}(z | x, E_i) \cdot f_X(x | E_i) \cdot P(E_i)$$

Now, to find the conditional density,

$$f_{ZX}(z|x) = \frac{f_{ZX}(z,x)}{f_X(x)} = \frac{\sum_i f_{ZX}(z|x, E_i) \cdot f_X(x|E_i) \cdot P(E_i)}{\sum_j f_X(x|E_j) \cdot P(E_j)}$$

QED.

### DERIVATION OF HFMF EQUATIONS

Chapter III introduced hypertrapezoidal fuzzy membership functions (HFMF). The derivation of HFMFs begins with two prototype points in a two-dimensional plane, as shown in Figure 55. The prototype points are the defining parameters for two fuzzy sets in state space. For a point,  $x$ , in the state space, the degree of membership  $\mu_i(x)$  depends on the Euclidean distance between  $x$  and each of the prototype points. “Near” prototype point  $\lambda_i$ ,  $\mu_i(x) = 1$  and  $\mu_j(x) = 0$ . “Between”  $\lambda_i$  and  $\lambda_j$ ,  $x$  is in the fuzzy region where the two fuzzy sets overlap. According to the Bayesian interpretation of fuzzy logic,  $\mu_i(x) + \mu_j(x) = 1$ .

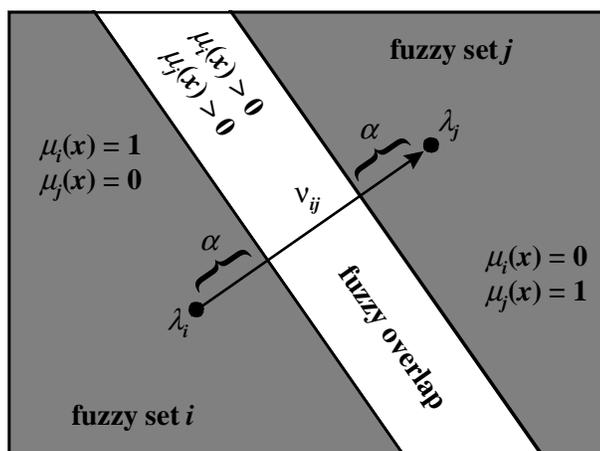


Figure 55. Partitioning a two-dimensional plane with two prototype points.

The width of the fuzzy overlap is set by the system designer with the crispness factor,  $\sigma$ . The author chose to define the crispness factor as

$$\sigma = \frac{2\alpha}{|v_{ij}|}.$$

Therefore, the distance from the prototype points to the region of overlap is

$$\alpha = \sigma \cdot \frac{|v_{ij}|}{2}. \quad (37)$$

Notice that the range of  $\sigma$  is  $[0, 1]$ . For  $\sigma = 0$ , the region of fuzzy overlap extends from prototype point to prototype point. For  $\sigma = 1$ , there is no fuzzy overlap and the fuzzy sets reduce to a minimum distance classifier.

Now we must determine if an arbitrary point,  $x$ , lies in the unity range of a fuzzy set, or within the fuzzy overlap. Consider the case of  $\mu_i(x) = 1$  and  $x$  lies on the boundary of the fuzzy region, as shown in Figure 56.

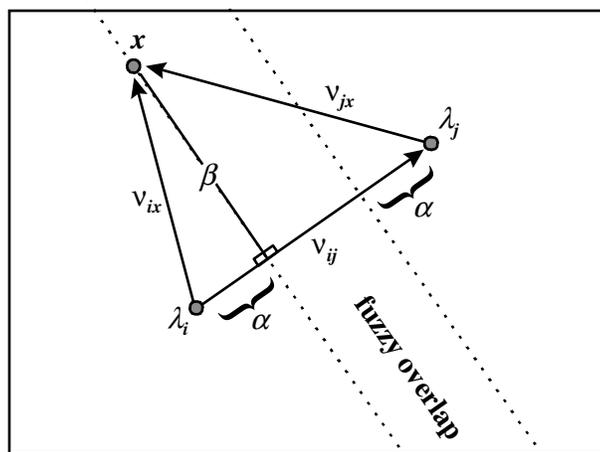


Figure 56. A point,  $x$ , on the boundary of the fuzzy overlap.

The two right triangles satisfy the Pythagorean theorem.

$$\beta^2 = |v_{ix}|^2 - \alpha^2 \quad \beta^2 = |v_{jx}|^2 - (|v_{ij}| - \alpha)^2$$

Combining the two equations yields

$$\begin{aligned}
|v_{ix}|^2 - \alpha^2 &= |v_{jx}|^2 - (|v_{ij}| - \alpha)^2 \\
|v_{ix}|^2 - \alpha^2 &= |v_{jx}|^2 - |v_{ij}|^2 - \alpha^2 + 2\alpha|v_{ij}|. \\
|v_{ix}|^2 &= |v_{jx}|^2 - |v_{ij}|^2 + 2\alpha|v_{ij}|
\end{aligned}$$

By substituting equation (37) and rearranging the terms, we discover the condition for which  $x$  lies on the boundary.

$$\begin{aligned}
|v_{ix}|^2 &= |v_{jx}|^2 - |v_{ij}|^2 + \sigma|v_{ij}|^2 \\
|v_{ix}|^2 &= |v_{jx}|^2 + (-1 + \sigma)|v_{ij}|^2 \\
\frac{|v_{ix}|^2 - |v_{jx}|^2}{|v_{ij}|^2} &= \sigma - 1
\end{aligned}$$

It is convenient to define the left side of the previous equation as a new function,  $\rho_{ij}(x)$ . Notice that  $\rho_{ij}(x)$  is a measure of the normalized, relative distance from  $x$  to the two prototype points. When  $\rho_{ij}(x) = 0$ ,  $x$  is equidistant from the two prototype points. When  $\rho_{ij}(x) < 0$ ,  $x$  is closer to  $\lambda_i$  and when  $\rho_{ij}(x) > 0$ ,  $x$  is closer to  $\lambda_j$ . Finally, notice that  $\rho_{ij}(x) = -\rho_{ji}(x)$ .

By considering the values of  $\rho_{ij}(x)$  in the three regions of the state space, the following conditions become apparent:

$$\begin{aligned}
\text{For } \mu_i(x) = 1: \rho_{ij}(x) &\leq (\sigma - 1) \\
\text{For } \mu_i(x) = 0: \rho_{ij}(x) &\geq (1 - \sigma) \\
\text{For overlap region, } 0 < \mu_i(x) < 1: &(\sigma - 1) < \rho_{ij}(x) < (1 - \sigma)
\end{aligned}$$

For the first two conditions, the degree of membership is either unity or zero. The next question is, “What is the degree of membership of an arbitrary point in the fuzzy region?”

