FUZZY FUZING LOGIC FOR UNMANNED AIR COMBAT VEHICLES

A Project

Presented

to the Faculty of

California State University, Chico

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

© Terrence L. Munsinger 2007

Fall 2007

FUZZY FUZING LOGIC FOR UNMANNED AIR COMBAT VEHICLES

A Project

by

Terrence L. Munsinger

Fall 2007

APPROVED BY THE DEAN OF THE SCHOOL OF
GRADUATE, INTERNATIONAL, AND INTERDISCIPLINARY STUDIES:

Susan E. Place, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

Ralph Hilzer, Chair

Benjoe A. Juliano, Ph.D.

PUBLICATION RIGHTS

No portion of this project may be reprinted or reproduced in any manner unacceptable to the usual copyright restrictions without the written permission of the author.

DEDICATION

I would like to thank Ralph Hilzer, and Benjoe Juliano, for the guidance and encouragement toward the preparation of this paper.

I also owe my wife, Laura, much gratitude and thanks for the encouragement and understanding she has provided throughout my studies at California State Chico over the last ten years.

TABLE OF CONTENTS

PAGE

CHAPTER

Appendix

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

ABSTRACT

FUZZY FUZING LOGIC FOR UNMANNED AIR COMBAT VEHICLES

by

© Terrence L. Munsinger 2007

Master of Science in Computer Science

California State University, Chico

Fall 2007

This project studies the incorporation of a fuzzy logic controller into a fuze data processing system, to control the fuze arming and safing functions. The fuzzy logic controller determines the rate at which a fuze firing train interrupter, or rotor, is rotated from a safe condition to an armed condition. The fuzzy logic controller can also rotate a rotor, which is in the armed condition, back to the safe condition, if a fuzed munition re-enters the launcher's safe separation envelope.

The purpose of this project is to develop and demonstrate graphically that a fuze, if fed the appropriate positional data, can determine if a safe-separation from a launcher has been achieved prior to the fuze reaching an armed condition and can return the fuze to a safe condition if the fuzed munition has re-entered the launcher's safe separation envelope. The fuzzy logic controller also controls the rate of arming to assure the safety of the launcher and the reliability of the munition.

CHAPTER I

INTRODUCTION

This chapter introduces the reader to the problem statement, purpose, definitions and limitations applicable to the task of implementing a Fuzzy Logic Controller (FLC) into the decision making process of a fuze.

Problem Statement

Throughout the history of mankind, the advancement of munition technologies by a political entity has usually been countered with an advancement of similar technologies by an opposing political entity. Recent engagements have demonstrated that Unmanned Air Vehicles (UAV) can serve as launch platforms for munitions; in effect UAVs have become Unmanned Combat Air Vehicles (UCAV).

Eventually UCAV technologies will advance to the point that both sides in a conflict will have a UCAV munition launching capability. These advancements will make air-to-air UCAV defensive capabilities necessary. Because UCAV maneuverability is not limited by the gravitational units (“g”s) that a human pilot can endure, future UCAV versus UCAV engagements will involve high “g” maneuvering to avoid each other’s air intercept missiles (AIM). The AIM carried by future UCAV must be capable of higher “g” level maneuvers in order to be able to intercept the target.

During a high “g” maneuvering engagement (dogfight) between UCAVs, it is possible for a launched AIM to leave the safe separation envelop of the launch UCAV (hereafter referred to as the “launcher”), reach the armed condition, and then while pursuing the target UCAV (hereafter referred to as the “target”), the AIM could re-enter the launcher’s safe separation envelop in an armed state. This is an unacceptable situation, which affects the launcher’s safety.

To assure mission safety for the launcher, the munition must incorporate a fuze controller that assures 1) the munition is not armed until a safe separation distance from the launcher is achieved, and 2) that if the munition re-enters the safe-separation envelope of the launcher, the fuze will return to the safe condition.

Purpose of the Project

This project establishes a need for and graphically demonstrates a FLC, which determines if safe separation distance from the launcher has been achieved by the AIM prior to the fuze reaching the armed condition and returns the fuze to the safe condition if the munition has re-entered the safe separation envelope of the launcher. The FLC is also used to control the rate of arming to assure the safety of the launcher and the reliability of the mission.

The FLC processes three inputs concerning the closure rate between the AIM and the launcher, the distance of the AIM to the launcher, and the distance of the AIM to the target. The FLC’s output is an arming rate, which is defined as a positive number when the fuze is progressing towards the armed condition and a negative number when the fuze is progressing towards the safe condition.

Limitations of the Project

This project is only designed with a level of complexity that is required to demonstrate the control of a fuze safe and arm functions. An electronic version of this concept is included in the pending patent, but for clarity, the mechanical version is used for this project.

There are many different pursuit or intercept courses that AIMs can use (e.g., deviated, constant bearing, beam rider, and proportional navigation) but for this project the pure pursuit course is used. A pure pursuit course is one where the pursuer constantly turns to head directly towards the target.

The graphic demonstrator called “Fuzzy Fuzing” is limited to 2 axes (X and Z) for simplicity. Even though transformational matrix equations are known to the author, simplified trigonometry equations were used in the graphic demonstrator to provide realistic animation to the launcher, AIM, and the target.

Chapter II, Literature Review, presents more information concerning fuzing and guidance theory and an introduction to Fuzzy Logic. The methodology of demonstrating the concept in discussed in Chapter III. Chapter IV presents the summary and conclusions, which consist of a presentation of the findings, research questions, conclusions relevant to the research questions, limitations of study design and procedures, and future research and recommendations.

CHAPTER II

LITERATURE REVIEW

Introduction

This chapter contains introductory information concerning fuzing theory, guidance theory, an introduction to fuzzy logic, and proposes a hypotheses that is used by the project.

Fuzing Theory

The *Weapons Systems Fundamentals-Analysis of Weapons, NAVWEPS OP 3000 Volume 2* [1] defines a fuze as the component of a munition’s warhead section or payload, which causes the detonation of the payload at the precise moment in time to cause the maximum amount of lethality against the intended target. It also introduces the concept of Safety and Arming in fuze design. The document establishes most of the basic fuze safety design criteria used in today’s fuze specifications.

A fuze has three distinct functions: 1) assures safety of the launcher and munition until the it has left the safe separation envelop of the launcher and reaches the target, 2) aligns the fuze explosive firing train elements, which enables the munition to be initiated upon receipt of a firing command from the TDD, and 3) initiates the explosive firing train elements and thusly the munition [1].

Another way of defining a fuze, is to group the functions of numbers 1) and 2) above into a Safe and Arm (S&A) device with the function of number 3) into a Target Detection Device (TDD) [1]. Basically the combination of an S&A device with a TDD produces what is commonly known as a fuze. This paper focuses primarily on the S&A device portion of a fuze.

Safety and Arming

The first function of the S&A device is to prevent inadvertant arming of the fuze until a pre-programmed or designed sequence of environmental events have been sensed and compared. Once this sequence of events has been sensed, the S&A's second function is to remove the safety features and progress into the armed condition by aligning the elements of the firing train [2]. These firing train elements consist of the primary explosive (detonator) and the secondary explosive (lead).

Upon receipt of a firing command from the TDD, the fuze's firing train will initiate the warhead causing maximum lethality against the intended target.

The S&A must also be able to preclude arming or initiation of the fuze firing train, while withstanding other external forces including vibration, acceleration, electro-magnetic radiation, heat, and impact shock [1].

Safe Separation

To assure maximum safety for the user of today's modern munition systems, the S&A must also implement a method that assures the armed condition is not attained until a safe separation is established between the launcher and the munition.

The safe separation between the launcher and the launched munition, precludes inadvertent blast and/or fragmentation damage to the launcher. This requirement is usually met by controlling the time that arming is started and by the time it takes for the S&A to progress from the safe condition to the armed condition.

Due to the probable high "g" maneuvering ability of future munitions, it is possible that an armed munition could re-enter the launcher's safe separation envelope due to the combined vectors of the launcher, the target, and the AIM.

By using an FLC to calculate the arming rate of an S&A the integrity of the launcher's safe separation envelope can be maximized while at the same time ensuring that maximum lethality is delivered to the target.

Description of Simulated S&A Device

The graphic demonstration uses a generic electro-mechanical S&A device design, which meets the requirements of MIL-STD-1316E [2] (see Figure 1).

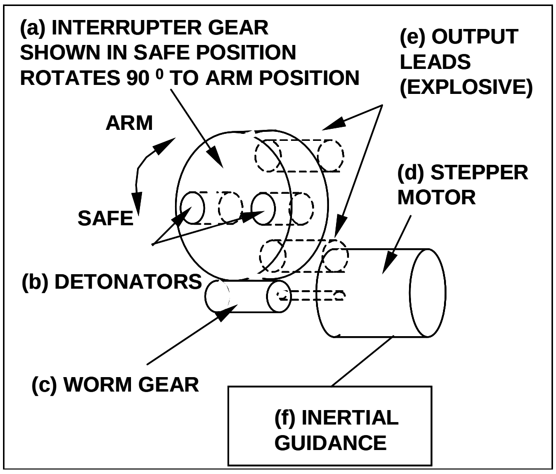


Fig. 1. Safety and arming device.

The interrupter gear (a), which is shown in the safe condition, is rotated by a stepper motor (d) via a worm gear (c). When the interrupter gear is rotated 90 degrees, the detonators (b) are aligned with the explosive output leads (e). If the fuzes inertial guidance (f) determines that the munition has re-entered the launch platform's safe separation envelope, it will command the stepper motor (d) to re-safe the interrupter (a) until safe separation is once again achieved when it will re-arm the interrupter.

Guidance Theory

The *Weapons Systems Fundamentals-Analysis of Weapons, NAVWEPS OP 3000 Volume 2* discusses variable flight paths for pursuit courses. On page 187 this reference defines the criterion for a pure pursuit course as "The criterion for the course is that the missile always heads directly toward the present target position. One important aspect of the pure pursuit course is best demonstrated by studying the flight path as seen from the target" [1].

The pure pursuit course concept is used by the Visual C++, "Fuzzy Fuzing" program developed for this project. The pure pursuit course is calculated by determining the yaw angle between the AIM and the target and setting the AIM on a course that heads directly towards the target. This provides a realistic animation of an intercept.

Fuzzy Logic

Fuzzy Logic was founded by Lotfi Zadeh, who was a mathematician and professor at UC Berkeley, during the 1960s [3]. Fuzzy Logic allows the user to determine the relative importance in a sysyem between precision and significance. Mr. Zadeh was quoted as saying, "As complexity rises, precise statements lose meaning and meaniful statements lose precision" [3]. Fuzzy Logic tends away from the hard lines of classical logic which demand either a true or false answer to a question, when in reality any question actually has an answer that has a membership value that is partially true and partially false. Mr. Zadeh developed the methods used to incorporate fuzzy logic into any control system. These methods allow the controller to determine just how much precision in a system is necessary and when a rough answer will have enough significance to assure proper operation of the system. A brief description of these methods follows along with how they have been implemented into the Fuzzy Fuze Arming Rate Controller.

Equations and methods from the Fuzzy Logic [3] textbook will be used to develop a FLC used to obtain data from the member functions and to determine what the arming rate of the fuze should be during target intercept maneuvers to assure safe separation distances are maintained.

Hypotheses

The hypotheses for this project is that Fuzzy Logic can be incorporated into the design of a fuze S&A device to achieve a safe and reliable fuzing system for a highly maneuverable munition system.

CHAPTER III

METHODOLOGY

Introduction

This project focuses primarily on the fuzzy logic control of the arming and safing functions of an AIM fuze S&A device. This section describes a simulated engagement scenario and the Fuzzy Fuzing Safe/Arm Rate Controller (FFSARC), whose purpose is to adjust the arming rate of the AIM S&A during target interception.

