INTELLIGENT EXECUTIVE GUIDANCE AGENT FOR GENERAL AVIATION AIRCRAFT UNDER FREE FLIGHT

A Thesis

by

JIE RONG

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

MASTER OF SCIENCE

August 2002

Major Subject: Aerospace Engineering

INTELLIGENT EXECUTIVE GUIDANCE AGENT FOR GENERAL AVIATION AIRCRAFT UNDER FREE FLIGHT

A Thesis

by

JIE RONG

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

MASTER OF SCIENCE

Approved as to style and content by:

John Valasek
John Valasek
(Chair of Committee)

Donald T Ward
Donald T. Ward
(Member)

John H. Painter
John H. Painter
(Member)

Remesh Talreja
(Head of Department)

August 2002

Major Subject: Aerospace Engineering

ABSTRACT

Intelligent Executive Guidance Agent for General Aviation Aircraft under Free Flight.
(August 2002)
Jie Rong, B.S., Peking University
Chair of Advisory Committee: Dr. John Valasek

Conflict detection and resolution is a critical capability for the realization of free flight, a new concept of air traffic management that allows pilots to select their own flight paths and airspeeds in real time. A particularly demanding situation within this environment occurs when multiple traffic and weather conflicts arise simultaneously. A solution that forms the basis for this thesis is an agent based hierarchical system that attempts to provide optimal and conflict free flight path guidance in these multiple conflict situations. An intelligent executive guidance agent, acting as a high-level arbitrator, receives guidance information from a previously designed lower-level weather agent and a traffic detection and collision avoidance agent. When the flight path guidance from the two-lower level agents conflicts, the executive agent arbitrates by considering the spatial and temporal characteristics of the conflicting guidance. It classifies them as either tactical or strategic in nature, and then prioritizes them according to a pre-defined rule base of conflict priorities. The arbitration function thus acts as a fuzzy controller, and gradually switches the guidance between the weather agent and traffic agent, providing conflict free flight path guidance, as the aircraft flies in and out of dangerous regions.

Results of test cases presented in the thesis demonstrate that the approach and algorithm can successfully resolve combined weather and traffic conflicts in real-time, subject to realistic imposed constraints. The severity of conflicting flight paths was managed within acceptable levels, and the ultimate recommended conflict free flight path is generally between those originally proposed by the weather and traffic agents. The algorithm does not exhibit any critical failures during cases tested, and proved robust and reliable. The proposed agent based hierarchical system, when integrated with a simplified flight management system coupled with a heading command and hold autopilot, offers an effective and reliable guidance and navigation system for generating safe, alternate flight paths in conflict situations.

ACKNOWLEDGMENTS

Thanks to Dr. John Valasek, Dr. John H. Painter and Dr. Donald T. Ward for their support and guidance in this effort, and for kindling my love of airplanes and air traffic control. In addition, thanks to Miss Sangeeta Bokadia and Mr. Surya Shandy, both of whom have worked with me on the same project for nearly two years, for their assistance and support. Designing and developing a CD&R algorithm as an agent based hierarchical system is a primary part of a research project of the Texas A&M University Flight Simulation Laboratory. The project is titled as "Cockpit Data Fusion with Fixed-Base Simulation Validation for Free-Flight Guidance" and funded by the State of Texas Advanced Technology Program, under grant number 000512-0301-1999. The author gratefully acknowledges this support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The current air traffic control (ATC) system is managed through air traffic radar surveillance, ground track, ground computation, and verbal communication between ground controller and pilot. Aircraft follow the flight paths requested from and assigned by ATC, and managed by a ground controller, who plays key role in maintaining the flight safety and providing potential conflict resolution. Though it remains a question if the annual air traffic rate will grow by three to five percent as expected for at least the next 15 years after September 11th, 2001, the current airspace architecture and management will still not be able to efficiently handle the rapid increase in air traffic volume. New advances in air-traffic management systems (ATMS) are necessary due to this increasing volume and complexity of air traffic, as well as the increasing demand for efficient air traffic control leading to decreased ground controllers' workload and reduced time and financial cost.

Free Flight, also called user preferred traffic trajectories, is an innovative concept designed to enhance the safety and efficiency of the National Airspace System (NAS). It is regarded by many from the aviation community as the future of the air traffic management.

Free flight may be simply defined as a safe and efficient flight operating capability under instrument flight rules (IFR), in which the pilot has the freedom to select his flight path and airspeed in real time.[1] It moves the NAS from a centralized command-and-control system between pilots and air traffic controllers, to a distributed system that allows pilots,

---

This thesis follows the style and format of *Journal of Guidance, Control and Dynamics*.

whenever practical, to choose their own route and file a flight plan that follows the most efficient and economical route.[2] Implementation of Free Flight, which offers benefits in system safety, capacity, and efficiency, is critical to advancing aviation by accommodating the nation's growing airspace needs.[2]

Besides retaining more autonomy in determining the flight trajectory, responsibility for aircraft separation safety rests increasingly with the pilot. Therefore, in the postulated free flight regime, ground controllers act only as supervisors and intervene only in exceptional cases. However, Free Flight also calls for limiting pilot flexibility in certain situations, so to ensure separation at high-traffic airports and in congested airspace, to prevent unauthorized entry into special use airspace, and for various safety related reasons.[2]

Free Flight will be used differently according to the situation. For example, in clear and uncrowded skies, pilots are able to take advantage of the full extent of Free Flight. On the other hand, the total flexibility of Free Flight may be restricted in air space with bad weather or high traffic density.[2]

1.2 Conflict Detection and Resolution in Free Flight

Practical realization of free flight relies greatly on increased digital data flow between aircraft and ground controllers, and with other aircraft in the immediate airspace. Ideally, all aircraft occupying a given airspace would share current information on their intent, the overall air traffic situation, weather conditions, and terrain variations. The current information would be sent to a Conflict Detection and Resolution (CD&R) algorithm, executing in an on-board aircraft computer. The CD&R algorithm will first determine whether potential conflicts exist. Not only do conflicts here refer to other aircraft that may cause fatal collisions, but also to severe weather conditions and mountain peaks.

All of the potential conflicts may be classified into three primary categories: weather, traffic and terrain. If certain conflicts are detected, the CD&R algorithm is responsible for computing and providing optimal and conflict-free flight path guidance after processing all of the relevant information. Upon confirmation by the pilot, the aircraft would then fly along the prescribed conflict-free flight path, via guidance provided by a Flight Management System (FMS), with a coupled autopilot (A/P). The pilot acts as a supervisor and takes over and controls the aircraft manually only when necessary. Intervention from the ground controller is rare, and only occurs for potential conflicts that the pilot might have overlooked. In other situations, a pilot may query the ground controller to advise on a particular optimal flight path provided by the onboard computer.

1.3 Research Objectives

In general, the overall research objective for this project is designing and implementing an aircraft conflict detection and resolution algorithm for General Aviation (GA) under Free Flight conditions. However, the algorithm is not restricted to GA aircraft, which is extensible for jets and commercial airlines without radical modification. Currently, the algorithm is only demonstrated for GA aircraft because we are lack of commercial transport model in the Engineering Flight Simulator, the primary evaluation instrument for the algorithm.

Three specific tasks are proposed to be completed in this project, which are listed as following:

- Resolve trajectory guidance conflicts by implementing suitable guidance software architecture and requisite algorithms, with validation by real-time, fixed-base flight simulation, under Free Flight conditions, which is the key technology item for enabling individual aircraft to compute and fly trajectories while simultaneously maintaining separation from other data-linked aircraft, from weather, and from terrain. This task is the primary research work for the author, and the accomplishment of this task requires the fulfillment of two other tasks described as following, which are carried out respectively by two colleagues of the author.

- Validate the conflict resolution guidance software using real-time, fixed-base flight simulation, specifically with regard to handling traffic restrictions for scenarios with multiple aircraft, which entails generating multiple traffic trajectories that are digitally communicated to the aircraft, according to the Automatic Dependent Surveillance Broadcast (ADS-B) format and scenarios. Mr. Surya Shandy is in charge of this research task.[3]

- Validate the weather restrictions guidance software using real-time, fixed-base flight simulation, for conditions of squall line weather. Simulated radar intensity data for a moving line of thunderstorms will be generated, and this intensity data will be integrated into the existing moving map display, and into the existing simulator weather graphic, as seen from the cockpit. Miss Sangeeta Bokadia is in charge of this research task.[4]

This thesis describes in detail the design and implementation of an *agent based hierarchical system.* It proposes an algorithm for conflict detection and resolution, and provides optimal and conflict free flight path guidance in situations where more than one type of conflict exists.