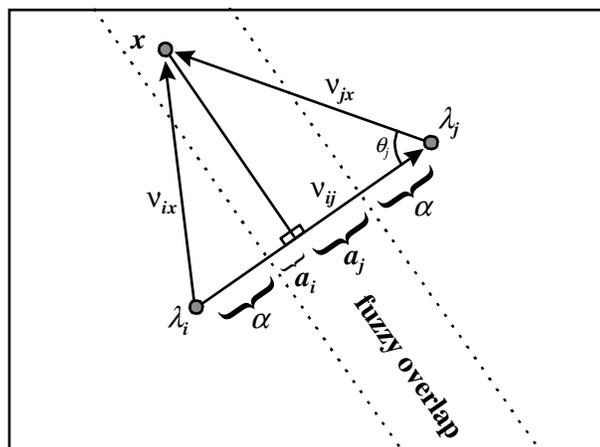


Figure 57. Degree of membership of point in fuzzy region.

First, we should introduce a new notation for  $\mu_i(x)$ .  $\mu_{i|j}(x)$  is the degree of membership in fuzzy set  $i$ , given that fuzzy set  $j$  is the only other set in the partition. To this point we have been working with only two fuzzy sets and two prototype points. However, when more than two sets partition a space, the degrees of membership are calculated two at a time. With reference to Figure 57, for the two fuzzy sets  $i$  and  $j$ , the degree of membership  $\mu_{i|j}(x)$  can be calculated according to the following equation:

$$\mu_{i|j}(x) = \frac{a_j}{a_i + a_j}$$

Notice that this definition causes the two fuzzy sets to resemble overlapping trapezoids in the fuzzy region, as shown in Figure 58.

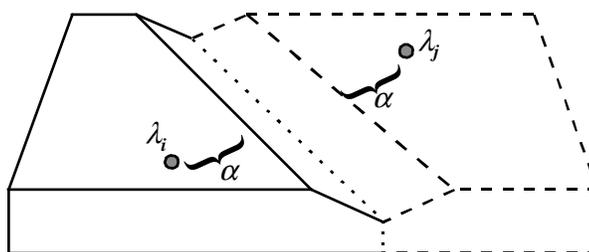


Figure 58. Two overlapping fuzzy sets defined on two dimensions.

Continuing with the development of  $\mu_{ij}(x)$ , and referencing Figure 57, yields the following:

$$\begin{aligned}
 \mu_{ij}(x) &= \frac{a_j}{a_i + a_j} \\
 &= \frac{|v_{jx}| \cdot \cos \theta_j - \alpha}{|v_{ji}| - 2\alpha} \\
 &= \frac{|v_{jx}| \cdot \frac{v_{ji} \cdot v_{jx}}{|v_{ji}| |v_{jx}|} - \alpha}{|v_{ji}| - 2\alpha} \\
 &= \frac{\frac{v_{ji} \cdot v_{jx}}{|v_{ji}|} - \sigma \frac{|v_{ji}|}{2}}{|v_{ji}| - 2\sigma \frac{|v_{ji}|}{2}} \\
 &= \frac{v_{ji} \cdot v_{jx} - \frac{\sigma}{2} |v_{ji}|^2}{|v_{ji}|^2 - \sigma |v_{ji}|^2} \\
 &= \frac{v_{ji} \cdot v_{jx} - \frac{\sigma}{2} |v_{ji}|^2}{(1 - \sigma) |v_{ji}|^2}
 \end{aligned}$$

Notice that  $\mu_{ji}(x) = 1 - \mu_{ij}(x)$ :

$$\begin{aligned}
 \mu_{ji}(x) &= \frac{a_i}{a_i + a_j} \\
 &= 1 - 1 + \frac{a_i}{a_i + a_j} \\
 &= 1 - \frac{a_i + a_j}{a_i + a_j} + \frac{a_i}{a_i + a_j} \\
 &= 1 - \frac{a_j}{a_i + a_j} \\
 &= 1 - \mu_{ij}(x)
 \end{aligned}$$

Therefore, the degree of membership, of a point  $x$ , in two fuzzy sets defined by  $\lambda_i$  and  $\lambda_j$ , is given by the following:

$$\mu_{i|j}(x) = \left\{ \begin{array}{ll} 0; & \rho_{i|j}(x) \geq 1 - \sigma \\ 1; & \rho_{i|j}(x) \leq \sigma - 1 \\ \frac{v_{ji} \cdot v_{jx} - \frac{\sigma}{2} \cdot |v_{ji}|^2}{(1 - \sigma) |v_{ji}|^2}; & \text{otherwise} \end{array} \right.$$

Notice that these equations are general for  $N$  dimensions.

The final step in the derivation of hypertrapezoidal fuzzy membership functions is the determination of  $\mu_i(x)$  when there exist more than two fuzzy sets in the fuzzy partition.  $\mu_i(x)$  could be calculated by multiplying together the  $\mu_{i|j}(x)$  for all  $j \neq i$  (i.e., the numerator of equation (38)). However, in order to satisfy the Bayesian requirement that the degrees of membership sum to one, the calculation of  $\mu_i(x)$  requires an additional normalizing sum (i.e., the denominator of equation (38)). Therefore, the degree of membership of a point in state space  $x$ , in HFMF  $i$ , is

$$\mu_i(x) = \frac{\prod_{j=1 \neq i}^M \mu_{i|j}(x)}{\sum_{k=1}^M \left( \prod_{j=1 \neq k}^M \mu_{k|j}(x) \right)}. \quad (38)$$

## BAYESIAN INTERPRETATION OF DECISION CONFIDENCE

The FMI's decision confidence of page 55, equation (35), was first introduced by Economides [9]. He developed an excellent justification for its use, which concludes, "Thus, the Bayes rationale for the ad hoc confidence function is that it is the Bayes conditional difference between the probabilities of a correct and an incorrect decision, scaled by the probability of a correct decision." That is,

$$\text{Confidence}(C_1, C_2) = \frac{C_1 - C_2}{C_1}$$

is equivalent to

$$\text{Confidence}(x) = \frac{P(C|x) - P(E|x)}{P(C|x)}$$

where  $P(C|x)$  and  $P(E|x)$  are the conditional probabilities of making the correct decision and an incorrect decision, respectively. The two values are also the highest and second highest mode certainties,  $C_1$  and  $C_2$ .

In addition to Economides' conclusion concerning a Bayesian foundation for the decision confidence, it can also be shown that the decision confidence is related to the *odds likelihood* of making the incorrect decision. The odds of an event [8] is defined as

$$O(X) = \frac{P(X)}{1 - P(X)}.$$

Using Economides' basic approach, the decision confidence can be expressed as

$$\begin{aligned} \text{Confidence}(x) &= \frac{P(C|x) - P(E|x)}{P(C|x)} \\ &= 1 - \frac{P(E|x)}{P(C|x)}. \end{aligned}$$

Assuming that  $P(C|x) + P(E|x) = 1$ , then the confidence can be expressed as

$$\begin{aligned} \text{Confidence}(x) &= 1 - \frac{P(E|x)}{1 - P(E|x)} \\ &= 1 - O(E|x) \end{aligned}$$

Therefore, the FMI's decision confidence is related to the odds likelihood of making an incorrect decision.

## APPENDIX B

### FMI MATLAB™ TOOLBOX

#### BACKGROUND

The FMI MATLAB™ Toolbox is a set of MATLAB commands for designing and analyzing the one-dimensional *ASTRA* Flight Mode Interpreter. The Flight Mode Interpreter is responsible for partitioning the N-dimensional state space of aircraft operation into “flight modes”. The flight modes describe some operational or procedural state of the aircraft. The FMI Toolbox primarily takes advantage of MATLAB plotting capabilities to help the system designer build a better Flight Mode Interpreter. The FMI Toolbox imports recorded data from the Engineering Flight Simulator of Texas A&M.