The FFSARC primarily perform two functions: 1) To rotate the S&A interrupter to the armed condition when the AIM has been launched and is moving away from the launcher, thereby reaching a safe separation distance, and 2) To rotate the S&A interrupter to the safe condition when it is moving towards the launcher and no longer maintains a safe separation distance from the launcher.

This concept can be implemented into any muntion fuzing system, but for clarity, this paper will examine fuzzy logic's contribution to the air-to-air engagement scenario.

The FFSARC system is simulated using Mathwork's MATLAB® Fuzzy Logic Toolbox and a simulated engagement scenario is graphically presented using a Visual C++ based OpenGL program named "Fuzzy Fuzing."

Simulated Engagement Scenario

The “Fuzzy Fuzing” program simulates the safing and arming functions of an electro-mechanical S&A during a two dimensional close-quarters dogfight engagement scenario between two UCAVs (see Figure 2). The target UCAV is on a perpendicular vector to the launch UCAV. After the AIM is launched, it achieves safe separation from the launch UCAV while the S&A has progressed into the armed condition.

Due to the approaching vector of the target UCAV, the AIM re-enters the launch UCAV’s safe separation envelope. The AIM’s S&A’s internal guidance determines its impending re-entery into the safe separation envelope and begins to re-safe the S&A device. Once the AIM leaves the safe separation envelope for the second time and determines it is no longer on a course that will bring it back into the launcher’s safe separation envelope, the S&A will be re-armed in time for interception of the target UCAV.

In Figure 2 the launch platform (r2) and safe separation envelope (a) is shown at AIM (m1) launch (t_0). The AIM is launched with the fuze S&A in the safe condition (b). The AIM (m2) turns toward the target aircraft at q1, using constant bearing navigation and leaves the safe separation envelope with armed fuze (c). Due to the launcher’s and target trajectories, the AIM turns back towards the launch platform’s safe separation envelope (d). The AIM (m3) re-enters safe separation envelope and re-safes fuze (e). While re-safing the S&A, the AIM continues to track the target (q2), safely passes the launcher at r2, leaves safe separation envelope, re-arms fuze at (g) and sends a firing command to the S&A destroying the target at (h).

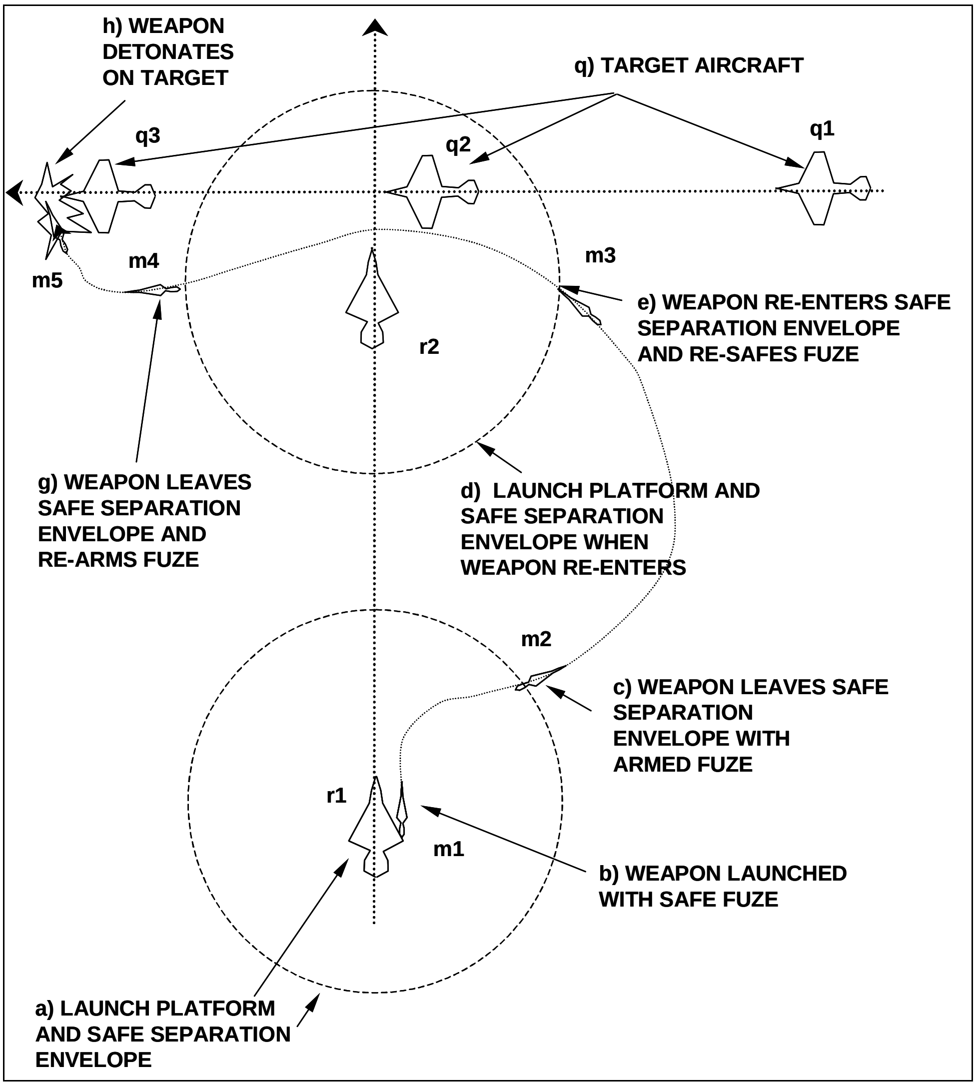


Fig. 2. Simulated engagement scenario.

Fuzzy Fuzing Safe/Arm Rate Controller

This section describes in detail the Fuzzy Fuzing Safe/Arm Rate Controller (FFSARC), which was developed for this project. As stated earlier, the purpose of this fuzzy logic controller is to adjust the arming rate (positive rate is toward the arm state, negative rate is toward the safe state) of a fuze. The FFSARC primarily performs two functions: 1) To have the fuze progress to the armed state when the AIM is launched and is moving away from the launcher and towards the target (thereby reaching a safe separation distance), and 2) To have the fuze progress to the safe state when it is moving towards the launcher and no longer maintains a safe separation distance from the launcher.

Fuzzy Inference System

A fuzzy inference system (FIS) provides the process of mapping inputs into an output using fuzzy logic [4]. The MATLAB® Fuzzy Logic Toolbox's [4] is utilized as the preliminary development tool to verify the FIS is working as intended.

There are two types of fuzzy inference systems: the Mamdani [5] type and the Takagi-Sugeno-Kang type. For this simulation, the Mamdani type is used. The five steps [4] used to develop the Mamdani model include the following:

1) Fuzzification of the input variables. The first step is to determine which inputs are necessary to perform the function and then assign linguistic terms to the values. For instance, the first input used in this study is the distance from the launcher to the AIM. Three triangular member sets are assigned the fuzzy values, "Close," "SafeSep," and "Far." These values range from 0 feet to 2,000 units. The input variables are discussed in greater detail in Section 3.

2) Application of the fuzzy operators (AND or OR). After the input variables have been assigned fuzzy values or are "fuzzified," the antecedents (if there are more than one) of the rules are assigned an operator such as "AND" or "OR." The operator uses the fuzzified antecedant and its output is applied to the output function. The Fuzzy

Logic Toolbox supports two AND functions and two OR functions. For AND the toolbox provides: “minimum” and “product” operations, and for the OR function, it provides: “maximum” and “probablistic or.” The minimum function was used for the AND operator and the maximum function was used for the OR operator in this project.

3) Implications from the antecedent to the consequence. The fuzzy operator is now applied to the membership function graphs and the result, if there is an intersection, is a truncated shape of the intersected chart. The input is a single number and the result is a portion of the original fuzzy set.

4) Aggragation of the consequences across the rules. All of the truncated fuzzy set output from the last step are now aggregated into one final fuzzy set for all of the input variables.

5) Defuzzification. The final step, in the Mamdani method, is to use the Standard Additive Model (SAM). The “defuzzification” step results in a final output expressed as a single number.

The FFSARC FIS is shown in Figure 3. This figure was developed with the MATLAB® Fuzzy Logic Toolbox’s Fuzzy Inference System Editor.

The FFSARC’s FIS is composed of three (3) inputs, which are titled: 1) Launcher to AIM distance (LAD), 2) Launcher to AIM closure rate (LACR), and 3) Target to AIM distance (TAD). The Mamdani Arm Rate Controller fuzzifies the inputs and outputs the defuzzified Arm Rate as a number between -1 and 1. Negative one (-1) signifies “FastSafe” and positive one (+1) signifies “FastArm.”

The FFSARC has three membersip functions, which correspond to the three inputs mentioned above. Figures 4 through 8 describe these membership functions.

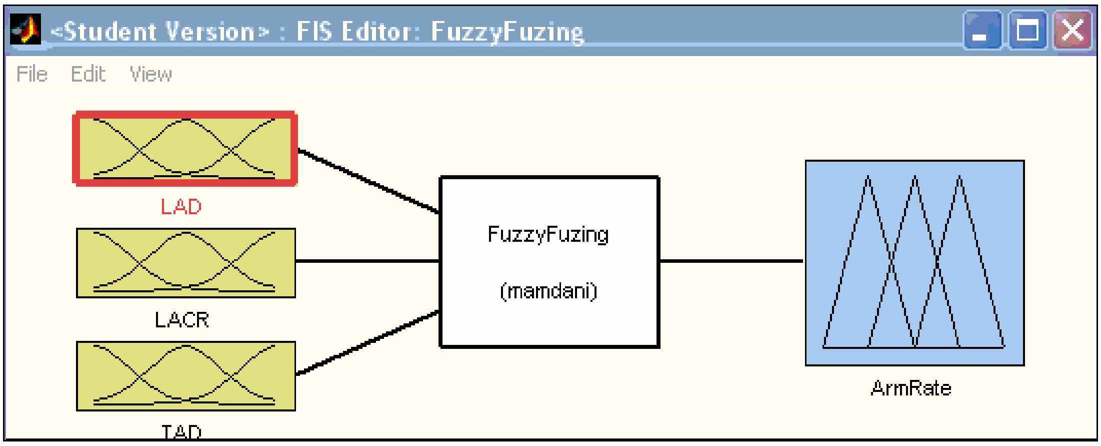


Fig. 3. FFSARC FIS.