1.4 Agent Based Hierarchy

The proposed algorithm for aircraft conflict detection and resolution is a *hierarchical agent based system*, composed of several independent intelligent agents. Typical independent agents would be a *weather agent*, *traffic agent* etc. The overall structure of the agent based hierarchical system is shown in Figure 1.1 for a two-agent system with a high level agent--the executive agent.

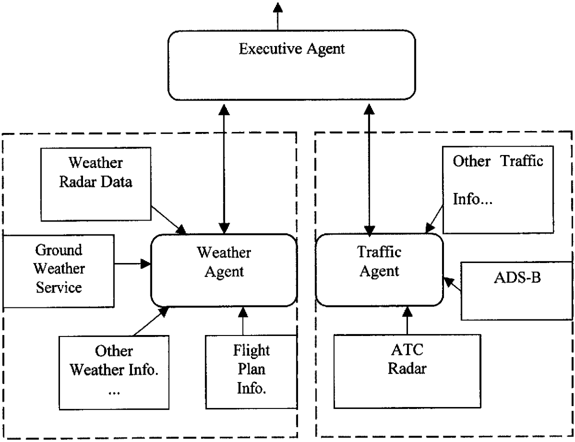


Figure 1.1 Overall Architecture of Agent Based Hierarchical System

A weather agent detects severe weather conditions based on data received from onboard weather radars, meteorological satellites, and data-linked ground weather information services. Considering the weather restrictions, the weather agent computes an optimal flight path that ensures the safety of the aircraft. The traffic agent detects air traffic in the neighboring airspace and keeps the aircraft out of the protected zones of other aircraft. Current air traffic scenarios are communicated digitally between individual aircraft, as well as between pilots and ground controllers using the Automatic Dependent Surveillance Broadcast (ADS-B) message. Negotiation between pilots of aircraft in adjacent airspace is also necessary in acquiring the optimal flight trajectories. Besides the two agents described above, there should be a terrain agent, which is not shown in Figure 1.1, uses navigation data and a geological database to avoid Controlled Flight into Terrain (CFIT). Implementation of the terrain agent will be one of the future research tasks.

Most of the current relevant researches of CD&R algorithms focus on either traffic or weather conflicts separately. Very little work has been done to date in dealing with the situation where more than one type of conflict occurs, individually and simultaneously. In the agent-based system, a high-level agent called the *executive agent* is responsible for solving the multi-conflict detection and resolution problem. The weather, traffic and terrain agents are assumed to be independent of each other, and make individual formulations of the flight paths required to avoid conflicts. Obviously, an arbitrator is needed since conflicts may arise between the flight paths generated by the different agents, which is especially true in cases of severe weather conditions with heavy air traffic, such as the terminal airspace near an airport during a storm. The intelligent executive guidance agent, acting as such a high-level arbitrator, receives guidance information from lower-level weather agent and traffic agents. By considering the spatial and temporal characteristics of the conflicting guidance, the executive agent classifies them as either tactical or strategic in nature, and then prioritizes them according to a pre-defined rule base of conflict priorities. The arbitration function thus acts as a fuzzy controller, and gradually switches the guidance between the weather agent and traffic agent, providing conflict free flight path guidance, as the aircraft flies in and out of dangerous regions. Therefore, the executive agent has the capacity and authority to recommend the ultimate flight guidance, and this feature requires it to possess intelligent characteristics such as reasoning, estimating, and decision-making.

The agent system currently is designed for the GA aircraft, which nowadays are not commonly equipped with some routine devices on commercial airlines or private jets. However, the design for the agent system requires the aircraft at least be equipped with the following three on-board devices: one Doppler weather radar with range at least 40 miles, one ADS-B device and one FMS. As the agent system is designed for the future free flight, it is expected these advanced equipments will be common on GA aircraft as well in the near future.

2 TECHNICAL BACKGROUND

2.1 Introduction

This chapter will first introduce the concept of *Knowledge-based control* Then, three important techniques used in creating and implementing the hierarchical agent system are introduced: fuzzy logic, intelligent agents and hierarchical control systems. Finally, the concept of *fuzzy hierarchical control*, which may be regarded as a combined technique of fuzzy logic and hierarchical control, will be briefly described as well.

2.2 Knowledge-Based Control

Knowledge-based control, or so-called *intelligent control*, applies the artificial intelligence methods such as fuzzy logic, neural network or genetic algorithms to generate intelligent or expert controllers.[5] These controllers address the need for robust control in dynamic systems, particularly those in which human operators are involved. Typical scenarios where knowledge-based controllers are useful are those in which decisions (controls) need to be taken based on large amounts of data exchanging. For example, in an aircraft cockpit under free flight conditions, a knowledge-based controller may monitor the data flow, process various type of information, alert pilots to emergences and even take corrective action, so that pilot workload may be drastically decreased and flight safety may be increased. Knowledge-based controllers have also been developed for the control of dynamic systems that operate in environments hostile to humans.

Several AI methods can be used to produce intelligent controllers, of which expert systems; neural networks, genetic algorithms, and fuzzy logic are the more popular ones. The hierarchical agent system introduced in this thesis is actually the application of a *fuzzy expert system* to a conventional *hierarchical control system*. Of course, fuzzy logic should not be the only choice as the AI technology for the current research and some previous research has shown that application of fuzzy logic or neural network to the knowledge control may lead to the similar results.[6] Selection of the fuzzy logic is mainly due to the successful previous applications of fuzzy logic to intelligent control problems, such as GAPATS research project.[7] The design and implementation of the agent system will be described in detail in Chapter 3.

2.3 Fuzzy Logic

Fuzzy Logic is a relatively new technology emerging from Fuzzy Set Theory attributed to Zadeh,[8] which now has become a powerful technology applicable to technical fields such as control, information systems, pattern recognition, and decision support.

Fuzzy logic is a departure from classical two-valued sets and logic. While classical two-valued logic uses strict binary decisions and assignments (true or false), fuzzy logic uses "soft" linguistic system variables (e.g. large; hot; tall) and a continuous range of truth-values in the closed interval [0,1]. Fuzzy Logic may be regarded as a multi-valued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, black/white, etc. Notions like 'rather warm' or 'pretty cold' can be formulated mathematically and processed by computers. In this way, an attempt is made to apply a more human-like way of thinking to the programming of algorithms. One of the main advantages of applying fuzzy logic to generate knowledge-based controllers is the *smooth control surface.* A smooth control surface is required in applications such as aircraft, subway trains, and elevators since passenger ride quality and comfort is a priority in the performance criteria.

2.3.1 Fuzzy Set

The basic notion of Fuzzy Logic is the *fuzzy set.*[9]

First, we introduce a conventional *crisp set.* A crisp set can be defined as a mapping from the elements of the Universe of Discourse to the two-elements set {0,1}. Figure 2.1 is an example of crisp subset A of all real numbers between 5 and 8. It is defined in a universe of discourse X of all the real numbers in the range between 0 and 10. Set A may be interpreted by its characteristic function as following, which assigns a value 1 or 0 to each element in interval [0,10],

$$l_A(x) = 1, x \in A \qquad l_A(x) = 0, x \notin A \tag{2.1}$$

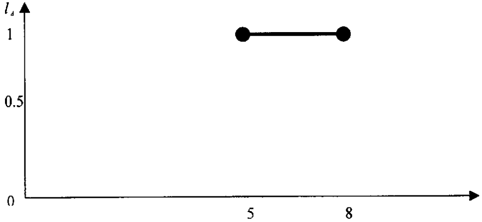


Figure 2.1 Crisp Set A = [5,8]

The elements assigned the value 1 means that the elements are in the set A, while the elements are not in the set A if they are assigned the value 0. Therefore, in the above example, the elements of X are either belonging or not belonging to crisp set A and no third option exists.