#### REQUIREMENTS

A correctly installed version of MATLAB, of course, is the first requirement for using the FMI Toolbox. The FMI Toolbox does not require any additional toolboxes. It will work with the Student Edition of MATLAB. However, the Student Edition of MATLAB is “limited to 8192 elements, with either the number of rows or columns limited to 32” [37]. The recorded data files from the EFS are 15 columns wide. Therefore, the Student Edition would be limited to 546 samples. For a 5 Hz sampling rate, this would equate to just under two minutes worth of data.

To use the FMI toolbox, the toolbox files can be installed in any directory. At the MATLAB command line, change to the directory containing the files. For example,

```

» cd \users\wally\matlab\fmi
» dir

.          idents.m    jeff08.txt    mbfjeff.m    plotmode.m    tay06.txt
..         jeff01.txt  jeff09.txt    mbfvance.m   readdata.m    tay07.txt
certain.m  jeff02.txt    jeff10.txt    plotalar.m   readme.txt    vance01.txt
compmbfs.m jeff03.txt    jeff11.txt    plotcert.m   tay01.txt     ward01.txt
drew01.txt  jeff04.txt    jeff12.txt    plotdata.m   tay02.txt     ward02.txt
fdm2mats.m jeff05.txt    jeff13.txt    plotflt.m    tay03.txt     ward03.txt
fmi.m      jeff06.txt    makembfs.m   plotmbf.m    tay04.txt     woo02.txt
fmisim.m   jeff07.txt    mbf2mats.m   plotmbfs.m   tay05.txt

```

The m-files are MATLAB functions. The txt-files are flight data files recorded in the Engineering Flight Simulator. `readme.txt` describes the recorded flight data files.

## COMMANDS

### *Help*

The functions available in the FMI Toolbox can be listed by typing `fmi` at the MATLAB command prompt:

```

» fmi

FMI MATLAB Development Toolbox
Version 1.0, by Wallace Kelly

THE MEMBERSHIP FUNCTIONS -----
plotmbf   plot the membership function for one input, one mode
plotmbfs  plot the membership functions for one input, all modes
compmbfs  compare the mbfs for an input on the same plot
plotmode  plot the membership functions for one mode, all inputs

makembfs  create a membership function data matrix
mbf2mats  get information from a membership function data matrix

THE FLIGHT DATA -----
plotdata  plots the input data and the transitions to each mode

readdata  load flight data into a flight data matrix
fdm2mats  get specific information from a flight data matrix

FLIGHT MODE INTERPRETATION -----
plotflt   show graphically the inferred and actual flight modes
plotcert  plot the certainty level and the number of alarms

fmisim    determine the flight mode from data
certain   calculate the certainties for a given input

MISCELLANEOUS -----
idents    defines identifiers for ASTRA modes and inputs

»

```

Help with individual functions can be viewed by typing `help function-name` at the command line:

```

» help idents

IDENTS
Defines convenient identifiers for FMI modes and variables
TAXI, TAKEOFF, CLIMBOUT, CRUISE, INITAPP, FINALAPP, LANDING
THRUST, ALPHA, ROLL, GEAR, FLAPS, IAS, ALT, ROC

Use FMI functions like, plotmode('mbfbest', CLIMBOUT);
instead of, plotmode('mbfbest', 3);

»

```

The `idents` function is a batch file which defines constants for the flight mode and input numbers. Instead of having to remember which flight mode number corresponds to *climbout*, the user can use CLIMBOUT, as shown above.

### **Membership Function Files**

A Flight Mode Interpreter is described in a MATLAB batch file. Variables defined in a batch file have the workspace as their scope, unlike variables defined in function files. The batch file should first define the names of the FMI inputs and the names of the flight modes. For example, `mbfjeff.m` begins with the following lines:

```
inlabels = str2mat('thrust', 'alpha', 'roll', 'gear',
    'flaps', 'airspeed', 'altitude', 'climbrate');

mbflabels = str2mat('taxi', 'takeoff', 'climbout', 'cruise',
    'initapp', 'finalapp', 'landing');
```

These lines define eight inputs and seven flight modes. Each flight mode is then defined by an 8-by-4 matrix. Each row of the matrix contains the *abcd* parameters of a trapezoid. For example, `mbfjeff.m` defines the flight mode *taxi* with the following matrix:

```
taxi = [
    [-inf -inf 45 55];
    [-inf -inf inf inf];
    [-3 -2 2 3];
    [-inf -inf inf inf];
    [-inf -inf inf inf];
    [-inf -inf 10 70];
    [-inf -inf 10 15];
    [-5 0 0 5]; ];
```

This matrix defines the membership function for thrust (the first row) during *taxi* mode to be an open-left trapezoid, which tapers off between 45 and 55 percent. The second row of the *taxi* matrix indicates that the second input, alpha, should not be used when inferring *taxi*. Finally, the membership function should include a matrix defining mode memory. The mode memory matrix in `mbfjeff` is

```
memmat = [
%taxi
    1 1 0 0 0 0 0;
%takeoff
    1 1 1 0 0 0 0;
%climbout
    0 0 1 1 0 0 0;
%cruise
    0 0 1 1 1 0 0;
%initapp
    0 0 0 0 1 1 0;
%finalapp
```

```

    0 0 0 0 0 1 1;
%landing
    1 1 0 0 0 0 1; ];

```

Each row of the memory matrix corresponds to a flight mode. Each element in a row corresponds to a flight mode that might follow the current mode. For example, the second row would indicate that *takeoff* could be followed by *taxi*, *takeoff*, or *climbout*.

### ***Plotting Membership Functions***

The FMI toolbox includes four functions for plotting fuzzy membership functions – `plotmbf`, `plotmbfs`, `compmbfs`, and `plotmode`. The first, `plotmbf`, plots the membership function for a single input, and a single mode. For example,

```

» help plotmbf

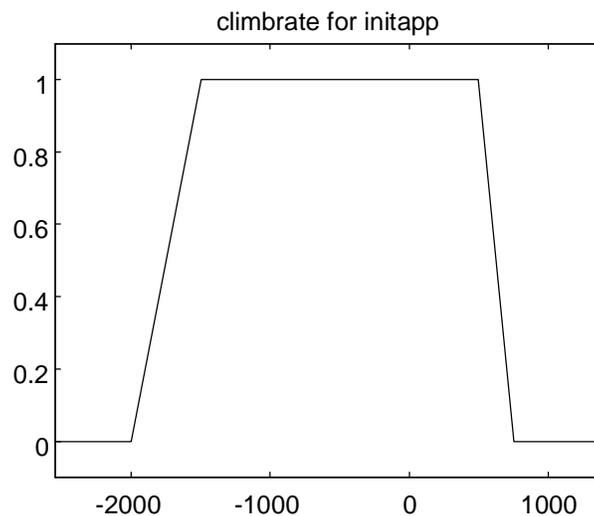
function plotmbf(mbfname, input, mode, mn, mx)

Plot the membership function
found in the file named mbfname
for the input number, input
and for mode number, mode
(optional) specify the min and max points to plot

Example: plotmbf('mbfjeff', 7, 3)

» idents
» plotmbf('mbfjeff', ROC, INITAPP)