The LAD membership function (Figure 4) is composed of three (3) fuzzy sets titled: “Close,” “Safe Sep,” and “Far.” The horizontal axis, which represents distance

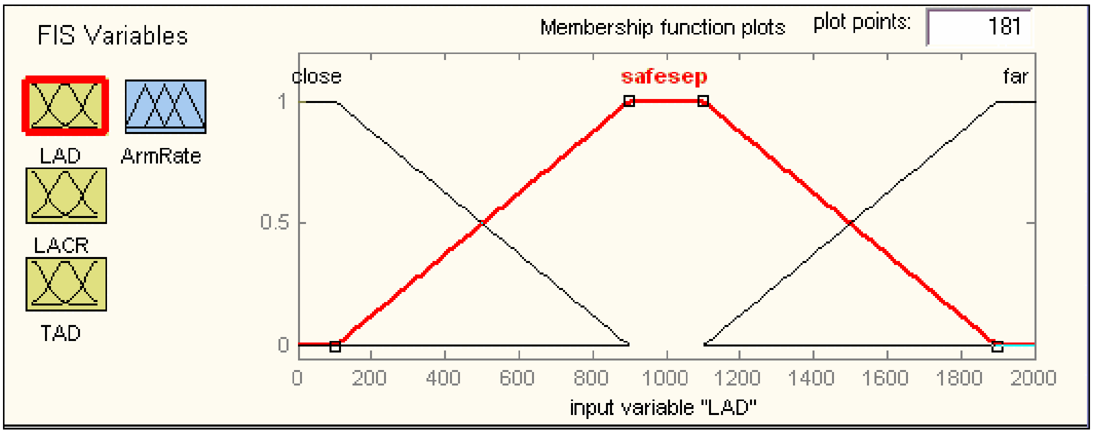
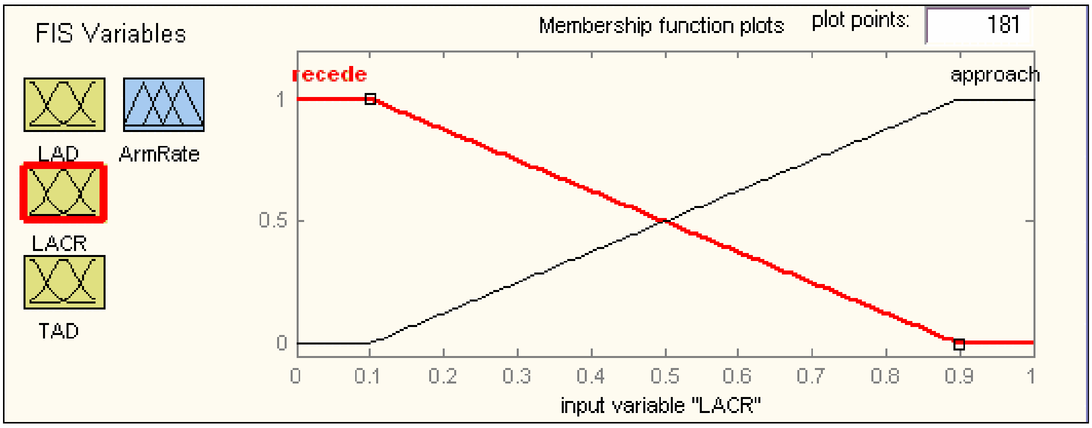


Fig. 4. Launcher to AIM distance (LAD) membership function.

from the launcher, ranges from 0 to 2,000 units. The vertical axis, which represents membership of any particular distance, ranges from 0 to 1. For example a distance of 200 feet is estimated to have a membership value of about 85% (or 0.85) in the “Close” fuzzy set and about 15% (or 0.15) in the “SafeSep” membership value. The three fuzzy sets are represented by trapezoids.

The LACR membership function (Figure 5) is composed of two (2) fuzzy sets titled: “Recede,” and “Approach.” The horizontal axis, which represents closure rate from the launcher, ranges from 0 to 1. The vertical axis, which represents membership of any particular distance, ranges from 0 to 1. For example a closure rate of 0.95 indicates the AIM is approaching the launcher with a membership in the approach set of 1.0 and a membership in the recede set of 0.0.



plot points: 181

Fig. 5. Launcher to AIM closure rate (LACR) membership function.

The TAD membership function (Figure 6) is composed of two (2) fuzzy sets titled: “Close” and “Far.” The horizontal axis, which represents distance from the launcher, ranges from 0 to 2,000 units. The vertical axis, which represents membership of any particular distance, ranges from 0 to 1. For example a distance of 400 units can be estimated to have a membership value of about 85% (or 0.85) in the “Close” fuzzy set and about 15% (or 0.15) in the “Far” membership value.

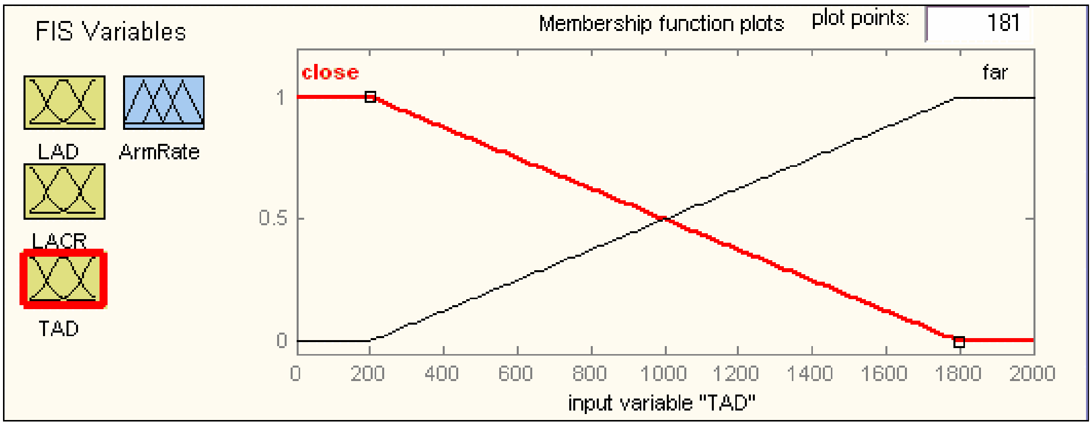


Fig. 6. Target to AIM distance (TAD) membership function.

The Arm Rate output membership function (Figure 7) is composed of seven (7) trapezoidal membership functions titled: “FastSafe,” “Safe,” “SlowSafe,” “noChange,” “SlowArm,” “Arm,” and “FastArm.” The horizontal axis represents decreasing to increasing angular rotation of the worm gear that drives the interrupter gear of the S&A mechanism (see Figure 1). The horizontal axis ranges from negative one (-1) to positive one (+1) and represents rotational speed of the rotor. A negative value

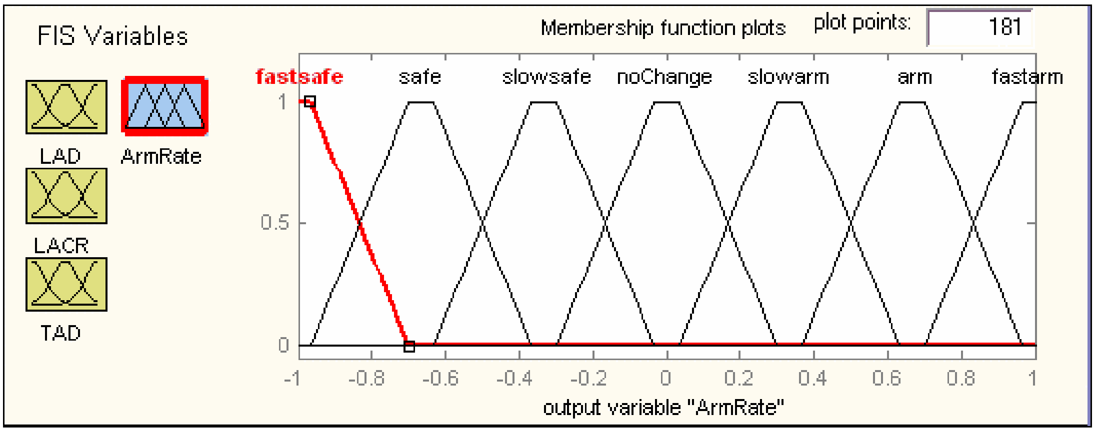


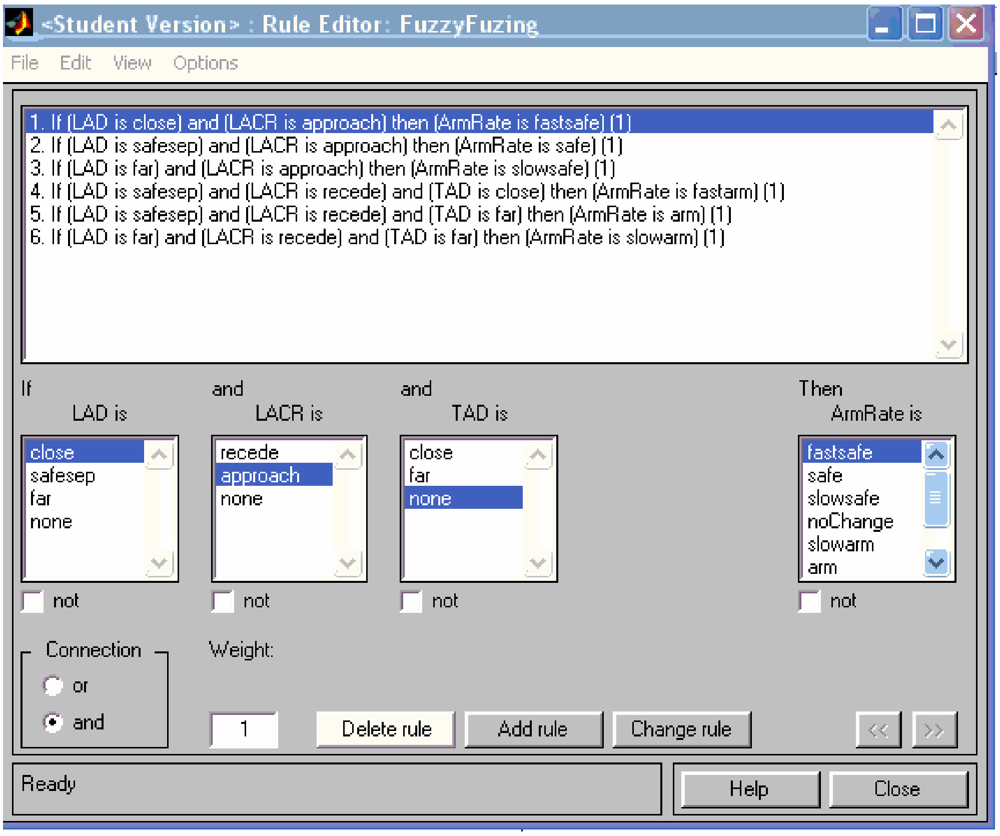
Fig. 7. Arm rate output membership function.

represents rotation toward the safe condition and a positive value represents rotation of the interrupter toward the armed condition. The vertical axis, which represents membership of any particular arming rate, ranges from 0 to 1.

Fuzzy Fuzing Safe/Arm Rate Controller

Rules

The next step in the development of the FFSARC FIS is to use the assigned membership functions to establish the fuzzy operators and compose rules using the MATLAB® Fuzzy Logic Toolbox's Fuzzy Inference System Editor. Figure 8 shows the FIS Rule Editor.



1. If (LAD is close) and (LACR is approach) then (ArmRate is fastsafe) (1)
2. If (LAD is safesep) and (LACR is approach) then (ArmRate is safe) (1)
3. If (LAD is far) and (LACR is approach) then (ArmRate is slowsafe) (1)
4. If (LAD is safesep) and (LACR is recede) and (TAD is close) then (ArmRate is fastarm) (1)
5. If (LAD is safesep) and (LACR is recede) and (TAD is far) then (ArmRate is arm) (1)
6. If (LAD is far) and (LACR is recede) and (TAD is far) then (ArmRate is slowarm) (1)

Fig. 8. FIS rule editor.

The FIS Rule Editor allows the user to quickly and efficiently establishes the set of rules by using the convenient pull down menus containing the variables from all of the fuzzy membership sets constructed earlier.

The following rules were established using the Rule Editor, for implementation into the FIS:

1) If LAD is Close, and LACR is Approach, then ArmRate is FastSafe.

2) If LAD is Safesep, and LACR is Approach, then ArmRate is Safe.

3) If LAD is Far, and LACR is Approach, then ArmRate is SlowSafe.

4) If LAD is Safesep, and LACR is Recede and the TAD is Close, then ArmRate is FastArm.

5) If LAD is Safesep, and LACR is Recede, and the TAD is Far, then ArmRate is Arm.