A fuzzy set is one to which objects can belong to different degrees, called grades of membership or confidence. Similar to the crisp set, a fuzzy set B can be defined as a mapping between elements of the Universe of Discourse and values in the interval [0,1]. Again, the number 1 assigned to an element signifies that it definitely belongs to the set B. The number 0 signifies that the element is definitely not in the set B. All other values mean a gradual membership to the set B. The mapping is often described as a function, for instance the Membership Function (MF) of B. Figure 2.2 is an example of a fuzzy set B.[9]

For example, suppose set B defines a set of young people according to their age. Instead of constructing the set B with strict separation between young and not young, we could allow more flexible descriptions like he/she belongs a little bit more to the set of young people, or he/she belongs nearly not the set of young people. These descriptions can be displayed graphically in Figure 2.2. As shown in this figure, a 25-year-old person would still be young to a degree of 50 percent according to the methodology.[9]

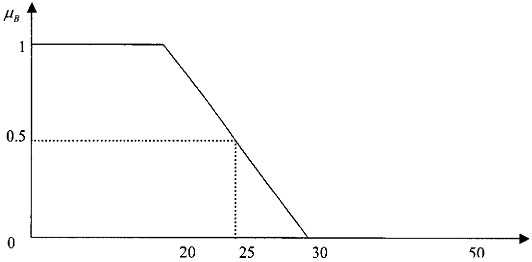


Figure 2.2 Membership Function of fuzzy set B = {set of young people}

Operations can be performed on fuzzy sets. Figures 2.3- 2.4 show examples of two fuzzy sets A and B. Figures 2.5-2.7 show some examples of the operations performed on the two sets.

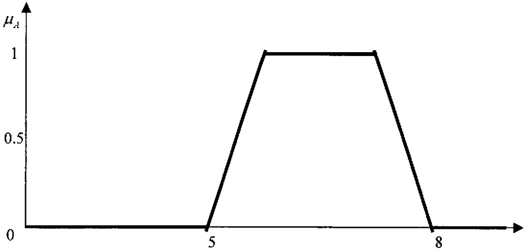


Figure 2.3 MF of fuzzy set A

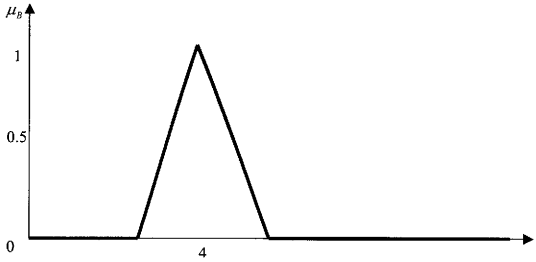


Figure 2.4 MF of fuzzy set B

Union $\mu_{A\cup B} = \mu_A \vee \mu_B$ $\vee$: Maximum Operator

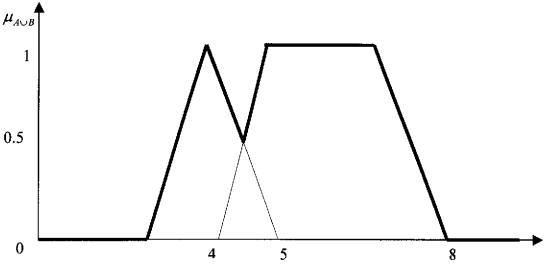


Figure 2.5 Union Operation of fuzzy sets

Intersection $\mu_{A \cap B} = \mu_A \wedge \mu_B$ $\wedge$: Minimum Operator

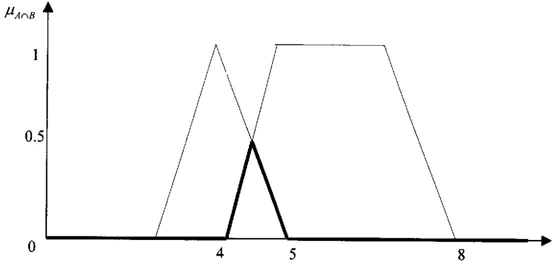


Figure 2.6 Intersection Operation of fuzzy sets

Complement $\mu_{\bar{A}} = 1 - \mu_A$

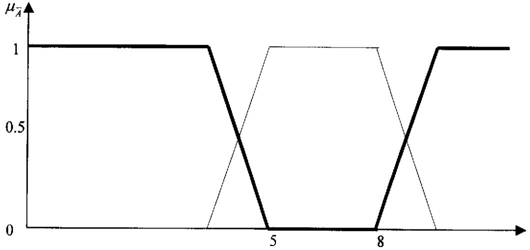


Figure 2.7 Complement (Negation) Operation of fuzzy set

2.3.2 Fuzzy Expert Systems

A *Fuzzy Expert System* is an expert system that uses a collection of fuzzy membership functions and fuzzy rules, instead of Boolean logic, to manipulate data. The set of fuzzy rules in a fuzzy expert system are called the *Rule Base*, or *Knowledge Base*. A fuzzy rule is usually of a form similar to the following:

IF (certain specified patterns occur in the data) **THEN** (take the appropriate actions, including modifying old data or asserting new data)

The left-hand side of the rule, the IF part, is technically called the *Antecedent*, and the right-hand side, the THEN part, is the *Consequent*. The antecedent of a fuzzy rule describes an elastic condition that can be satisfied to a certain degree. This is different from the antecedent of a conventional rule, which is a rigid condition that is either satisfied or dissatisfied. The consequent of a fuzzy rule depicts a conclusion that may be drawn when the condition is satisfied. It may be either a *crisp consequent*:

IF…THEN y = a, where a is nonfuzzy numeric value or symbolic value

or a *fuzzy consequent*:

IF…THEN y is A, where A is a fuzzy set

or a *functional consequent*:

IF x_1 is A_1 AND IF x_2 is A_2 AND … IF x_n is A_n THEN y = $a_0 + \sum_{i=1}^{n} a_i \times x_i$, where

$a_{0,} a_1, \ldots, a_n$ are constants.

An example of a fuzzy rule is as following:

If x is *low* and y is *high* then z is *medium* where x and y are input variables and z is an output variable. *Low, high and medium* are all fuzzy sets, each of which has a defined membership function. In the antecedent of this fuzzy rule, input variables x and y are assigned to some fuzzy values like *low* and *high*, instead of some precise and nonfuzzy value.

If more than two input variables exist in the antecedent, *Antecedent Operators* should also be defined. *Intersection (AND)* and *Union (OR)* are the most common choices for antecedent operators. There are many ways to define these two operators, and the point wise *"Product/Sum"* and *"Min/Max"* are often used. Figure 2.8 illustrates these four antecedent operators.

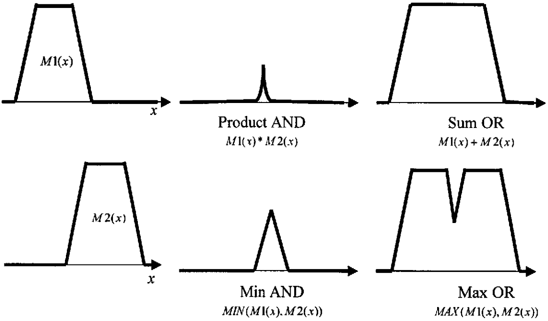


Figure 2.8 Antecedent Operators

The general inference procedure for constructing fuzzy experts systems consists of three steps: *Fuzzification, Inference and Defuzzification.* Figure 2.9 illustrates the three steps in *Inference Procedure.*

Under fuzzification, the membership functions of fuzzy values assigned to the input variables are applied to actual values of input variables, to determine the degree of truth (confidence) of each rule premise.

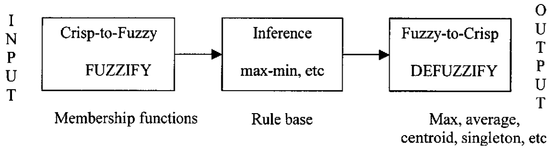


Figure 2.9 Inference Procedures

Inference is subdivided into two steps. Generally, the first one is *correlation*, and the second is *combination*.