```



»

`plotmbfs` plots the membership functions for a single input, and for all the modes. The plots are tiled, as shown:

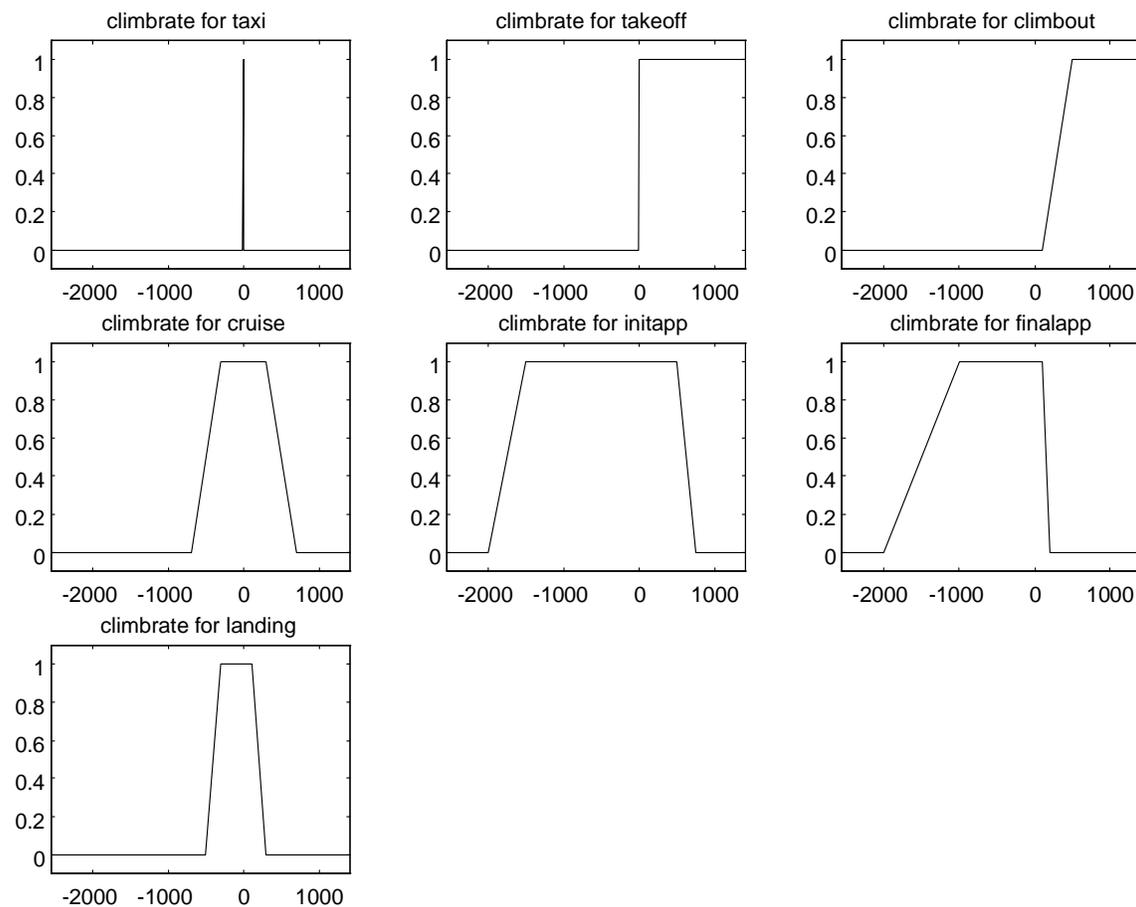
```
» help plotmbfs
```

```
function plotmbfs(mbfname, input, mn, mx)
```

```
Plot the membership functions
found in the file named mbfname
for the input number, input
(optional) specify the min and max points to plot
```

```
This function tiles the plots for all the flight modes
See compmbfs to compare the mbfs on the same plot
```

```
» plotmbfs('mbfjeff', ROC)
```



```
»
```

To compare the membership functions for a single input and various modes on the same axes, use `compmbfs`.

```
» clg
» help compmbfs
```

```
function compmbfs(mbfname, input, modes, mn, mx)
```

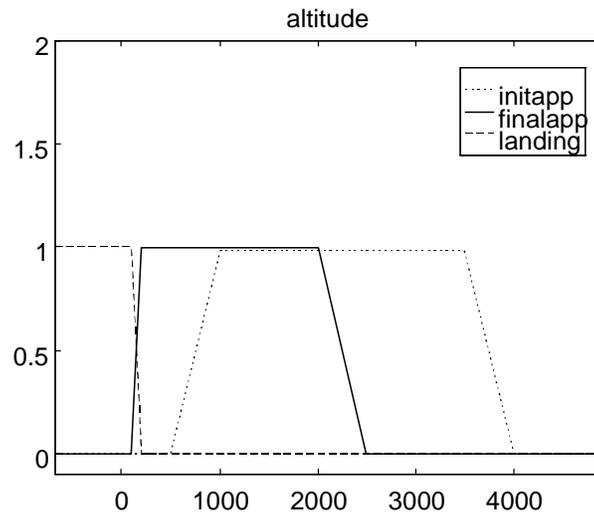
```
Plot the membership functions
```

found in the file named mbfname  
 for the input number, input  
 (optional) specify that only certain modes be plotted  
     for example: modes = [3; 4; 5]  
 (optional) specify the min and max points to plot

This function plots the modes on the same axes  
 See plotmbfs to plot the mbfs in a tile format

Note, the plots are scaled slightly to improve visibility  
 in the overlap regions

» compmbfs('mbfjeff', ALTITUDE, [INITAPP; FINALAPP; LANDING])



»

Finally, plotmode allows the user to view the membership functions for all the inputs of one mode.