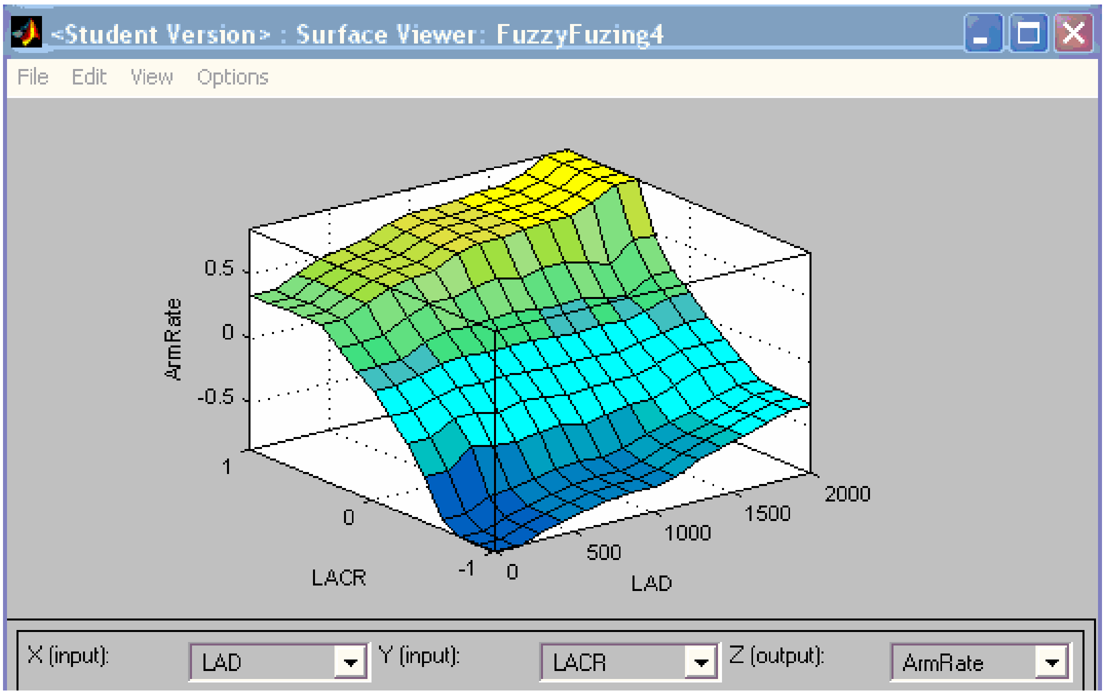
6) If LAD is Far, and LACR is Recede, and the TAD is Far, then ArmRate is SlowArm.

Upon close inspection of the six rules above it is apparent that the safety of the launcher takes precedence over the reliability of the mission. The first three rules only consider the approach and distance of the AIM to the launcher. Rule numbers four through six are concerned with arming while the AIM is receding from the launcher. It adjusts the arming rate depending on the distance from the target. Also the order of the rules does not affect the final output solution.

FIS Surface Plot Feature

After the rules have been established, a surface plot of the FFSARC FIS can be viewed, via the FIS GUI.

The FFSARC Surface Plot (Figure 9) allows the user to view a three-dimensional plot of the fuzzy logic controller. This plot shows the two inputs: launcher to AIM distance (LAD), and the closure rate between the launcher and the AIM. Notice that the Arm Rate is negative (safing) when the SafeSep is less than 1000 and the AIM is approaching (separation is decreasing). The Arm Rate is positive (arming) when the SafeSep is more than 1000 and the AIM is receding (separation is increasing). This is what was expected during the set-up of the FIS discussed in the preceding paragraphs.



X (input): LAD
Y (input): LACR
Z (output): ArmRate

Fig. 9. FFSARC surface plot LAD and LACR.

The plot depicted in Figure 10 shows two inputs: the closure rate between the launcher and the AIM, and the target to AIM distance (TAD). The Arm Rate is output.

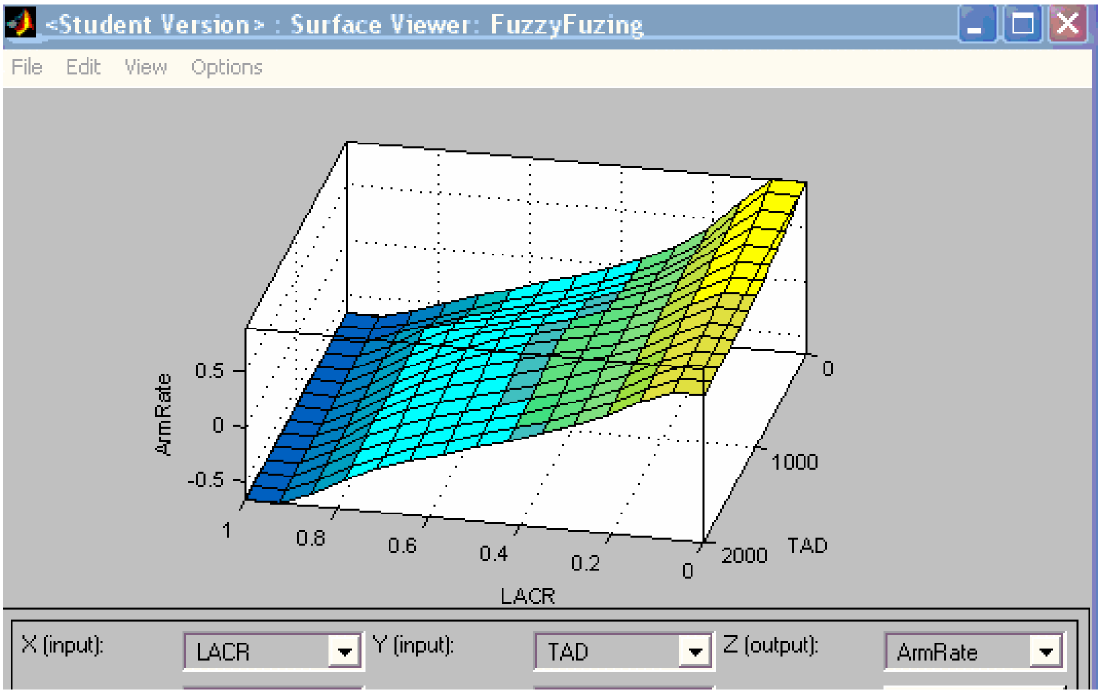


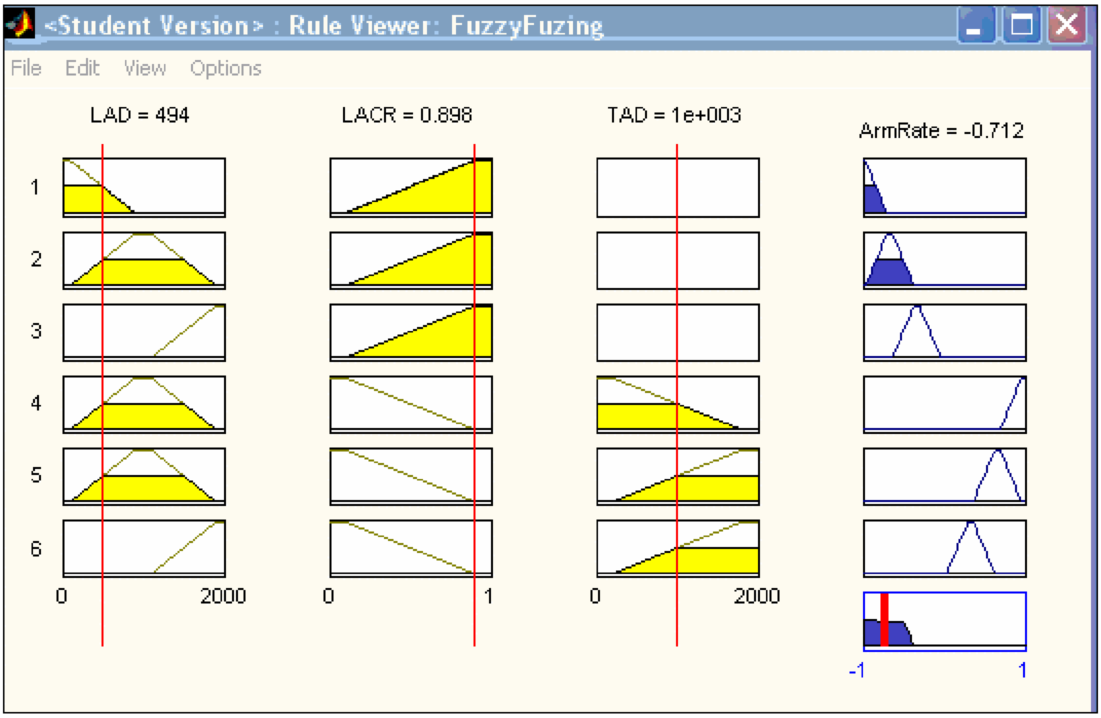
Fig. 10. FFSARC surface plot LACR and TAD.

Notice that the Arm Rate is negative (safing) when the LACR is greater than 0.5, which signifies that the AIM is approaching (separation is decreasing).

The Arm Rate is positive (arming) when the LACR is less than 0.5, which signifies that the AIM is receding (separation is increasing). Note that the target to AIM distance does not play a significant role in this plot since the safety of the launcher takes precedence over the reliability of the mission. This is expected due to the set-up of the FIS discussed in the preceding paragraphs.

FIS Rule Viewer

The FIS Rule Viewer encompasses each of the FIS member elements discussed in the previous paragraphs into one view. Figure 11 represents an LAD of 494 units, an LACR of 0.898 (signifying the AIM is approaching the launcher), and a TAD of



LAD = 494

LACR = 0.898

TAD = 1e+003

ArmRate = -0.712

Fig. 11. FFSARC rule viewer (AIM is approaching launcher).

1000 units. The resulting output is an ArmRate of negative 0.712, which signifies that the S&A is progressing towards the Safe condition, or what the S&A should be doing at this time, safing.

The first three columns of the Rule Viewer are the three input membership functions. In the FIS Rule Viewer, each of the thin redlines can be adjusted to simulate various input variable values. The fourth column represents the output of each of the rows. Because the rules are set up using the AND operator, the minimum area of each row is carried over to the output column. The bottom box of the fourth row represents the aggregate of the minimums of each row. The thick redline at the bottom box in the fourth column represents the final defuzzified output of the FFSARC. This defuzzified solution is the centroid of the aggregate of the output column.

Figure 12 represents a LAD of 1,650 units, an LACR of 0.0904(signifying the AIM is receding from the launcher) and a TAD of 542 units. The resultant output is an ArmRate of 0.573, which signifies that the S&A is progressing towards the Arm condition, or exactly what the S&A should be doing at this time, arming.