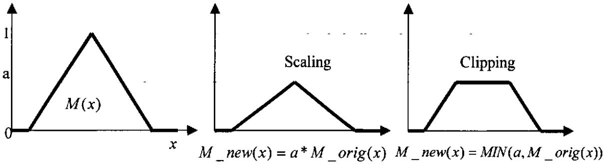


Figure 2.10 Correlation Operators

During correlation, the confidence of the antecedent of each rule is computed, and applied to the consequent of each rule. Many kinds of operators exist for this step, and *product correlation* and *minimum correlation* are often used. In product correlation, the output membership function is scaled by the confidence of the antecedent of the rule (scaling). In minimum inference, the output membership function is clipped off at a height corresponding to confidence of the antecedent of the rule (clipping). The two operators are illustrated in Figure 2.10.[10]

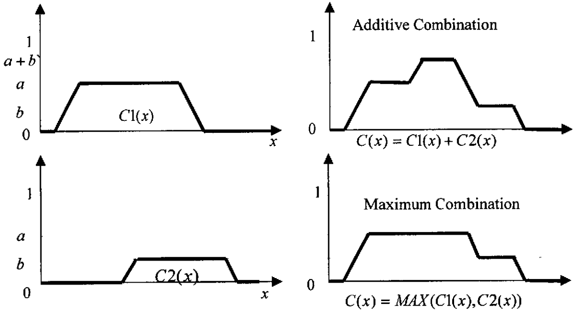


Figure 2.11 Combination Operators

For combination, all of the fuzzy subsets assigned to each output variable are combined to form a single fuzzy subset for each output variable. Similar to correlation, many operators exist in this step also, among which *maximum combination* and *summation combination* are two often used. In maximum combination, the combined output fuzzy subset is constructed by taking the point wise maximum over all of the fuzzy subsets assigned to variables by the inference rule. In summation combination, the combined output fuzzy set is constructed by taking the point wise sum over all of the fuzzy subsets assigned to the output variable by the inference rule. Figure 2.11 illustrates these two operators.[10]

The final step is defuzzification, which is used to convert the fuzzy output set to a crisp number. *Centroid* and *Mean of Maximum* are two most common operators used in this step. In the centroid method, the crisp value of the output variable is calculated as the weighted average of a fuzzy set. The mean of maximum method calculates the average of all variable values with maximum membership degrees.

2.4 Intelligent Agents

An *agent* is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.[11] An agent may be constructed by making a mapping which specifies what action the agent should take in response to any given percept sequence. Agent programs are designed by implementing the agent mapping from percepts to actions, and require information either quantitative or qualitative on the possible percepts and actions of the agent, what goals or performance measure the agent is supposed to achieve, and what sort of environment it will operate in.[9] This is called as the PAGE (Percepts, Actions, Goals, Environment) description. The following table is an example of a PAGE description of a medical diagnosis agent.

Table 2.1 Example of PAGE description of a medical diagnosis agent

	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings Patients answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital

Agents may be classified in several categories. *Reflex agents* respond immediately to percepts, *goal-based agents* act so that they will achieve their goal(s), and *utility-based agents* try to maximize the utility.[11] The environment of an agent has properties of different types, which may be distinguished as *accessible* vs. *inaccessible*, *deterministic* vs. *nondeterministic*, *episodic* vs. *nonepisodic*, *static* vs. *dynamic*, and *discrete* vs. *continuous*. All the agents developed in the current research may be categorized among goal-based agents as they have clear objectives to avoid specific types of conflicts and at the same time, they possess capacities to generate certain plans to achieve these objectives.

2.5 Hierarchical Control Systems

Utilizing the structure of hierarchical, multilevel systems, hierarchical control provides an efficient way to control large or complex systems. It examines parts of the system at different levels of abstraction and detail, and then organizes the sequence of tasks to be executed in a hierarchical way.

The common characteristic in all sorts of hierarchical control systems is that the decision-making process of the overall system has been divided into several levels. There may be several *decision-maker units* in the structure, but not all of them directly access the controllers. The other decision-maker units that define the tasks and coordinate the lower units are at a higher level in hierarchical structure. Just like the human decision point of view, since one person or one unit would not be able to make all of the decisions required to run a complex organization, it is quite reasonable to divide the decision making process into several levels.

Generally, there are many levels in a hierarchical control system. Normally, the highest level of the hierarchical struture defines the tasks and coordinations, while the lowest level has direct access to the controllers. And within each layer of the strucuture, there are different control objectives handled by different controllers or operating procedures. There are lots of good reasons for organizing the control of large systems in a distributed hierarchy. Among these are: deeper understanding facilitated by the hierarchical structure, reduction in complexity of communication and computation, modularity and adaptability to change, robustness, and scalabity. [12]

2.6 Fuzzy Hierarchical Control

In this research, fuzzy logic has been utilized in developing intelligent hierarchical control systems. Within a hierarchical system, the higher levels handle the more abstract views of a problem and thus represent the information in a more qualitative form. The lower levels deal with quantitative information more efficiently. In addition, the higher levels depend more on heuristics while the lower levels depend more on differential algebraic formulations. In this sense, in a hierarchical intelligent control system, fuzzy logic may be best utilized for the high-level or supervisory functions. It may handle knowledge represented as linguistic statements, obtained from human experts and containing soft, qualitative terms. Fuzzy If-Then rule-based systems are developed as well so that the rule-base systems the fuzzy controllers may perform tasks like estimating, arbitrating or decision-making, based on inference.

The proposed agent-based system is a hierarchal system composed of two low-level agents and one high-level agent. All the three agents maybe regarded as intelligent as they are designed to cooperate with the pilot to perform some specific aviation tasks. Specifically, a fuzzy expert system is the primary technology utilized to implement the executive agent, and it is demonstrated to be an efficient method in realizing the necessary intelligent behaviors of executive agent, such as estimating, reasoning and decision-making. Implementation of the agent-based system will be introduced in detail in the next few chapters.

3 HIERARCHICAL AGENT BASED SYSTEM

3.1 Introduction

This chapter details the design and implementation of the agent system. The weather and traffic agents are the primary components of the system, and both are introduced. The differences between the two agents are highlighted, as they play a critical role in selecting a sequential conflict resolution process. Finally, the executive agent is developed at length in this chapter. The executive agent is the highest-level component of the agent system, and its design and implementation is the primary objective of this research.

3.2 Weather Agent

A weather agent detects severe weather conditions based on data received from onboard weather radars, meteorological satellites, and data-linked ground weather information services. Considering the weather restrictions, the weather agent computes an optimal flight path that ensures safe passage through the weather. Any destructive weather phenomenon is known as severe weather, while this term usually refers to localized storms. These weather conditions correspond to the localized regions of strong wind shear, violent updrafts and downdrafts and heavy downpours, like thunderstorms, microbursts, tornados and squall lines, which all can cause considerable damage to an aircraft.[4] Since the currently designed weather agent concentrates mainly on thunderstorms and squall lines, the following sections briefly introduce both of these two weather conditions and their threats to the flight safety of an aircraft.

3.2.1 Thunderstorm

A thunderstorm is a convective storm accompanied by lightning, thunder, and a variety of weather such as locally heavy rain showers, hail, updrafts, downdrafts, and sudden temperature changes.[4] Thunderstorms are typically towering clouds with anvil shaped tops as seen in Figure 3.1, and they travel from speeds near zero to 60 miles per hour[4].

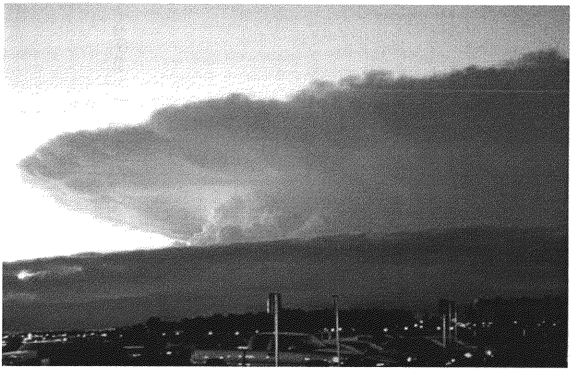


Figure 3.1 Storm Structure in Thunderstorm

3.2.2 Squall Line

A Squall line is a line of thunderstorm cells as seen in Figure 3.2 that is accompanied by a continuous gust front near its advancing edge.[4] They are 12-50 miles wide and a few hundred to 1250 miles long.[4] Strong gusty winds, rapid temperature drop, heavy rain, thunder and lightning, and often hail and tornadoes usually accompany them.[4]

Figure 3.2 A Weak Squall Line Passing through Central Oklahoma

3.2.3 Squall Line Hazard to Aircraft

Severe thunderstorms pose great threat to the flight safety of an aircraft. The hazards associated with the thunderstorms primarily arise from the turbulence. The violent updraft and downdraft of the turbulence may destroy aircraft or drive them to the ground.[4] Some other hazards related to the thunderstorms are icing and lightening. Hence, the safest flight path is the one that avoids the thunderstorm cells. Flight through a thunderstorm cell and under a cluster of thunderstorms should be strictly avoided.[4]

3.2.4 Severe Weather Avoidance Algorithm

Embedded within the weather agent is a squall line detection and avoidance algorithm. In the current research, the A^* search method with slight modification is the algorithm used for the weather agent which takes radar images of the thunderstorms as input, and determines the safest path between the two points in the flight with minimum detour[17]. Simulated Doppler weather radar data is currently used for the weather agent.