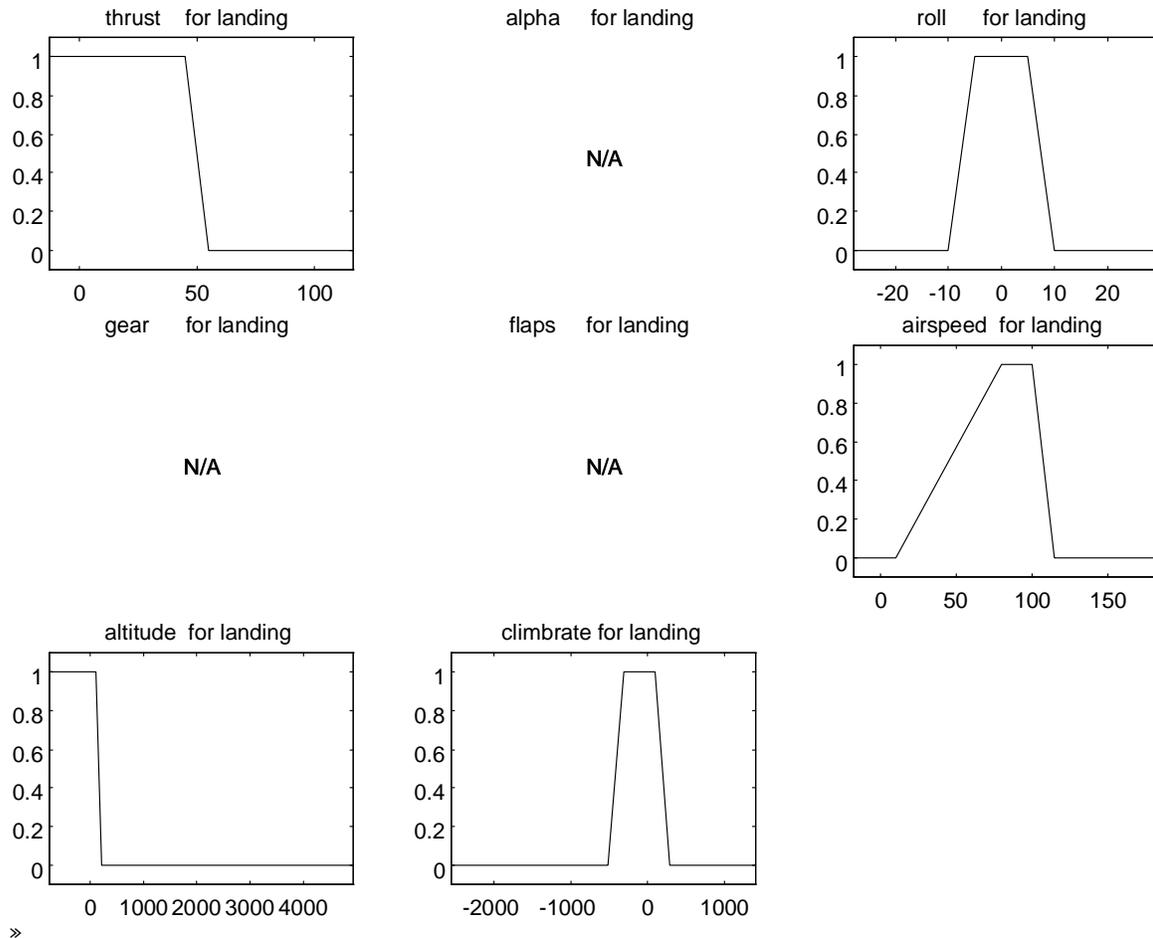
» help plotmode

```
function plotmode(mbfname, modenum)
```

```
Plot the membership functions
found in the file named mbfname
for the mode number modenum
for all the inputs.
```

This function tiles the plots.

```
» plotmode('mbfjeff', LANDING)
```



### ***Plotting Flight Data***

The FMI toolbox also includes the capability to plot flight data from recorded EFS data. `plotdata` allows the user to plot all the stored data, windows of data, and single variables. Vertical dotted lines on the plot show where the pilot indicated a change in flight mode. Vertical dashed lines indicate the FMI's inference.

```
» help plotdata
```

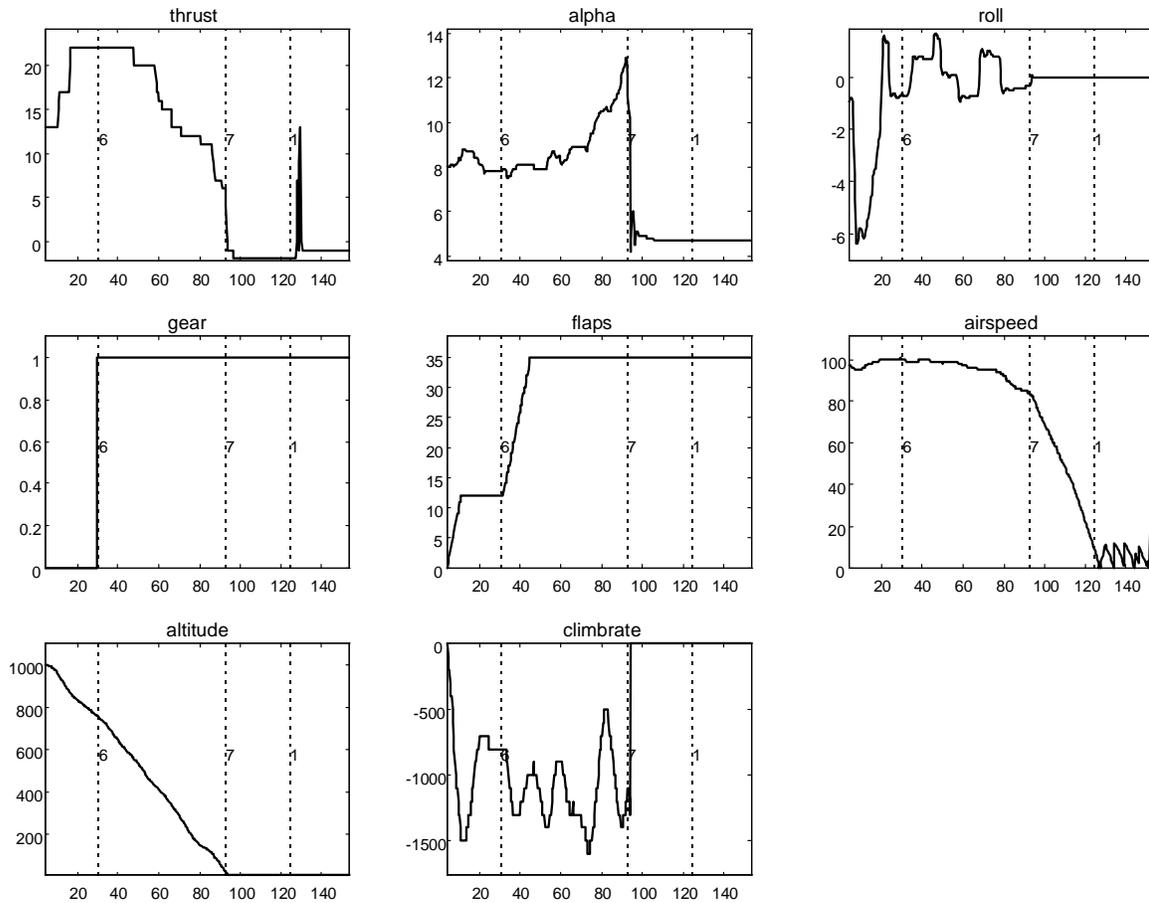
```
function plotdata(fname)
function plotdata(fname, inputnum)
function plotdata(fname, start, stop)
function plotdata(fname, start, stop, inputnum)

function plotdata(fname, mbfname)
function plotdata(fname, mbfname, inputnum)
function plotdata(fname, mbfname, start, stop)
function plotdata(fname, mbfname, start, stop, inputnum)
```