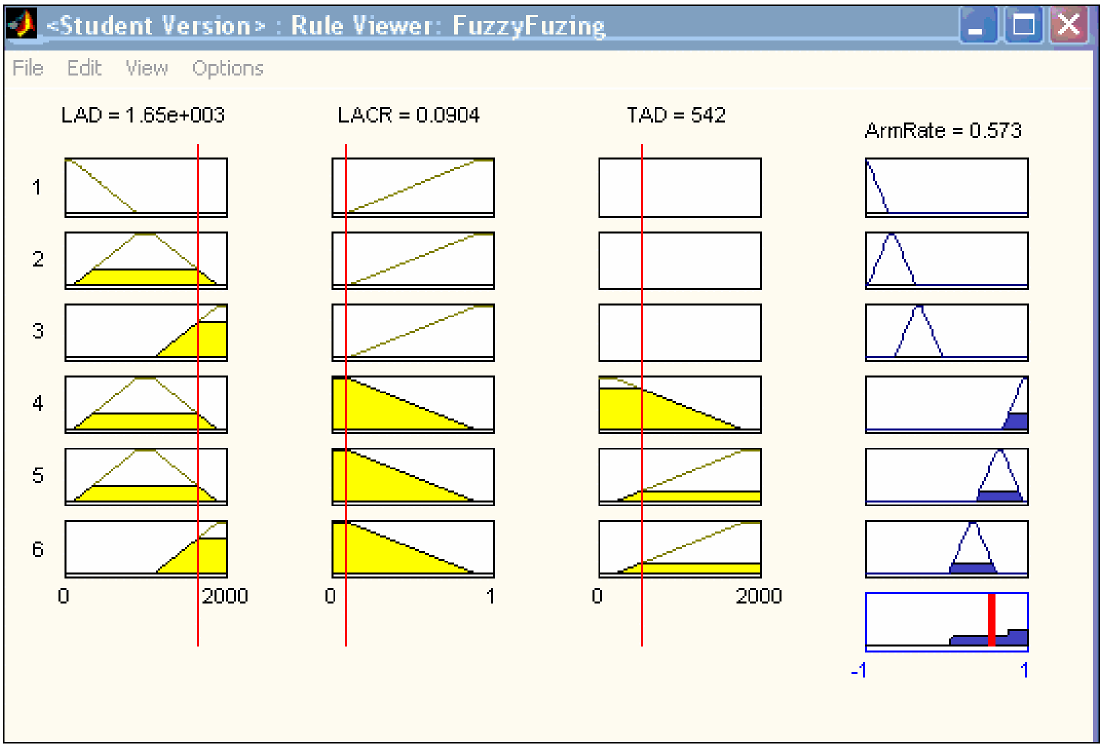


Fig. 12. FFSARC Rule Viewer (AIM is receding from launcher).

Resolution

Chapter III began with the presentation of the problem concerning the safety of launch vehicles when a friendly munition re-enters the launcher's safe separation envelop, while it is pursueing a target. A simulated engagement scenario described a feasible situation where this can occur. This is a problem because current fuze technology cannot return to a safe condition once it has armed. If the munition initiates too close to the launcher, the launcher could be damaged or destroyed.

The resolution to this problem was developed by creating a FFSARC that determined the closure rate and the distance between the launcher and the muntion. By determining these two inputs and the closure rate between the munition and the target, the arming rate could be adjusted so that the S&A rotation could be either towards the safe condition or the armed condition. The development of the FFSARC followed the standard techniques established by Mr. Zadeh and Mr. Mamdani including defining membership function, a fuzzy inference system, rules, and a final defuzzification of the output. The final output of the FFSARC is a positive or negative number between minus 1 and positive 1. A negative number represents rotation of the S&As rotor towards the safe condition and a positive number represents rotation towards the armed position.

Once the FFSARC was functioning as required, its membership functions, rules, and defuzzification method were coded into the Fuzzy Fuzing GUI.

CHAPTER IV

GRAPHIC USER INTERFACE

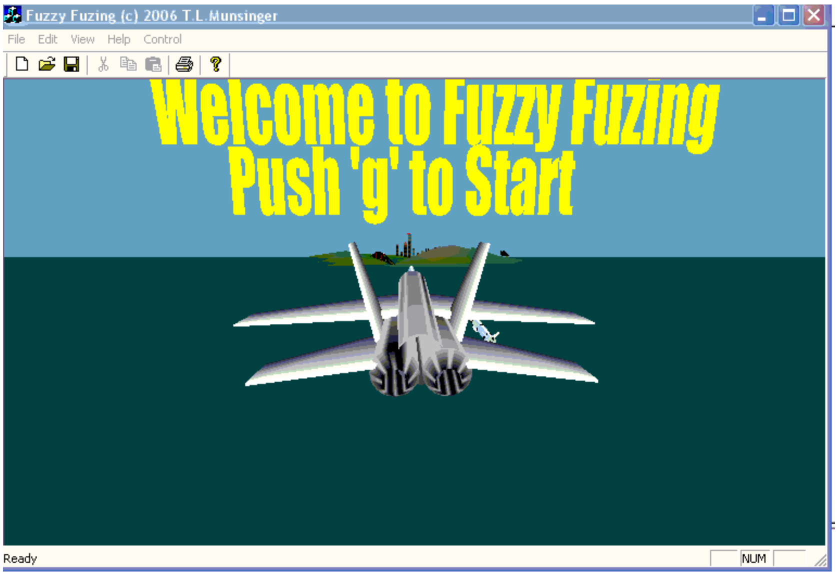
Introduction

The engagement scenario and the FFSARC that were developed in Chapter III are graphically presented in the “Fuzzy Fuzing” Program. After the FIS was verified, the functions of the FFSARC were programmed using Microsoft’s VISUAL C++ [5] and Sun Microsystems’s OPENGL [6] to provide a visual verification of the concept. The GUI presents an animation of the physical responses of the launcher, the AIM, and the target during an engagement including the AIM’s acquisition and interception of the target and a visual representation of the fuze rotor to the FFSARC program outputs.

The opening view instructs the user to press “g” to start. After “g” is pressed the launcher flies toward the island and passes several messages that instruct the user on the different keyboard functions that are available (Figure 13).

Pressing “h” brings up the help view, which instructs the user on the keyboard functions that are available. When you start the demonstration after pressing “g”, the user can use the “I”, “J”, “K”, and “L” keys to maneuver around Fuzzy Fuzing or the user can then press “t” to reset, fire the AIM, and view the overhead map to observe the engagement scenario in action (Figure 14).

Figures 15 through 23 show the Fuzzy Fuzing GUI in action. These figures are screen captures that were made while the GUI was running.

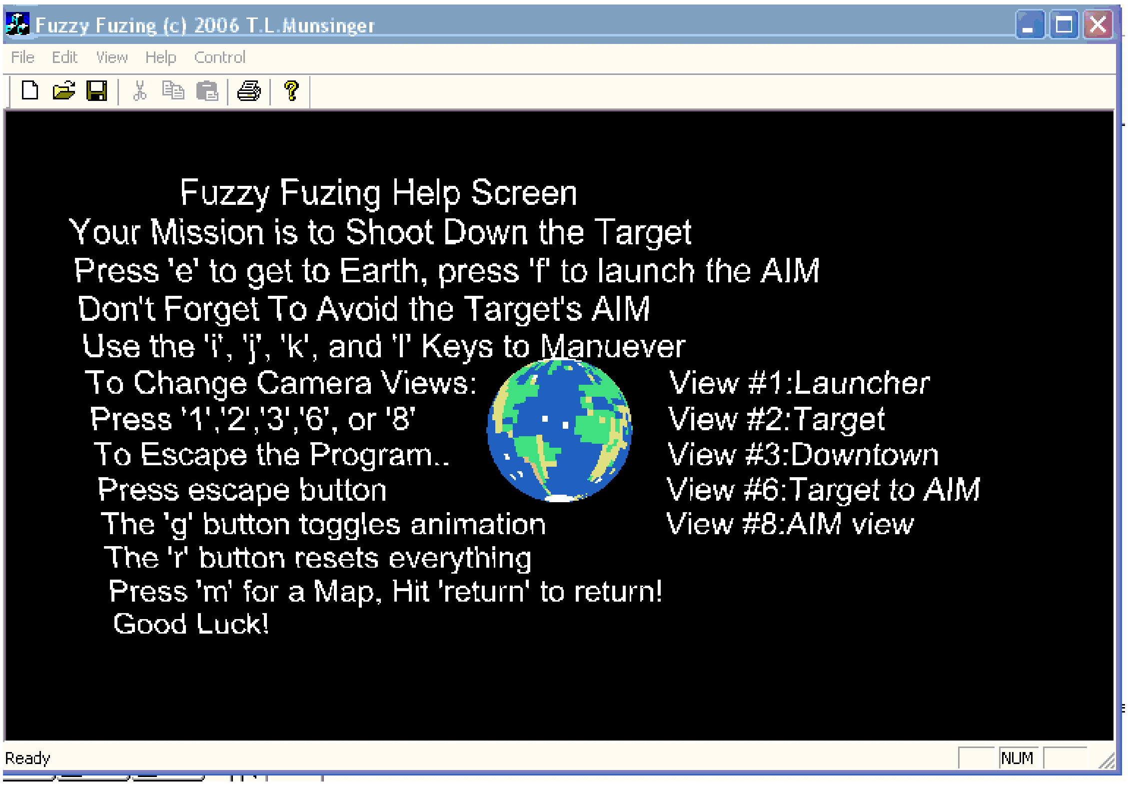


Welcome to Fuzzy Fuzing
Push 'g' to Start

Fig. 13. Fuzzy fuzing opening view.

Pressing “t” will bring up this overhead view of the engagement scenario. The launcher’s position and vector is denoted by the white “L” at the bottom center, the red circle around the launcher signifies its safe separation envelop. The AIM is the red “A” seen receding from the launcher. The target is denoted by the black “T” seen at right center. The launcher, AIM, and target have corresponding colored trails behind them to denote their respective paths or vectors (Figure 15).

The S&A rotor is shown in the upper left hand corner. On the rotor the detonator is represented by the red cross and the explosive lead is shown as a yellow circle. The white triangle denotes the rotor position, either safe, arm or somewhere in-between. In Figure 15, the rotor is progressing towards the arm condition, due to positive LACR and increasing LAD. The armrate equals a positive 0.1314 which equates to a slow arm output from the FFSARC.



Fuzzy Fuzing Help Screen
Your Mission is to Shoot Down the Target
Press 'e' to get to Earth, press 'f' to launch the AIM
Don't Forget To Avoid the Target's AIM
Use the 'i', 'j', 'k', and 'l' Keys to Manuever
To Change Camera Views:
Press '1','2','3','6', or '8'
To Escape the Program..
Press escape button
The 'g' button toggles animation
The 'r' button resets everything
Press 'm' for a Map, Hit 'return' to return!
Good Luck!

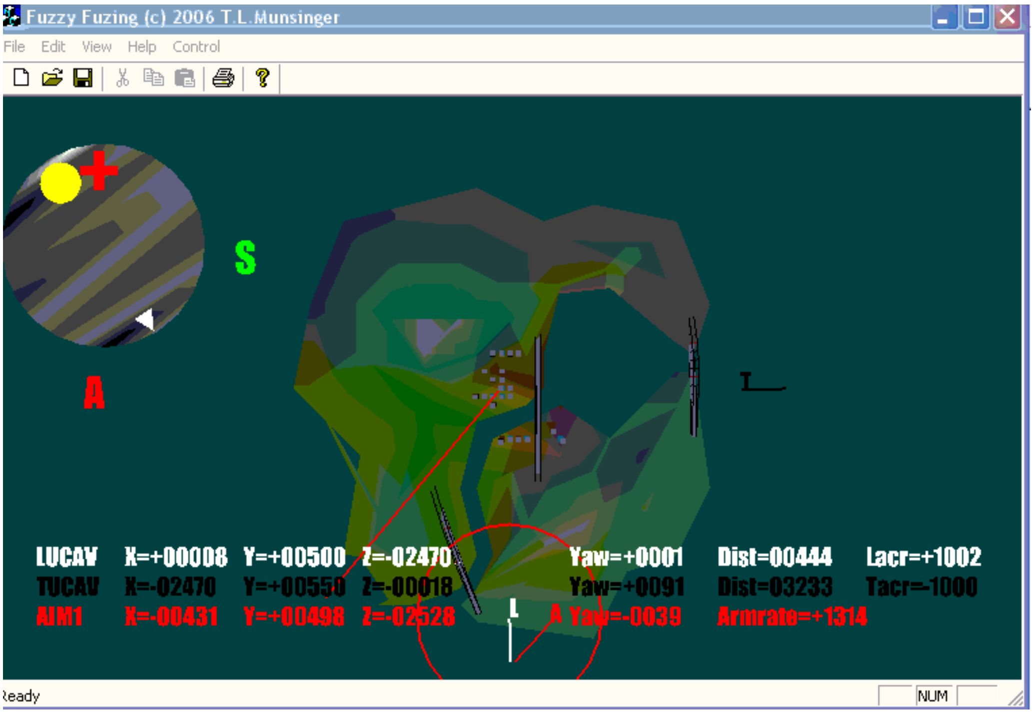
View #1:Launcher
View #2:Target
View #3:Downtown
View #6:Target to AIM
View #8:AIM view

Fig. 14. Help view.