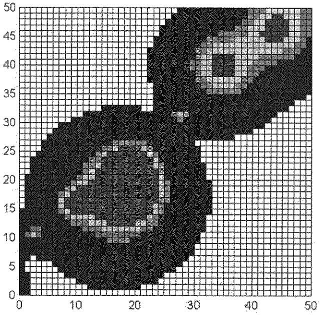


Figure 3.3 Simulated Radar Image of a Thunderstorm

Figure 3.3 illustrates a simulation of the radar image of a thunderstorm provided by Doppler weather radar. See the M.S thesis by Bokadia for complete details.[4]

3.3 Traffic Agent

The traffic agent detects air traffic in the neighboring airspace and keeps the aircraft out of the protected zones of other aircraft. A combination of a knowledge based expert system and optimal control is utilized in the traffic agent for the traffic conflict detection and resolution module[3]. The agent takes ADS-B state vectors of other aircraft in the immediate airspace as input, and provides traffic conflict detection and warnings. Safe airborne separation between aircraft is maintained using protected and alert zones as illustrated in Figure 3.4. An aircraft's *protected zone* surrounds the aircraft and should never overlay with another aircraft's protected zone.[2] The *alert zone* surrounds a larger area, and the aircraft can maneuver freely until its alert zone meets with another alert zone.[2] The size of the zones is determined by the aircraft's velocity, performance, communication, navigation and telecommunication equipment.[2]

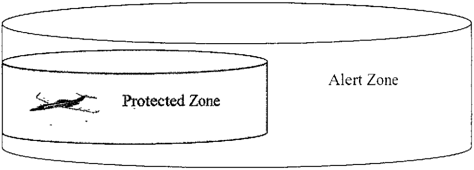


Figure 3.4 Protected Zone and Alert Zone

Incorporated within the traffic agent are a traffic conflict detection module and a traffic conflict resolution module. The former seeks and detects all traffic within the alert zone, and passes the valid traffic information to the conflict detection module and traffic information display.[3] A traffic conflict is assumed to take place when the estimated flight trajectory of an objective aircraft passes through the protected zone of the subjective one. The traffic conflict calculation is based on the horizontal radius and height of the protected zone, and takes into account the aircraft's flight zone, position, and flight mode for its calculation.[3] If a conflict is detected, the traffic status and conflict information is passed to the resolution module and the pilot's display. The resolution module utilizes an expert system and optimal control to determine the best maneuver to avoid the traffic, which may be selected from several available options like a turn maneuver, a climb maneuver or a combination of a turn and climb maneuver.[3] A rule-based system was constructed to select the best available maneuver.[3]

For the expert system part of the traffic agent, knowledge and expertise of a pilot and ATC is required. Air traffic regulations and Instrument Flight Rules (IFR) should be properly included into its knowledge base, such that the expert system is technically sound and sufficiently complete.[3] For the optimal control part, the selected trajectory is optimized using an objective function consisting of the delta magnitude of acceleration.[3]

$$J = \sum_{i=1}^{m} \ddot{x}_{a_i} - \ddot{x}_i$$

$$t = t_o \rightarrow i = 1$$

$$t = t_f \rightarrow i = m$$

The primary constraint used for the optimization is the required separation between aircraft. There are some secondary constraints like aircraft states limitation, aircraft performance limitation and time to maneuver limitation.[3] The optimizer applies a fixed step size 'Breadth First Search' setup, such that the solution obtained is minimum acceleration, and satisfies all of the active constraints imposed on the system.[3] The reader is referred to the M.S. thesis by Shandy for complete details.3

3.4 Integration of Weather Agent and Traffic Agent

As two separate and independent agents, the weather agent and the traffic agent have inherently different characteristics, both spatially and temporally. For instance, a typical squall line covers a large airspace that may be more than hundreds of miles in length and tens of miles in width. In order to guide the aircraft past or through the squall line, a weather agent must recommend a new flight path with perhaps significant deviation from the original planned flight path. This can increase flight time and therefore increase fuel expenditure and cost. In comparison, the required en-route protected zone of an aircraft over United States airspace is quite small, having a radius of 2.5 nautical miles and a height of 2000 ft (1000 ft below 29000 ft, 4000 ft over oceanic airspace) [13]. The traffic agent therefore need only request small changes in airspeed, heading, or altitude to keep the aircraft out of the protected zones of other aircraft. For these reasons, the weather agent is conceptualized as being strategic in nature, while the traffic agent is considered tactical. Another difference between these two agents is related to how frequently weather or traffic conflicts might occur along the entire flight path of an aircraft. Though the pre-flight weather is notoriously inaccurate and weather develops extremely rapidly, pilots must account for en-route weather conditions during pre-flight planning. Thus, opportunities for the aircraft to encounter unexpected severe weather condition during the flight are greatly decreased. Conversely, since traffic conflicts occur much more frequently and unexpectedly, especially in congested airspace near airports, the traffic agent is likely to be exercised much more often during a given flight.

The different characteristics of the two agents are addressed in the design of the agent system. The general philosophy in the system design is solving the weather conflict (using the weather agent) and the traffic conflict (using the traffic agent) sequentially, which will be introduced in detail in the next section. Compared to parallel processing, sequential processing is a more intuitive choice, which directly addresses not only the simplicity of realization of the necessary fuzzy control systems, but also the different inherent characteristics of the two agents. However, it does not mean that the parallel processing has obvious disadvantages or is not suitable to the current problem, as by now little research has been done on the application of the parallel processing.

3.5 Sequential Conflict Resolution Process

In Lass's M.S thesis [Ref. 14], a fuzzy system composed of dual experts with one arbitrator was established as part of the knowledge-based aircraft approach controller. Each expert was developed to meet a specified goal, one being to follow the flight plan and the other to keep the aircraft in a safe operating region. The arbitrator acted as a fuzzy controller and made the decision using a pre-defined rule base. It gradually switched the controls from one expert to another expert, as the aircraft moved in and out of dangerous regions. A similar idea is adopted in creating the agent based hierarchical system, in which the executive agent plays a role similar to that of Lass's arbitrator.

In the agent based system, weather conflicts should be solved first, since, (as mentioned in the previous section), weather conflicts in general cover a large area and require a large magnitude of deviation from the initial planned trajectory to avoid unsafe regions. A large trajectory deviation will result in a change in the future air traffic that might be encountered, since the path taken is different. Thus, traffic information should also be processed to reflect the path changes proposed by the weather agent. Note that traffic conflict detection calculations need to be re-evaluated for the proposed weather conflict-free trajectory. In addition to the large deviation required to avoid bad weather, such conflicts in general tend to be less dynamic but more persistent than traffic conflicts. Thus, based on the nature of the problem, generation of weather conflict-free trajectories is assigned as the first priority in the system. The traffic agent will utilize the proposed conflict-free path (if weather conflicts exist) to calculate the new guidance maneuver necessary to satisfy its own constraints.

The sequential conflict resolution process also considers that the traffic agent operates far more frequently than the weather agent. Generally, the traffic agent is required to execute every five or ten seconds for a GA class aircraft, [3] while the weather agent may execute only every one or two minutes.[4] This means that during the interval between two consecutive executions of the weather agent, the traffic agent may have already run five times or more.

In the agent system, the traffic agent is engaged after the system finishes sequencing through the weather conflict avoidance tasks. The sequential prioritization of the weather agent and traffic agent is a method similar to the hybrid approach used in mobile robot navigation. The weather agent acts as a model-based planner to generate a flight path from an incomplete model of the environment where only weather conflicts exist. This flight path is then used by the traffic agent, which acts as a sensor-based controller to navigate the aircraft such that it follows the path while evading potential traffic conflicts to the aircraft.

If the weather agent detects weather conflicts and issues a new flight path, the traffic information used by the CD&R module of the traffic agent will reflect this path change. However, a conflict still possibly arises between the two agents as the traffic agent performs the traffic CD&R calculation only from the perspective of current traffic information and disregards any weather conditions. In the sequential conflict resolution process, a conflict between the weather and traffic agents means: if the aircraft maintains the flight path recommended by the weather agent, it will encounter a traffic conflict. Meanwhile, a new traffic-conflict free path issued by the traffic agent to avoid this traffic conflict leads the aircraft towards the area of high thunderstorm intensity. Conflict between the two agents usually happens when the aircraft flies near the thunderstorm, or through a squall line. Therefore, in case both the weather and traffic agents detect potential conflicts and each of them provide an individual new flight path, the executive agent acts as an arbitrator, coordinating between the two agents and resolve the potential conflicts between them. A scheme for the executive agent to fuse the recommended paths from two agents is critical and is developed in section 3.6.