Plots the input data and the pilot specified flight modes

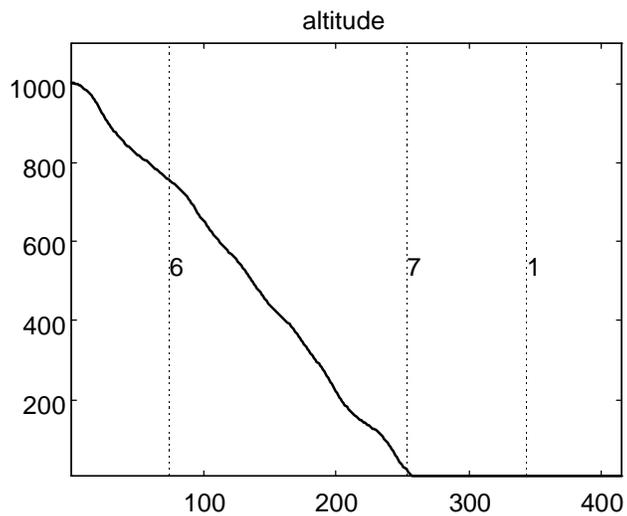
(optional) If inputnum is included, only one variable is plotted  
 (optional) Include start and stop to view windows of data  
 (optional) Include the file name mbfname to have the inferred modes also displayed using the membership functions found in mbfname

```
» plotdata('jeff05.txt')
```

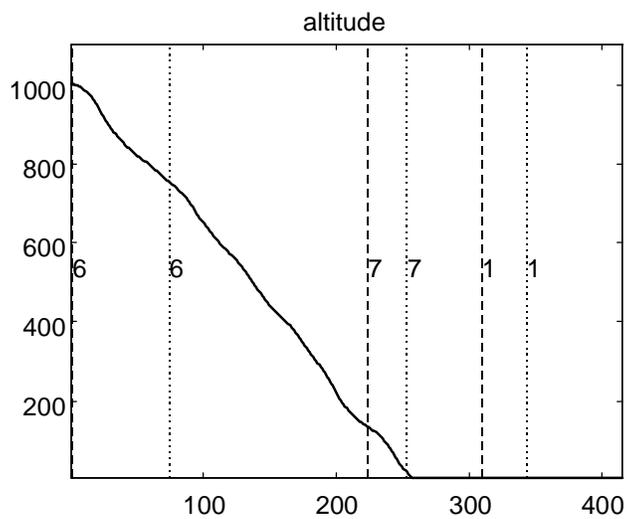


```
» clg
```

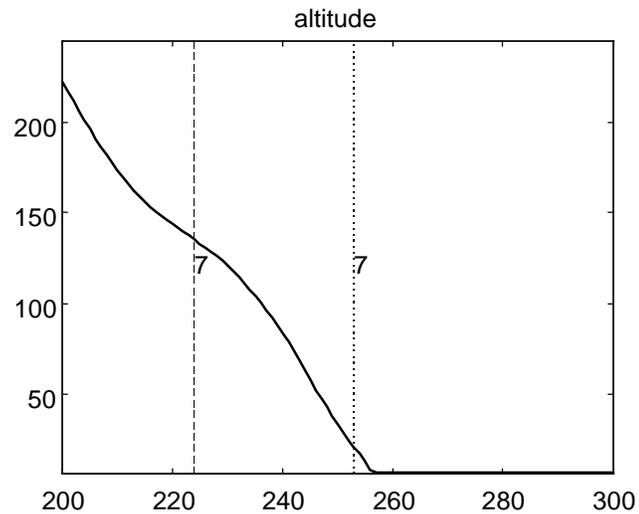
```
» plotdata('jeff05.txt', ALTITUDE)
```



```
» plotdata('jeff05.txt', 'mbfjeff', ALTITUDE)
```



```
» plotdata('jeff05.txt', 'mbfjeff', 200, 300, ALTITUDE)
```



```
»
```

Notice that the last plot shows the Flight Mode Interpreter switched to *landing* around 140 feet, while the pilot waited until he was a little closer to the ground. It also shows that there is no nervousness in the transition.

### ***Plotting Flights***

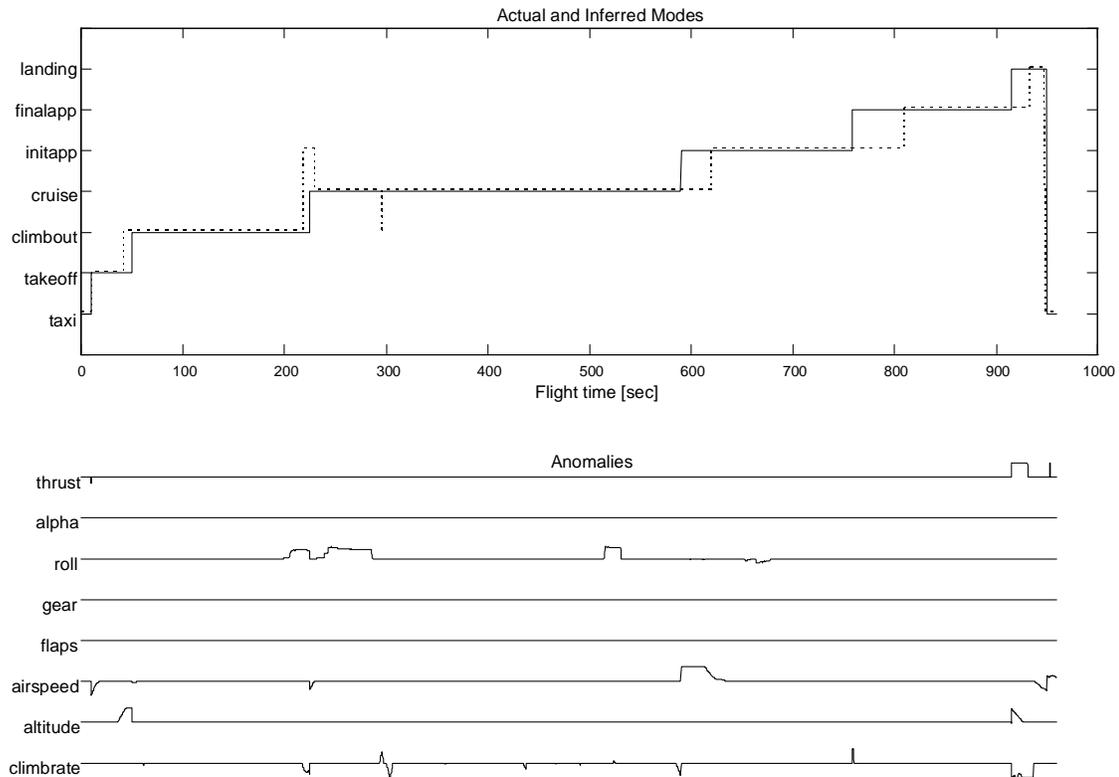
One of the most useful capabilities of the FMI Toolbox is the `plotflt` command. `plotflt` generates a plot of the pilot specified and the FMI inferred flight mode.

```
» help plotflt
```

```
function plotflt(fname, mbfname, start, stop)
```

```
Plots the inferred mode and the pilot entered modes
using the data in fname,
and the membership functions stored in mbfile.
(optional) plot the window from start to stop.
```

```
» plotflt('jeff13.txt', 'mbfjeff')
```



```
»
```

The solid line is the pilot-specified flight mode and the dashed line is the flight mode inferred by the FMI. This example shows two problems in the flight mode interpretation. First the FMI incorrectly infers *initial approach* during the transition from *climbout* to *cruise*. Then, during the *cruise* phase, the FMI momentarily switches to *climbout*. In addition to the flight mode plot, `plotflt` helps the designer understand which inputs caused the FMI to make such errors. The plot labeled “Anomalies” shows which inputs did not match the pilot specified flight mode. For example, in the transition from *climbout* to *cruise*, the roll angle and *climbrate* seem to have contributed to the incorrect inference.