At the bottom of this view X, Y, and Z coordinates, distances, and closure rates pertaining to locations and vectors of the launcher, AIM, and target are displayed.

Figure 16 shows the scenario a few seconds later. The AIM is pursuing the target and starting to re-enter the launcher’s safe separation envelop. Because the LACR is negative the armrate has turned negative and the S&A rotor is starting to progress towards the safe condition.

Now both the AIM and the target have re-entered the launcher’s safe separation envelop. Because of the negative value of LACR the armrate is negative causing the rotor to almost reach the safe condition (Figure 17).



```
LUCAV   X=+00008  Y=+00500  Z=-02470    Yaw=+0001   Dist=00444    Lacr=+1002
TUCAV   X=-02470  Y=+00550  Z=-00018    Yaw=+0091   Dist=03233    Tacr=-1000
AIM1    X=-00431  Y=+00498  Z=-02528    Yaw=-0039   Armrate=+1314
```

Fig. 15. Engagement scenario view 1.

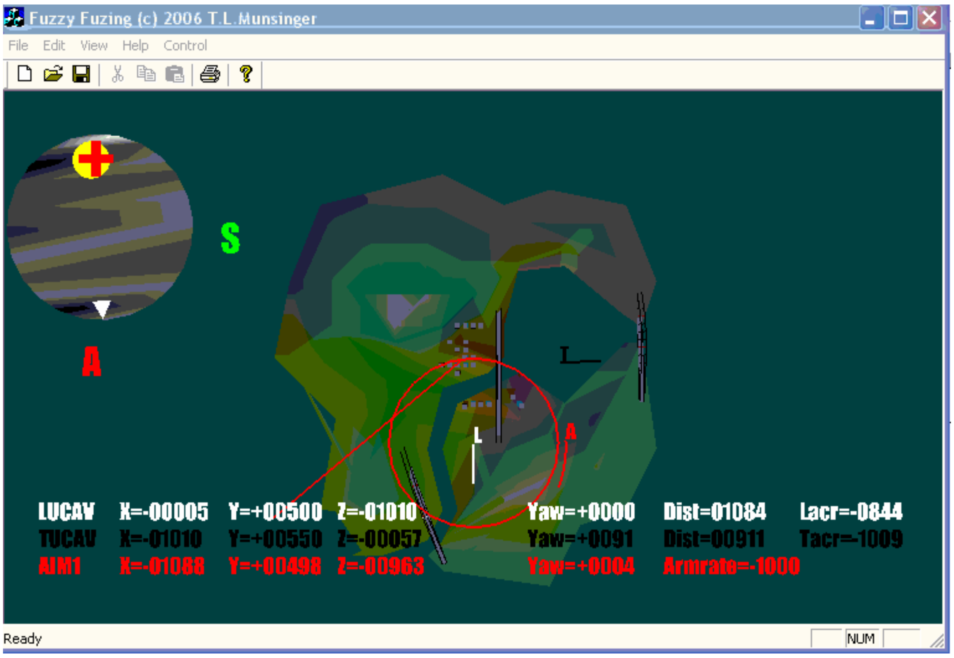
Now that the target and the AIM have achieved safe separation from the launcher, the S&A rotor quickly progresses to the armed condition (Figure 18).

Pressing “8” switches the camera view from the overhead to the armed AIM’s view of the target. There is not much time for the target (Figure 19).

The AIM has drawn within 100 units of the target causing the TDD to fire the detonator (Figure 20). Asta la vista, baby!

Figures 21 through 24 show other camera views that can be used during viewing the engagement scenario.

Pressing “f’ launches the AIM and automatically changes the camera, the AIM to target view, allowing the user to view the AIM as it pursues and intercepts the target (Figure 21).



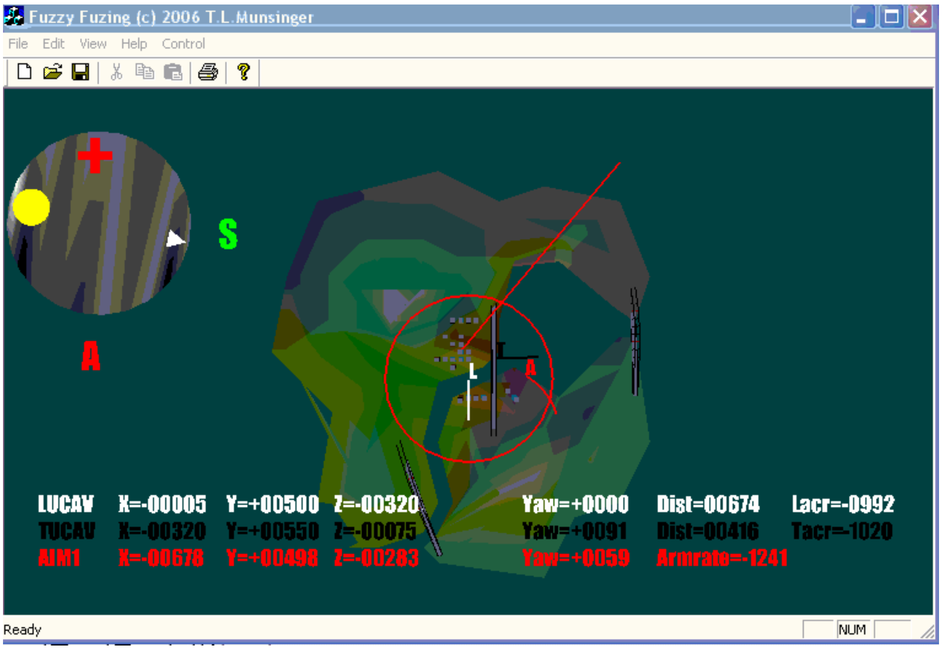
```
LUCAV  X=-00005  Y=+00500  Z=-01010  Yaw=+0000  Dist=01084     Lacr=-0844
TUCAV  X=-01010  Y=+00550  Z=-00057  Yaw=+0091  Dist=00911     Tacr=-1009
AIM1   X=-01088  Y=+00498  Z=-00963  Yaw=+0004  Armrate=-1000
```

Fig. 16. Engagement scenario view 2.

After launching the AIM, pressing “6’ automatically changes the camera view, which allows the user to view the AIM from the target’s point of view as it is pursued and intercepted (Figure 22).

Pressing “3’ automatically changes the camera view, to the top of a building to view the launcher with its AIM. You can also press “a” which puts the launcher and the target in an “autopilot” mode. The two UCAVs fly in a figure “8” pattern and can offer some very realistic aerial ballets. Figure 23 shows the launcher with the white smoke trail and the target with the black trail. The AIM has a red trail. Trails were added to help distinguish between the AIM, launcher, and the target.

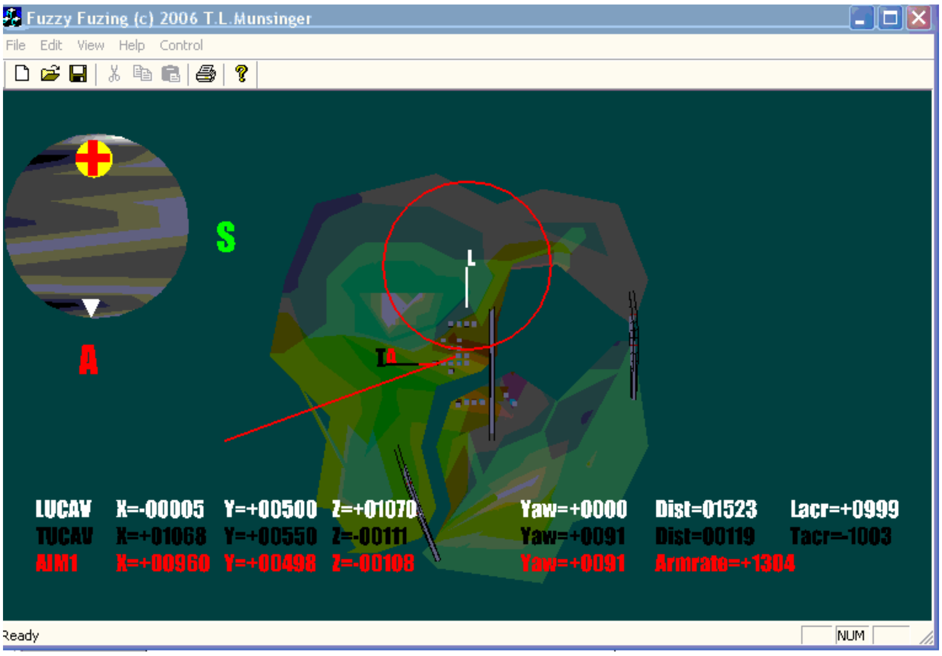
Pressing “3’ automatically changes the camera view, from the top of a building to the launcher. The launcher provides us with this close up view while it was in the autopilot mode (Figure 24).



```
LUCAV  X=-00005  Y=+00500  Z=-00320   Yaw=+0000  Dist=00674     Lacr=-0992
TUCAV  X=-00320  Y=+00550  Z=-00075   Yaw=+0091  Dist=00416     Tacr=-1020
AIM1   X=-00678  Y=+00498  Z=-00283   Yaw=+0059  Armrate=-1241
```

Fig. 17. Engagement scenario view 3.

Chapter IV has demonstrated that using OpenGL to graphically present the problem stated in Chapter II allows the researcher a very useful tool for proving a hypothesis. The Fuzzy Fuzing program allows the engagement scenario to be set up in any way deemed necessary to demonstrate various situations.



```
LUCAV  X=-00005  Y=+00500  Z=+01070   Yaw=+0000  Dist=01523      Lacr=+0999
TUCAV  X=+01068  Y=+00550  Z=-00111   Yaw=+0091  Dist=00119      Tacr=-1003
AIM1   X=+00960  Y=+00498  Z=-00108   Yaw=+0091  Armrate=+1304
```

Fig. 18. Engagement scenario view 4.

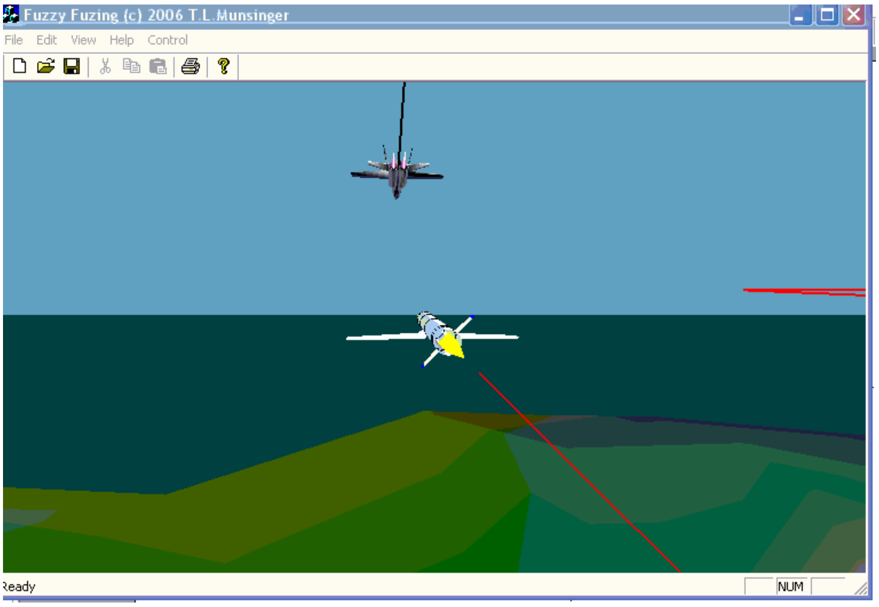


Fig. 19. Engagement scenario view 5.