Since the FAA prohibits General Aviation aircraft from flying through and above thunderstorms, the weather agent in its current form only recommend the motion in a two-dimensional horizontal plane.[4] However, the traffic agent provides the avoidance behaviors in a three-dimensional plane. Conflicts between the weather and the traffic agents only possibly arise when both of them recommend horizontal flight path changes, and no conflict occurs when the traffic agent only issues a change in altitude. Therefore, in the current research, the executive agent is designed as a two dimensional agent and recommends resolution for the conflicts in the horizontal plane.

3.6 Executive Agent

The executive agent is a fuzzy controller that acts as an arbitrator. It is composed of three internal modules: a weather conflict evaluation module, a traffic conflict evaluation module, a rule-based arbitrator and two auxiliary modules, as shown in Figure 3.5. The primary inputs to the executive agent are the new waypoints on the flight paths that are recommended separately by the weather and the traffic agents, attempting to resolve the current weather and traffic conflicts. Other inputs include the latest information on weather conditions provided by a radar image, and the current traffic situation.

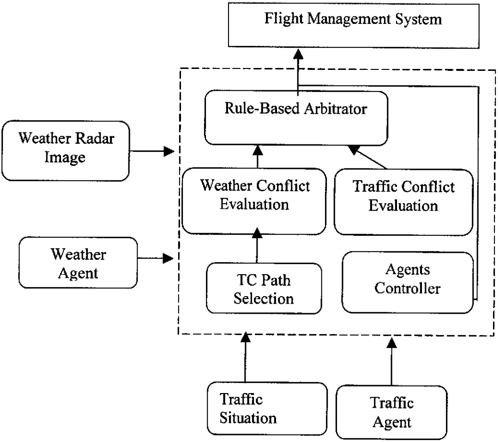


Figure 3.5 The Executive Agent Structure

At mentioned in Chapter 1, the executive agent possesses certain intelligent attributes like reasoning, estimating and decision-making. In this research, this functionality is realized by applying one common fuzzy decision-making technique—fuzzy synthetic evaluation. The term *synthetic* is used to denote the process of evaluation whereby several individual elements and components of an evaluation are synthesized into an aggregate form; [15] The whole is a *synthesis* of the parts.[15] The term *fuzzy* indicates that the various elements are non-numeric and represented in terms of fuzzy variables. The synthetic evaluation performed by the executive agent on the overall conflict situation is based on two components, the current weather condition and the current traffic scenario. The weather and the traffic conflict evaluation modules calculate respectively the severity of the weather, and traffic conflicts in terms of fuzzy terms such as "slight", "severe", etc. The qualitative values of weather and traffic conflicts are passed as inputs to the rule-based arbitrator that contains a Mamdani fuzzy rule inference system with 16 fuzzy rules. Weighing between the extent of the present weather and traffic conflicts, the arbitrator endeavors to avoid the potential collisions between the two agents. It controls the severity of both conflicts to an acceptable level and provides an optimal solution to resolve both severe traffic and weather conflicts. The output from the inference system will decide whether the aircraft should completely follow the traffic conflict free flight trajectory provided by the traffic agent, or deviate from it and fly towards a new heading in case the recommended traffic conflict free path may encounter a significant weather conflict. It also needs to determine how much the deviation should be if it is the latter case. Thus, the design objective of the executive agent acting as an arbitrator and coordinator of low-level agents may be realized. Each component of the executive agent is described in detail in sections 3.6.1to 3.6.3.

Choosing the membership function for the weather conflict, the traffic conflict and the necessary adjustments is a completely ad hoc procedure, which turns out to be the most difficult and time consuming work in the implementation of the executive agent. It is easy to define a set of fuzzy rules, but deciding the shapes making up the membership functions is much harder. The selection of the membership functions in this research is most based on intuition, instead of other sort of automated mechanism like neural network or generic algorithm. For example, in the traffic conflict severity module, there is no well-founded reason for why choosing the linguistic values as described in the next section. The variable Traffic Conflict may be defined in may other ways as well. For example, *None* may be replaced by *No* as long as they have the similar syntax meaning and same membership functions, so may *Slight* be replaced by *Small* or other words. Furthermore, instead of four linguistic values, the variable "traffic conflict" may have three values, e.g., *None*, *Slight* and *Severe*, or five values, e.g., *None*, *Slight*, *Middling*, *Severe* and *Fatal*. However, after several times of trial, it shows that the number of the fuzzy values has no great effect on the system performance. In addition, the types of membership functions of these linguistic values need not be triangular as illustrated in Figure 3.7. Other types that might also be used are trapezoid functions, or Gaussian functions. However, as long as the fuzzy values of the variable "traffic conflict" are defined in a reasonable way and the executive agent performs as desired, triangular function becomes a natural choice, as it is the simplest type at all.

3.6.1 Traffic Conflict Severity

The traffic conflict evaluation module determines the possibility of occurrence of traffic collisions with other aircraft and the severity of the traffic conflicts at the current moment. An important issue is determining the present severity of the traffic conflict, in the context of how far away the conflict point is to the current aircraft position, and how soon the collision will actually occur. Of course, the closer the two aircraft are, the more severe the traffic conflict is. The vertical and horizontal separation between one aircraft and another are required to determine conflict severity and avoidance urgency. If the protected zones of two aircraft overlap at a moment when they are nearest to each other, a traffic collision is said to occur.

As the executive agent is a 2-dimensional agent, the traffic conflict is only concerned in the horizontal plane. The quantitative value of the traffic conflict between any two aircraft may be simply defined as a function of the current distance between the two aircraft, the radius of protected zone and alert zone as shown in formula (3.1)

$$\begin{aligned} & TF_i = 1 - \frac{\text{Current\_Distance-2*R\_Protected\_Zone}}{2*\text{R\_Alert\_Zone}} \\ & \text{if } TF_i > 1,\ TF_i = 1 \qquad (3.1) \\ & \text{if } TF_i < 0,\ TF_i = 0 \end{aligned}$$

From the above formula, the value of TF_i becomes one when the current distance between two aircraft is less than the diameter of the protected zone, indicating that a traffic conflict occurs between the two aircraft horizontally. If more than one traffic conflict exists, then

$$TF = \underset{i}{MAX}(TF_i) \qquad (3.2)$$

which means the current overall severity of the traffic conflict is defined simply as the max among the traffic conflicts brought on by all of the bogey aircraft.

The size of the protected zone set in the traffic conflict module is 1.5 miles in radius, [3] which already taken into account flight path tracking errors and positional errors caused by GPS inaccuracy. As described before, the radius of the alert zone is determined by various factors. For simplicity, relative velocity between the two aircraft is the only factor considered in the current research and the radius of the alert zone is defined as follows:

$$\text{R\_Alert\_Zone} = \text{Relative\_Distance\_Difference}/2 + \text{R\_Protected\_Zone} \tag{3.3}$$

where Relative_Distance_Difference implies the change of the distance between the two aircraft in one minute. The formula (3.1) is rewritten as

$$\begin{aligned} &TF_i = 1 - \frac{\text{Current\_Distance-2*R\_Protected\_Zone}}{\text{Relative\_Distance\_Difference} + 2 * \text{R\_Protected\_Zone}} \\ &\text{if } TF_i > 1,\ TF_i = 1 \\ &\text{if } TF_i < 0,\ TF_i = 0 \end{aligned} \tag{3.4}$$

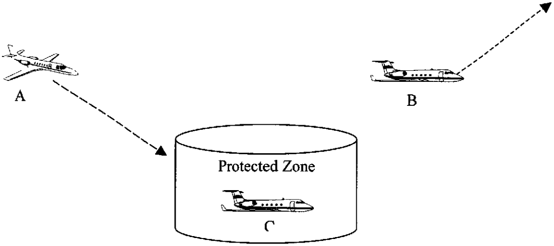


Figure 3.6 Flying towards or Flying away

Formula (3.4) illustrates another important aspect in determining the traffic conflict, the rate of closure between approaching aircraft. As shown in Figure 3.6, aircraft A is more of a threat to aircraft C than aircraft B, since it flies in a direction towards the protected zone of C, while B is flying away from C. Therefore, even though both A and B may be equidistant from C at the current moment, aircraft A poses a more serious traffic conflict than aircraft B does. The difference of the distance between two aircraft from moment to moment is used to measure the rate of closure. A rapidly decreasing distance between two aircraft toward each other indicates a severe traffic conflict. If the distance between two aircraft decreases slowly or even if one aircraft flies away from another one, the probability of a traffic conflict is low.