The FMI Toolbox has been a valuable tool for tuning the one-dimensional *ASTRA* Flight Mode Interpreter. However, the MATLAB implementation of the FMI is not identical to the current C++ implementation. The *ASTRA* system includes a Navigation module which makes distance calculations based on a flight plan. These distances can not be included in the MATLAB FMI until a Navigation module is implemented in MATLAB.

## APPENDIX C

### HFMF MATLAB™ TOOLBOX

#### BACKGROUND

The HFMF MATLAB™ Toolbox is a set of MATLAB functions for designing and analyzing hypertrapezoidal fuzzy membership functions (HFMF). HFMFs are a convenient method of specifying multidimensional fuzzy relationships. An HFMF system partitions a state space into fuzzy sets using relatively few parameters. This toolbox takes advantage of the plotting capabilities of MATLAB to help the HFMF designer visualize the membership functions in an HFMF system. The described toolbox has assisted in the design of HFMFs for flight mode interpretation in the *ASTRA* system.

#### REQUIREMENTS

A correctly installed version of MATLAB, of course, is the first requirement for using the HFMF Toolbox. The HFMF Toolbox does not require any additional toolboxes. It will work with the Student Edition of MATLAB. However, the Student Edition of MATLAB is “limited to 8192 elements, with either the number of rows or columns limited to 32” [37]. This limitation will affect the size of any data used to test the HFMF system, but should not limit the HFMF calculations.

To use the HFMF toolbox, the toolbox files can be installed in any directory. At the MATLAB command line, change to the directory containing the files. For example,

```
» cd \users\wally\matlab\hfmf
» dir

.          dom.m      muij.m      ndmake.m   plot2d.m   sample1.m
..         gapats.m   ndadd.m     nearpnt.m  ppndmat.m  sample2.m
combine.m  hfmf.m           ndinfo.m    plot1d.m   rhoij.m    sample3.m

»
```

#### COMMANDS

##### *Help*

The functions available in the HFMF Toolbox can be listed by typing `hfmf` at the MATLAB command prompt:

```
» help hfmf
```

HFMF MATLAB Toolbox  
Version 1.0, by Wallace Kelly

```

THE MEMBERSHIP FUNCTIONS -----
ndmake  make a new hypertrapezoidal matrix
ndadd   add a lambda value to a hypertrapezoidal matrix
ndinfo  get and set information in a hypertrapezoidal matrix
ppndmat pretty print the information in a hypertrapezoidal matrix

PLOTTING FUNCTIONS -----
plot1d  plots a one-dimensional slice of hypertrapezoidal functions
plot2d  plots a two-dimensional slice of hypertrapezoidal functions

FUZZY INFERENCE -----
dom     calculate the degree of membership of a point
muij    calculate an intermediate value for dom
rhoij   calculate an intermediate value for muij
nearpnt find the nearest lambda point to a given point

```

»

Help with individual functions can be viewed by typing `help function-name` at the command line:

» `help nearpnt`

```

function [npts, dxc] = nearpnt(x, lams)
Find the nearest point on the vectors connecting
the hypertrapezoidal lambdas to the point x
and the distance to that vector lambda

x has numsamples rows, and numdimensions columns
lams has numlambdas rows, and numdimensions columns
npts has numlambdas rows, and numdimensions columns
dxc are the distance to the vector lambda

```

»

The `nearpnt` function finds the closest lambda and the distance to that lambda from a point (or an array of points), `x`. This function actually allows for “vector lambdas”, which are a set of lambdas all serving as prototype points for a single fuzzy set.

### ***HFMF Matrices***

A system of HFMFs is internally represented in the HFMF Toolbox by a matrix. The matrix stores the number of dimensions, number of sets, crispness factor, labels, and all the prototype points. An HFMF matrix is created with `thendmake` command:

» `help ndmake`

```

ndmat = ndmake(N, sigma, name1, name2, ... name8)

```

Make a new N-dimensional hypertrapezoidal matrix  
 N is the number of dimensions  
 sigma is the crispness factor  
 ndmat is the n-dimensional hypertrapezoidal matrix  
 the names are optional, and are the labels of the dimensions

Example: `hfmf = ndmake(2, 0.5, 'airspeed', 'altitude')`

```
» hfmf = ndmake(3, 0.5, 'airspeed', 'altitude', 'power');
»
```

To see the information stored in an HFMF matrix, use the `ppndmat` command. The `ppndmat` command “pretty prints” the HFMF information in a convenient format:

```
» help ppndmat
```

```
ppndmat(ndmat)
Pretty print the information in the
hypertrapezoidal membership function
```

```
» ppndmat(hfmf)
```

```
Dimensions: 3
Sets: 0
Crispness: 0.50
Domain names:   airspeed   altitude   power
»
```

Information about the HFMF partition is also available through `thendinfo` command.

```
» help ndinfo
```

NDINFO is used to get and set information in an HFMF matrix

```
These can modify ndmat
ndmat = ndinfo(ndmat, 'sets|sigma', value)
ndmat = ndinfo(ndmat, 'scale', scale)
ndmat = ndinfo(ndmat, set#, lambda#, lam)
```

```
These just return information
[N M sigma] = ndinfo(ndmat)
value = ndinfo(ndmat, 'max|min', domain#)
name = ndinfo(ndmat, 'domain', domain#)
domain# = ndinfo(ndmat, 'domain', 'name')
name = ndinfo(ndmat, 'set', set#)
set# = ndinfo(ndmat, 'set', 'name')
lam = ndinfo(ndmat, 'lambda', set#)
nlam = ndinfo(ndmat, 'nlambda', set#)
```

Return or change the information stored  
 in an n-dimensional hypertrapezoidal matrix

N is the number of dimensions  
 M is the number of sets

sigma is the crispness factor

ndmat is an n-dimensional hypertrapezoidal matrix

lam are the lambdas that define the ith set

nlam are the normalized lambdas

value is the new value to assign to the property

set is the fuzzy set of interest

scale is Mx2 matrix of [min, max] pairs for normalizing

```
» [N M sigma] = ndinfo(hfmf)
```

```
N = 3
```

```
M = 2
```

```
sigma = 0.5000
```

```
»
```

ndinfo can also be used to set parameters in the HFMF matrix.

```
» hfmf = ndinfo(hfmf, 'sigma', 0.75); ppndmat(hfmf)
```

```
Dimensions: 3
           Sets: 2
Crispness: 0.75
Domain names:  airspeed  altitude  power
InHanger:      0.00      0.00      0.00
InSky:         100.00    1000.00  60.00
              150.00    3000.00  100.00
»
```