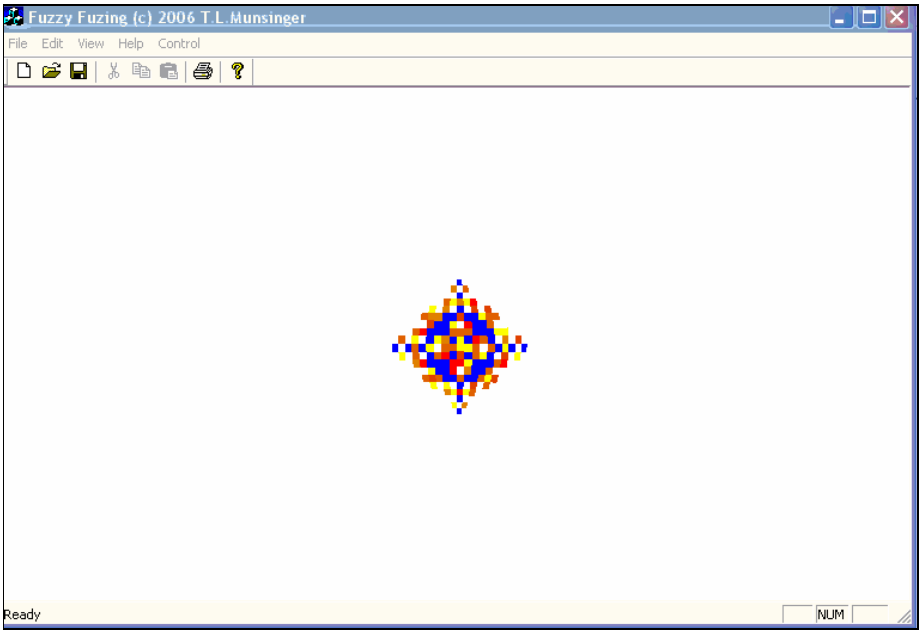


Fig. 20. Engagement scenario view 6.

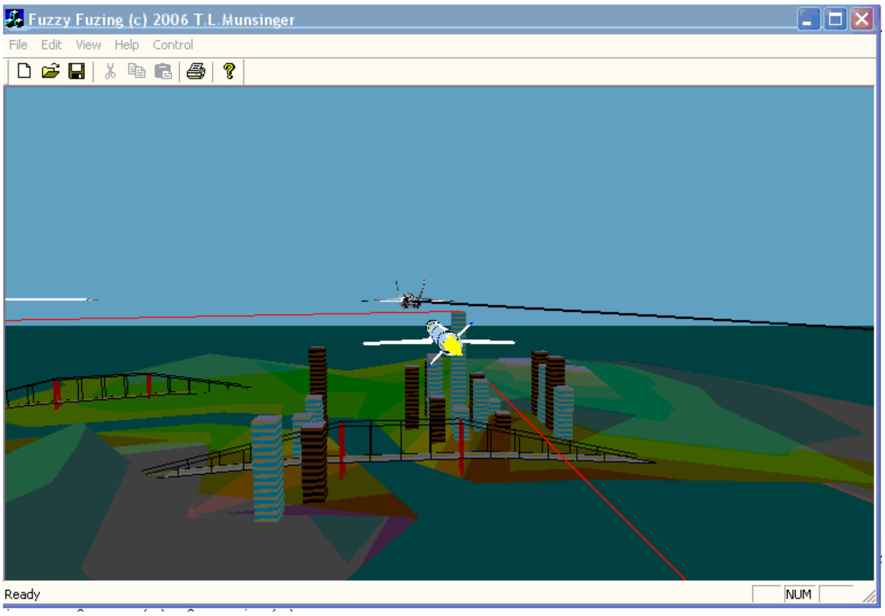


Fig. 21. AIM view.

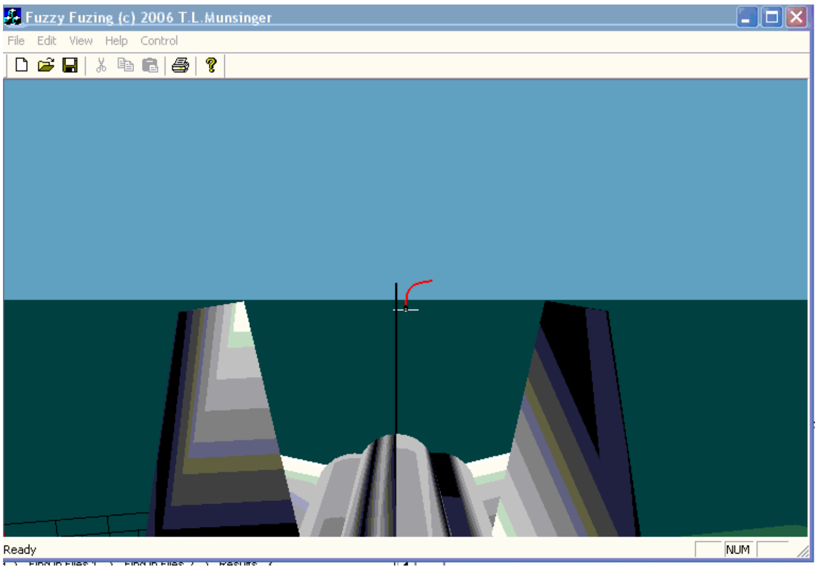


Fig. 22. Target to AIM view.

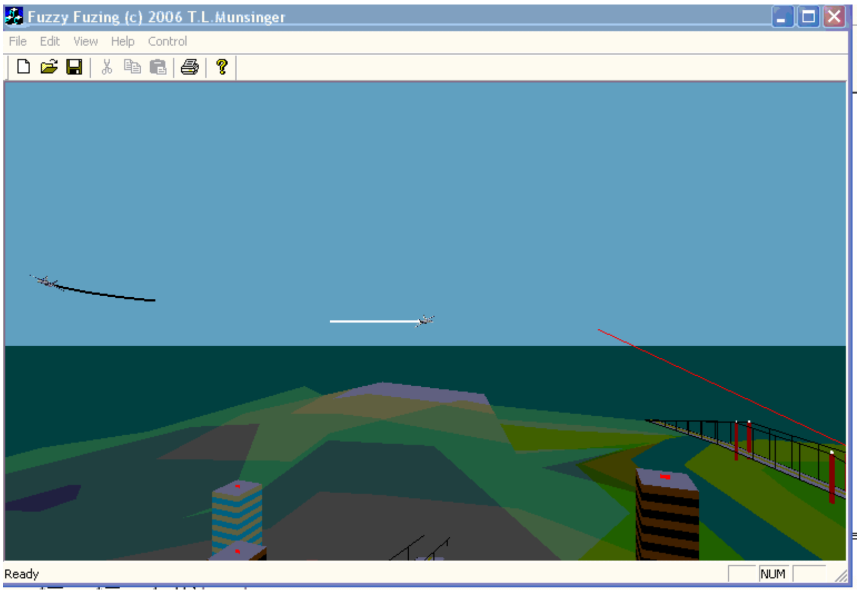


Fig. 23. City view.

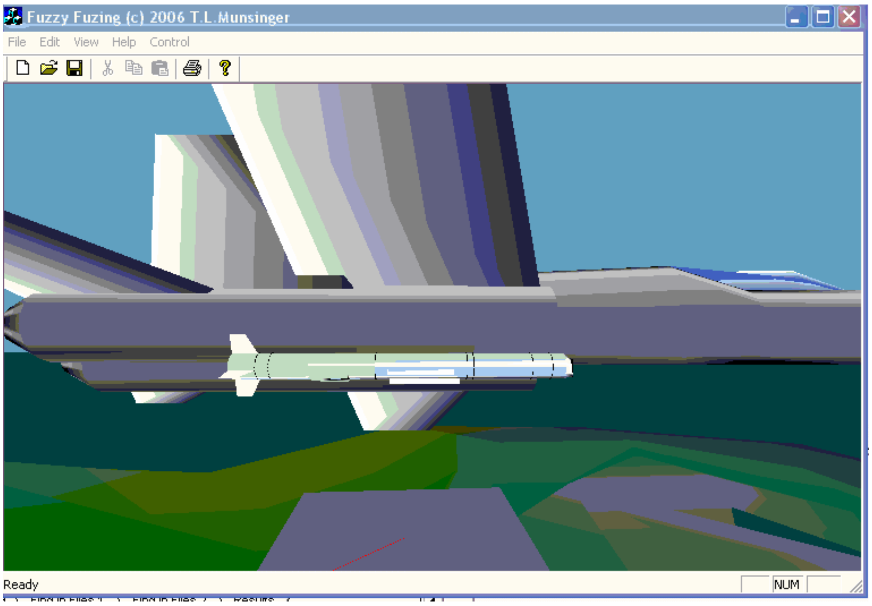


Fig. 24. City view of close approach.

CHAPTER V

SUMMARY AND CONCLUSIONS

Introduction

This paper studies problems created when the introduction of a new technology surpasses the capability of an associated technology. In this case, the new technology is the development of highly maneuverable UCAVs, which have surpassed the ability of current S&A technology to provide launcher safety during air-to-air combat scenarios.

The problem is illustrated by an engagement scenario between three players, the launcher, the AIM, and the target. The scenario includes an S&A device progressing to the armed position while pursuing its target, but without current technology allowing for the re-safing of the S&A as it re-enters the launcher's safe separation envelop.

The solution to this problem is to incorporate fuzzy logic into the S&A controller logic. This, along with the development of a method to determine the closure rate between the launcher and the AIM facilitates the development of an S&A technology, that allows an S&A to return to the safe condition when it re-enters the launcher's safe separation envelop. The following is describes how this solution is determined to be viable.

Presentation of the Findings

The first part of the solution is to define the problem. The concepts of S&A function, guidance theory, and fuzzy logic are applied to possible engagement scenarios between a launcher, an AIM, and a target. A particular engagement scenario describes a launcher and a target, which are approaching each other on perpendicular vectors. The launcher launches an AIM at the target and the AIM S&A progresses to the arm condition as the AIM recedes from the launcher. Due to mutually approaching perpendicular vectors, the target enters the launcher’s safe separation envelop as does the AIM with its armed S&A.

The problem is that an armed AIM is now approaching its own launcher. Detonation of the AIM at this point would severely damage the launcher, which is not the intent of the launch. The intent is to severely damage and/or destroy the target not the launcher.

To alleviate this problem a method to control the “safeness” and “armness” of the S&A is developed. Instead of using a binary logic device, which would instantaneously switch the S&A between a safe (0) and armed (1) condition, fuzzy logic is used to provide a smoother transition between the two opposite S&A states.

Fuzzy logic also offers the capability to smoothly change states based on inputs without requiring instantaneous switching.

The fuzzy logic solution is developed in a prescribed method described in Chapter III. The first step is to determine what inputs and output(s) are required. The required inputs for the solution are: 1) Distance between the Launcher and the AIM, 2)

Closure rate between the Launcher, the AIM, and 3) Distance between the Target and the AIM. The required output is the arming rate of the S&A rotor.