The value of TF is a quantitative number between 0 and 1 and it needs to be first fuzzified into a qualitative number before fed into any fuzzy inference system. A fuzzy variable *Traffic Conflict Severity* is defined, which is illustrated in Figure 3.7. Its universes of discourse ranges from 0 to 1 and it has four linguistic values: *None*, *Slight*, *Middling* and *Severe*, which represent the different degrees of traffic conflict.

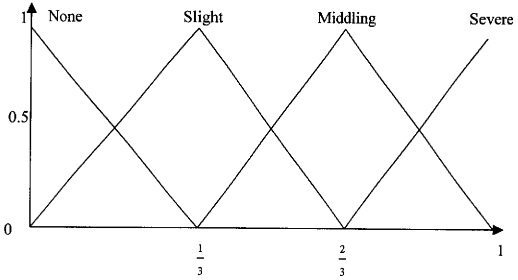


Figure 3.7 Membership Functions for Traffic Conflict Severity

The value of TF in formula 3.2 is fed into the fuzzy variable Traffic Conflict, and its qualitative value may be obtained. For example, if TF is 0.5, then the current Traffic Conflict is *Slight* to the degree of 0.5 and *Middling* to the degree of 0.5 as well. It is *None* or *Severe* to the degree of 0, which means it is neither *None* nor *Severe*.

3.6.2 Weather Conflict Severity

Similar to the traffic conflict evaluation module, the weather conflict evaluation module determines current severity of the weather conflicts. It contains a simple fuzzy system deriving the qualitative value of the current order of weather conflict severity, illustrated in Figure 3.8. The universes of discourse of linguistic variable Weather Conflict is set upon the thunderstorm intensity ranging from 0 dBZ to 30dBZ. It has four fuzzy values: *None*, *Slight*, *Middling* and *Severe* and membership functions for these linguistic terms are based upon the definition of the prohibited zone for aircraft in thunderstorms. In the weather agent, aircraft may not fly into area with thunderstorm intensity greater than 30 dBZ. Therefore, the area with the thunderstorm intensity greater than 27 dBZ is considered to impose severe weather conflict. The input to the fuzzy system is the maximum storm intensity in the aircraft adjacent airspace.

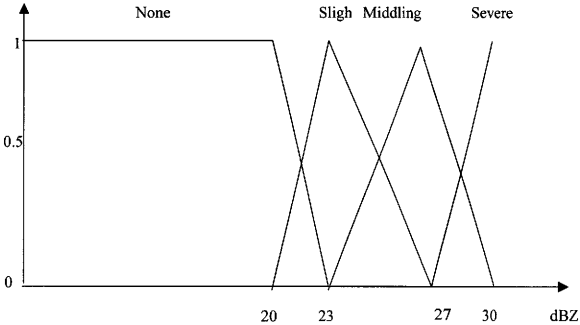


Figure 3.8 Membership Functions for Weather Conflict Severity

Besides determining the severity of the current weather conflict, another function of this module is calculating the safest or least weather conflict heading for the aircraft. As the executive agent executes every five second, (choice of the update rate is explained in Chapter 6), a forty second look-ahead horizon (looking eight steps forward), is a satisfactory provision for the calculation. This means the aircraft will pass through the area of least storm intensity in its adjacent airspace for at least the next forty seconds, if it flies along this *Safest Heading*. The zone of regard for determining such a heading is a fan shape originating from the current position of the aircraft. Its angle is the difference between the headings of the new waypoints provided respectively by the weather agent and the traffic agent. Its radius is generally set as the distance the aircraft may cover in the next

40 seconds with the current airspeed. An example of such area is shown as area 012 in figure 3.9.

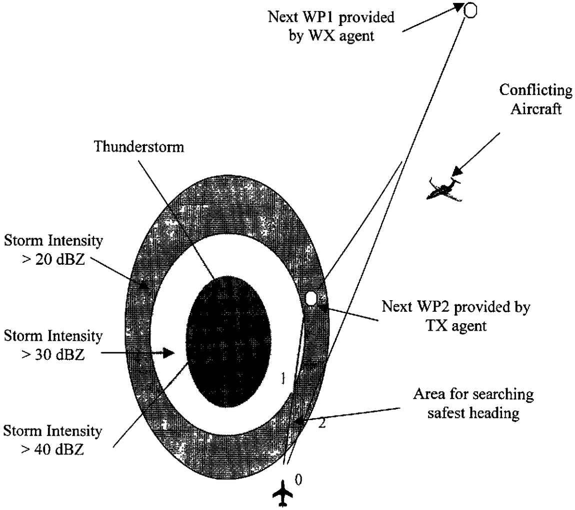


Figure 3.9 Typical Weather and Traffic Conflict Scenario

3.6.3 Rule-Based Arbitrator

One fuzzy Mamdani rules-based inference system is established in the rule-based arbitrator, as illustrated in Table 3.1.

Table 3.1 Rules for Overall Conflict Inference System

IF Traffic Conflict	**AND** Weather Conflict	**THEN** Adjustment
None	None	No
None	Slight	Small
None	Middling	Big
None	Severe	Radical
Slight	None	No
Slight	Slight	Small
Slight	Middling	Moderate
Slight	Severe	Huge
Middling	None	No
Middling	Slight	No
Middling	Middling	No
Middling	Severe	Moderate
Severe	None	No
Severe	Slight	No
Severe	Middling	No
Severe	Severe	Small

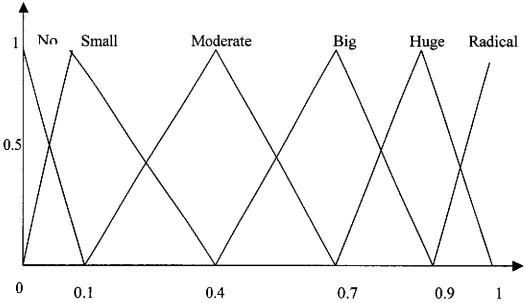


Figure 3.10 Membership Functions for Adjustment

The antecedent operator used in the inference procedures is the *Min* operator, and the correlation method and the defuzzification method are the *Min* and *Mean of Maximum* respectively. Fuzzy values of Traffic Conflict and Weather Conflict are input to the arbitrator from the traffic conflict evaluation module and the weather conflict evaluation module respectively. *Adjustment* is another fuzzy variable with six linguistics values: *No, Small, Moderate, Big, Huge* and *Radical*, as illustrated in Figure 3.10. It implies how much the aircraft should deviate from the traffic conflict free flight path recommended by the traffic agent in case it passes through airspace with high thunderstorm intensity. The adjustment is intended to be great if the current weather conflict is serious while the current traffic conflict is small. In situations where both the weather conflict and traffic conflict are severe, resolving the traffic conflict should have higher priority than avoiding the weather conflict, which means the adjustments should be moderate in these cases. However, in such extreme cases, an alert will be issued and displayed on the FMS interface. Thus, the pilot is notified of the dangerous situation and may choose to fly the aircraft manually if he only trust himself in these cases, though the agent system guarantees to provide a safe flight path. The adjustments should also be small if weather conflicts are not severe. The final output from the inference system, the defuzzified value of fuzzy variable Adjustment, is a real number, A, having value between 0 and 1.

Another input to the arbitrator is the Safest Heading determined by the weather conflict evaluation module described in the previous section. The executive agent commands a new heading for the aircraft according to the following:

$$\text{cmdHDG} = \text{safestHDG}*A + (1-A)*\text{txHDG};\ 0<=A<=1 \tag{3.3}$$

where cmdHDG, safestHDG, txHDG denote the final commanded heading of the executive agent, the Safest Heading, and the heading towards the next waypoint recommended by the traffic agent. Therefore, if Adjustment is equal to 0, the aircraft will follow the flight path towards the waypoint provided by the traffic agent. Otherwise, if Adjustment equals to one, the aircraft will turn to the "safest heading". Except for these two extreme situations, the aircraft will generally change it's heading to a new one between the headings recommended by the traffic agent and the "safest heading", according to the weighted average of (3.3).