### ***Prototype Points***

To add prototype points to an HFMF matrix, use the ndadd command. Prototype points are expressed as a row of elements matching the inputs specified in the ndmake command.

```
» help ndadd
```

```
ndmat2 = ndadd(ndmat1, lam)
```

```
ndmat2 = ndadd(ndmat1, m, lam)
```

Add a lambda value to the n-dimensional hypertrapezoidal matrix

ndmat1 is the original n-dimensional hypertrapezoidal matrix

lam is the new lambda value

m is the set name with which to associate this lambda

```
» hfmf = ndadd(hfmf, 'InHanger', [0, 0, 0]);
```

```
» hfmf = ndadd(hfmf, 'InSky', [100, 1500, 60]);
```

```
» ppndmat(hfmf)
```

```
Dimensions: 3
           Sets: 2
Crispness: 0.50
Domain names:  airspeed  altitude  power
```

```

    InHanger:    0.00    0.00    0.00
    InSky:      100.00   1500.00   60.00
»

```

Additional lambda points can be added to existing sets to create vector lambdas.

```

» hfmf = ndadd(hfmf, 'InSky', [150, 3000, 100]);
» ppndmat(hfmf)

Dimensions: 3
Sets: 2
Crispness: 0.50
Domain names:  airspeed  altitude  power
InHanger:      0.00      0.00      0.00
InSky:         100.00    1500.00   60.00
               150.00    3000.00  100.00
»

```

### ***Calculating Degrees of Membership***

The command `dom` is used to calculate the degrees of membership in the fuzzy sets of an HFMMF partition. The following example shows how to determine the degrees of membership in *InHanger* and *InSky* for airspeed = 10, altitude = 0, and power = 20. It shows that for these inputs, the degree of membership in *InHanger* is 0.9272.

```

» help dom

function mu = dom(x, ndmat, product)

Returns the degree of membership of points x
using the hypertrapezoids defined in ndmat

x is numsamples by dimensions
mu is numsamples by dimensions
if product is included, product inference is used

» mu = dom([10 0 20], hfmf, 1)

mu = 0.9272    0.0728
»

```

`dom` can also determine the degrees of membership for an entire set of samples. The following example shows how to determine the degrees of membership as airspeed, altitude, and power increase:

```

» x = [(0:15:150)', (0:300:3000)', (0:10:100)']

x =
     0         0         0
    15        300        10
    30        600        20
    45        900        30
    60       1200        40
    75       1500        50

```

90	1800	60
105	2100	70
120	2400	80
135	2700	90
150	3000	100

```
» mu = dom(x, hfmf, 1)
```

```
mu = 1.0000      0
      0.9337     0.0663
      0.7006     0.2994
      0.4676     0.5324
      0.2346     0.7654
      0.0172     0.9828
      0          1.0000
      0          1.0000
      0          1.0000
      0          1.0000
      0          1.0000
```

```
»
```

Notice that as the variables increase, the degree of membership decreases for *InHanger* and increases for *InSky*. Also notice that the degrees of membership sum to unity, a design requirement of the HFMF derivation.

### ***Plotting HFMFs***

There are two functions for plotting hypertrapezoidal fuzzy membership functions – `plot1d` and `plot2d`. `plot1d` plots the degrees of membership for the fuzzy sets as a single variable is varied. Consequently, `plot1d` asks the user for values for the other variables.

```
» help plot1d
```

```
function plot1d( ndmat, i, numpoints, product )
```

```
Plot a one-dimensional slice of the hypertrapezoidal
membership function described in ndmat.
```

```
Plots the ith dimension.
```

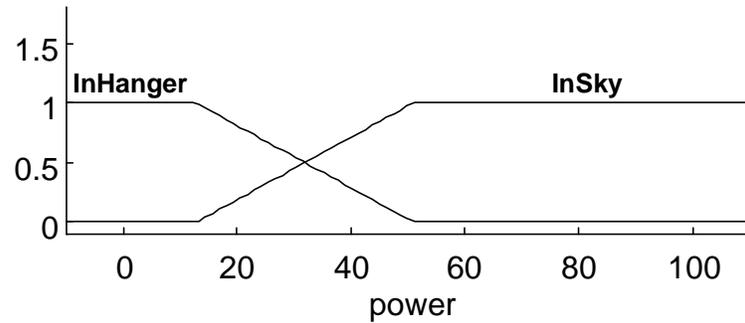
```
If product is included, product inference is used.
i, numpoints, and product are optional.
```

```
» plot1d(hfmf, 3)
```

```
Dimensions: 3
Sets: 2
Crispness: 0.75
Domain names:  airspeed  altitude  power
InHanger:      0.00      0.00      0.00
InSky:         100.00    1000.00  60.00
               150.00    3000.00  100.00
```

```
Enter a value for airspeed [75.000]: 60
```

Enter a value for altitude [1500.000]: 0



»

plot2d is similar to plot1d, except it allows two variables to be varied on the same plot.

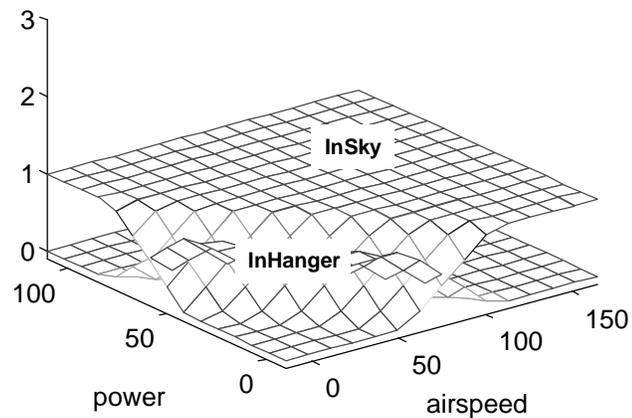
» plot2d(hmf, 1, 3)

```

Dimensions: 3
Sets: 2
Crispness: 0.75
Domain names:   airspeed   altitude   power
InHanger:      0.00       0.00       0.00
InSky:         100.00     1000.00    60.00
               150.00     3000.00    100.00

```

Enter a value for altitude [1500.000]:



»

## VITA

Wallace Eugene Kelly, III received his B.S. and M.S. degrees in Electrical Engineering from Texas A&M – Kingsville in 1993 and 1994, respectively. His technical interests have included speech and image signal processing, control systems, neural networks, fuzzy logic, object-oriented programming, network computing, and more recently, smart-cockpit technologies. His Ph.D. studies at Texas A&M have provided him the opportunity to explore new theoretical concepts, and apply those concepts in practical applications. During the course of his studies, he taught Electrical Engineering courses as an assistant lecturer, and worked on two funded research projects. One of those projects was the *ASTRA* program described in this dissertation. The other involved methods of autonomous monitoring of space missions and was funded by a major NASA contractor. Wallace plans to graduate with his Ph.D. in Electrical Engineering from Texas A&M in August 1997. Upon graduation, Wallace will join the Advanced Technology Center of Rockwell Collins' Avionics and Communications Division, in Cedar Rapids, Iowa.

Wallace is married to Terri Lynn Kelly. They have a two year old son, Andrew, and are expecting their second child in November 1997. Their permanent mailing address is 4102 Wyndale, Corpus Christi, Texas 78418.