As a fuzzy inference system is being developed, it is coded into the Visual C++/OpenGL graphical demonstration model titled “Fuzzy Fuzing.” Samples of the C++ code and their function are provided throughout this Chapter to describe how information from the FFSARC is transformed into a graphical representation of the problem and the solution.

Each input is assigned three membership functions, represented by a graph of trapizoidal areas. Trigonometry is used to determine the membership function value between zero (0) and one (1). For example the C++ code used to express the LAD “close” fuzzy set is as follows:

```
/*
LAD: close=   |if LAD < 400, then = 1
              |if LAD > 900, then = 0
              |if LAD >= 400 && LAD <= 900, then = tan(.00199) x (900-LAD)
         */
          if(lad<400){
                  lclose=1.0;
                 }
          if(lad>900){
                  lclose=0.0;
                 }
          if((lad>=400)&&(lad<=900)){
                  lclose=tan(0.00199)*(900-lad);
                 }
```

Coding techniques similar to this are used for all of the input membership functions. The next step is to produce code that represents the rules from Chapter III:

```
//1) If LAD is Close, and LACR is Approach, then ArmRate is FastSafe.

         rule1=min(lclose, lapproach);
```

```
        //2) If LAD is Safesep, and LACR is Approach, then ArmRate is Safe.

        rule2=min(safesep, lapproach);

        //3) If LAD is Far, and LACR is Approach, then ArmRate is SlowSafe.

        rule3=min(lfar, lapproach);

        //4) If LAD is Close, and LACR is Recede and the TAD is Far, then ArmRate is SlowArm.

        //rule4=min(min(lclose, lrecede), tfar);

        rule4=min(lclose, lrecede);

        //5) If LAD is Safesep, and LACR is Recede, and the TAD is Near, then ArmRate is Arm.

        //rule5=min(min(safesep, lrecede), tnear);

        rule5=min(safesep, lrecede);

        //6) If LAD is Far, and LACR is Recede, and the TAD is Close, then ArmRate is FastArm.

        //rule6=min(min(lfar, lrecede), tclose);

        rule6=min(lfar, lrecede);
```

After minimum values of the rules are known, the armrate can be easily determined. The C++ code for computing armrate is as follows:

```
        negsafe=max(max(rule1,rule2),rule3)*-1;

        posarm=max(max(rule4,rule5),rule6);

        armrate=negsafe+posarm;
```

Once the arm rate is calculated, the S&A rotor is drawn using OpenGL with the following C++ code fragment:

```
void  CTLMProjView::rotor(void) // map vertices to faces
{
```

```
int armed=0;

glEnable(GL_COLOR_MATERIAL);
glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);

        FIS();

        if((fire1==1)&&(lad>25)){

        if(armrate > 0){
                if((ryaw <=3)&&(ryaw>=-90)){
                        ryaw = ryaw-(armrate);
                }
        }

        if(armrate <= 0){
                if((ryaw <=0)&&(ryaw>=-92)){
                        ryaw = ryaw-(armrate);
                }
        }

        }//end of if fire1=1

        glPushMatrix();
        glEnable(GL_LIGHT3);
                rpolygon(19,18,17,16);//draws rotor
                rpolygon(20,19,16,15);
                rpolygon(21,20,15,14);
                rpolygon(22,21,14,13);
                rpolygon(23,22,13,12);
                rpolygon(0,23,12,11);
                rpolygon(1,0,11,10);
                rpolygon(2,1,10,9);
                rpolygon(3,2,9,8);
                rpolygon(4,3,8,7);
                rpolygon(5,4,7,6); //draws rotor

                rpolygon(44,43,42,41);//draws det
                rpolygon(45,44,41,40);
                rpolygon(46,45,40,39);
                rpolygon(47,46,39,38);
                rpolygon(48,47,38,37);
                rpolygon(25,48,37,36);

                rpolygon(26,25,36,35);
```

```
                        rpolygon(27,26,35,34);
                        rpolygon(28,27,34,33);
                        rpolygon(29,28,33,32);
                        rpolygon(30,29,32,31); //draws det

                        rpolygon(50,51,52,50);//white triangle


        glDisable(GL_LIGHT3);

   glPopMatrix();

}
```

The “rotor()” function uses the armrate output from the function “FIS()” method to determine the yaw angle of the rotor. The rotor is then graphically represented by the OpenGL/Visual C++ program, titled, “Fuzzy Fuzing.” “Fuzzy Fuzing” visually verifies that an FLC is incorporated into a fuze arming controller and can provide safety and reliability for future highly maneuverable munitions.

“Fuzzy Fuzing” also contains the C++ code needed to “draw” the launcher, the AIM and the Target in much the same manner as the previously mentioned rotor, albeit they are much more complex. All of these dynamic objects are placed in a simulated landscape, which consists of an ocean, sky, an island, and a cityscape. The static landscape provides a backdrop for the animated objects to move against. Movement of these objects is achieved once again through simple trigonometric equations such as:

```
// move LUCAV
        lux = lux + (sin(rvar*lyaw)*speed);

        if (cos(rvar*lyaw) <0)
                luy = luy + (sin(rvar*lpitch)*speed);
        else
                luy = luy - (sin(rvar*lpitch)*speed);

        luz = luz + (cos(rvar*lyaw)*speed);
```

The pursuit path of the AIM is generated with the following C++ code:

```
opp = tux-aim1x;
adj = tuz-aim1z;
hyp = sqrt((adj)*(adj)+(opp)*(opp));
aim1x = aim1x + (sin(rvar*aim1yaw)*speed*1.2f);
aim1y = aim1y;
aim1z = aim1z + (cos(rvar*aim1yaw)*speed*1.2f);

if((tux>aim1x)&&(tuz>aim1z)){ //case 1

        aim1yaw=asinf(opp/hyp)/rvar;

        }
if((tux>aim1x)&&(tuz<aim1z)){ //case 2

        //aim1yaw=110.2;
        aim1yaw=((acosf(opp/hyp)+PI/2)/rvar);
        }
if((tux<aim1x)&&(tuz<aim1z)){ //case3

        aim1yaw=(-(asinf(opp/hyp)+PI)/rvar);

        }
if((tux<aim1x)&&(tuz>aim1z)){ //case4

        aim1yaw=(asinf(opp/hyp)/rvar);

        }
```

Basically, the yaw angle between the AIM and the Target is calculated and this value becomes the yaw angle of the AIM the next time it is drawn. Also the AIM’s speed is 120% of the Target’s speed, so an interception is inevitable. The values of the membership functions are also calculated with non-complex equations such as a distance formula. The LAD variable is calculated by the following code segment:

```
ladist[d].lad=sqrt((lux-aim1x)*(lux-aim1x)+(luy-aim1y)*(luy-aim1y)+(luz-aim1z)*(luz-
aim1z));
```

Tha LACR variable is determined by the following:

`lacr = ((ladist[d-1].lad - ladist[d-2].lad)/(ladist[d].lad - ladist[d-1].lad));`

The code represents a small percentage of that needed to produce the Fuzzy Fuzing GUI, but it does provide a feel for the overall simplicity inherent when using fuzzy logic as a system controller. The GUI affords a very realistic demonstration of the problem and solution and can be manually controlled to simulate any number of different engagement scenarios.

Conclusion

The safety and reliability of a “fuzzy” S&A is made clear by the “Fuzzy Fuzing” GUI. Only three inputs, LAD, LACR, and TAD, are required to enhance the safety of the launcher and the reliability of the AIM. The S&A device can only progress to an armed condition when the LACR is positive and the LAD is greater than the required safe separation distance.

Research Questions

A primary question raised by this project is, can a method be developed to provide LAD, LACR, and TAD positional information to an AIM’s fuze central processor unit while the AIM is in pursuit of a target, without compromising the AIM’s mission?

A secondary question is, does implementation of the fuzzy fuzing concept meet the requirements of MIL-STD-1316E?

Conclusions Relevant to Research Questions

With respect to the first question, currently no known wireless data link exists between a launcher and a munition. Wireless data links between munitions and targets definitely do exist and are used as the basic environment for TDDs to verify target proximity. Basically a munition needs a system similar to the one used by a TDD, except that its primary function is to detect the position and velocity of the launcher.

Regarding the second question, not only does the fuzzy fuzing concept meet the requirements of MIL-STD-1316, but it surpasses them by allowing the fuze to return to a safe condition if it re-enters the safe separation envelop of the launcher.

Limitations of Study Design and Procedures

This project is limited to the development of the FFSARC FIS and a Fuzzy Fuzing program used to visually demonstrate the concept of fuzzy fuzing. As was mentioned above, this preliminary concept development project is sufficient to prove that fuzzy fuzing can increase the safety of a launcher, and at the same time maintain reliability of munitions.

Recommendations for Future Research

Future research in this area could include higher fidelity flight simulations, such as both a FFSARC and a modified TDD simulator that provides launcher velocity and positional data to the AIM. The flight simulator must be able to simulate higher “g” levels than is now possible by manned aircraft due to the unmanned condition of future UCAV engagements.

The flight simulator must be able to run or many different engagement scenarios and the fuzzy fuze’s performance should be monitored and analyzed during the development process.

It is recommended that further research into this area develop S&A and fuzing technology in general, so that it can maintain functional capability, provide enhanced safety and provide reliability in an ever changing air-to-air combat environment.

REFERENCES

REFERENCES

[1] The Chief of the Bureau of Naval Weapons, *Weapons Systems Fundamentals-Analysis of Weapons, NAVWEPS OP 3000 Volume 2*. 1960.

[2] Department of Defense, “Design Criteria Standard: Fuze Design, Safety Criteria For. MIL-STD-1316E. Distribution Statement A,” 1998. Retrieved October 23, 2006 from the World Wide Web: www.klabs.org/DEI/References/MilSpecs/milspecs/mil_std_1316e_fuze_design.pdf

[3] J. Yen and R. Langari., *Fuzzy Logic, Intelligence, Control, and Information.* Upper Saddle River, NJ: Prentice-Hall Inc., 1999.

[4] The MathWorks, *Fuzzy Logic Toolbox*. Natick, MA: The MathWorks, Inc., 2001.

[5] Microsoft, *Visual C++, Version 6 Software*. Redmond, WA: Microsoft, Inc.

[6] Sun Microsystems, *Sun OpenGL Application Programming Interface*. Santa Clara, CA: Sun Microsystems, Inc.

APPENDIX A

DEFINITIONS

Air Intercept Missile – a munition that is launched from an aircraft and is intended to intercept and destroy another aircraft.

Firing Train Interrupter – the mechanism that maintains an out-of-line condition of a fuze explosive train until the fuze arms.

Fuze - an electro-mechanical ordnance component, which maintains safety and reliability of a munition system. A fuze incorporates a target detection device and a safe and arm device.

Fuzzy Inference System - the process of mapping inputs into an output using

Gravitational Units (“g”s) – acceleration due to gravity at the Earth’s surface (32.2 ft/sec^2).

Post-Safe-Separation-Safety - control of unintentional functioning after a proper arming delay has transpired.

Safe and Arm Device - a device that prevents inadvertent arming of the fuze until a pre-programmed or designed sequence of environmental events have been sensed and which removes the safety features and progresses to the armed condition.

Safe Separation Distance – minimum distance between the launcher and a munition beyond which the launcher cannot be damaged by an initiation of the munition.

Target Detection Device – a device that detects when a munition is in close proximity to a target.

Unmanned Air Vehicle - a remotely controlled aircraft, usually piloted by a ground pilot.

Unmanned Combat Air Vehicle – an unmanned air vehicle that can carry and launch munitions.