The above tactics employed by the executive agent to arbitrate and coordinate between lower level agents may be explained more clearly using an example. As illustrated in the previous Figure 3.9, an aircraft originally flies along the flight path towards waypoint one, provided by the weather agent to avoid a thunderstorm on its left side. However, at some time the traffic agent detects a potential traffic conflict caused by another aircraft, and issues a new flight path towards waypoint two for the aircraft to avoid this traffic conflict. The two flight paths are both sent to the executive agent for arbitration. First, the executive agent finds that the flight path recommended by the traffic agent leads the aircraft to an area of higher thunderstorm intensity (Figure 3.10), which means a potential conflict arises between the weather and traffic agents. Then, the executive agent calculates the current severity of weather and traffic conflicts. At the current moment the weather conflict is severe, since the aircraft is close to a thunderstorm, the traffic conflict is slight, as the bogey aircraft is still far away. Therefore, instead of flying directly towards waypoint 2, the next waypoint recommended by the traffic agent, the executive agent reasons that the aircraft should fly along a new heading that may cross over the thunderstorm area of less intensity. So it searches for the safest heading within area 012, which may be towards waypoint one or not. Finally, it derives the defuzzified value of Adjustment based on the qualitative values of the current Weather and Traffic conflict, using the rule-based inference system in Table 3.1. The final commanded heading is obtained using formula (3.3). The Flight Management System and the autopilot system of the aircraft then guides the aircraft to turn and fly along the new heading until the executive agent runs again, and a newer heading is issued.

Besides the three primary modules, the executive agent also contains another two auxiliary ones. In the agent system, the traffic agent provides two traffic conflict free flight paths to the executive agent, one is the optimal one and the other is less optimal. The TC path selection module checks both of the paths. If it detects that the optimal one passes through the airspace with thunderstorm intensity more than 28 dBZ, from the two traffic conflict free flight paths, it chooses the one across the area with less thunderstorm intensity, which reduces the risk of increasing weather conflicts. The update frequencies for the weather and traffic agents are predefined, however, the agent controller module may call the weather and traffic agents when necessary. It activates the weather agent when the aircraft deviates from the weather conflict free flight path more than 2 miles, as it turns out to be waste of time and fuel to return to the old flight path instead of finding a new weather conflict free path from the current position. It activates the traffic agent when the speeds, headings or rates of climb of bogey aircraft change, which may give rise to new traffic conflicts. The evaluation of the agent system shows that introduction of the two auxiliary modules improves the performance of the whole agent system.

4 VALIDATION OF AGENT SYSTEM BY SIMULATION

4.1 Introduction

Validation of the agent system by real-time, fixed-base simulation is an important task for this research project, and distinguishes it from other similar research in related fields. This chapter will introduce some necessary equipment and software for the validation of the agent system, describing the entire architecture of the integrated, enhanced flight simulation system that contains the existing and new modules.

4.2 General Aviation Pilot Advisor and Training System

The General Aviation Pilot Advisor and Training System (GAPATS) is a computerized airborne expert system developed under a previous research project sponsored by NASA Langley Research Center.[7] GAPATS uses fuzzy logic to infer the flight mode of an aircraft from sensed flight parameters. This inference, along with an embedded knowledge base and pilot inputs, is used to assess the pilot's flying performance, and recommendations are issued for pilot actions.[7] Such a system improves safety by enhancing the pilot's situational awareness, and by reducing the cost and time required to achieve and to maintain pilot proficiency.[7] Figure 4.1 illustrates the modular layout of GAPATS and the interfaces between software components and hardware components.[7] GAPATS is composed of several independent modules, which includes Navigation Module, Flight Mode Interpreter, Pilot Advisor, Head-Down Display, etc. The interface and integration of these different modules is designed around a central data object that is used to coordinate the data communication between the different modules. This data object is illustrated in Figure 4.2. Each module has exclusive write access to the values it is responsible for updating, and all modules have read access to all of the information in the data structure.[7]

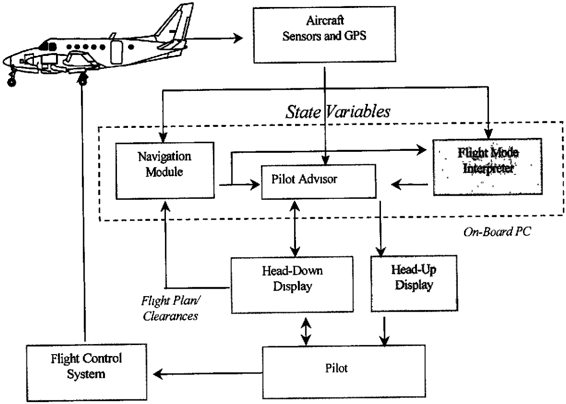


Figure 4.1 GAPATS Architecture

The agent system is designed to take as much advantage as possible of the existing resources and capabilities provided by GAPATS. The GAPATS Navigation Module and Head Down Display are the two features most used for the agent system. The Navigation Module provides all of the information required to perform basic flight planning to the destination, such as present position and direction, destination, ground speed, present time, etc. All of this capability is used by the agent system for detecting potential conflicts and computing conflict-free flight trajectories. The multi-function head down display (HDD) in GAPATS HDD is essential for proper functioning of the agent system, since its moving map can display radar images of squall lines, local traffic and collision warning messages, and the ultimate recommended conflict free flight path. As illustrated in Figure 4.3, it is composed of a flat-panel screen and 12 buttons. In addition to displaying agent system data and information, the HDD also provides check-lists, weight and balance, flight planning, training, and in-flight navigation displays.[7] Figure 4.4 shows a moving map with a weather radar image indicating a squall line ahead of the aircraft.

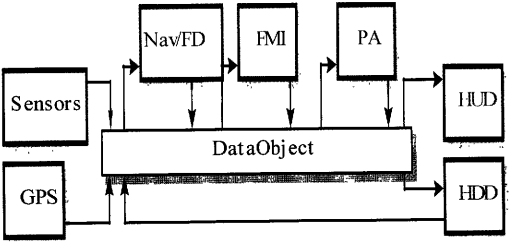


Figure 4.2 Data Object Interfaces

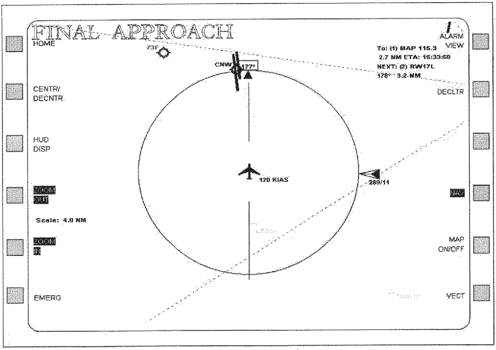


Figure 4.3 Multi-Function Display

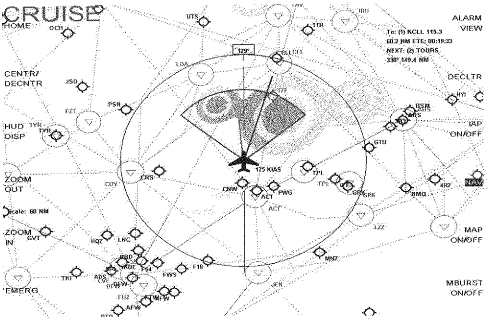


Figure 4.4 Moving Map with Weather Radar Image

4.3 Real-Time Engineering Flight Simulator

Development and evaluation of the agent system is being conducted in the Texas A&M University Flight Simulation Laboratory (FSL). The centerpiece of the laboratory is the real-time, nonlinear, six degree-of-freedom fixed base engineering flight simulator (EFS). It contains a cockpit with reconfigurable, multifunctional displays that can be rapidly modified and tailored to fit individual project needs for a wide range of general aviation, commercial, and military cockpit displays.[16] The EFS is currently capable of simulating three different aircraft: the Rockwell Commander 700; McDonnell-Douglas AV-8A Harrier; and Northrop F-5A Freedom Fighter.[16] Figure 4.5 depicts a pilot operating the EFS.

Figure 4.5 Engineering Flight Simulator Cockpit and Environment

Brief descriptions of the software and hardware resources of the EFS are presented in the following sections.

4.3.1 Cockpit

A refurbished T-37 cockpit serves as the cockpit of the EFS, with the tail section, existing wiring, wings, and hydraulic lines of the original cockpit removed (Figure 4.6). The cockpit includes both a traditional center force stick and a side-stick, facilitating the simulation of various types of aircraft.

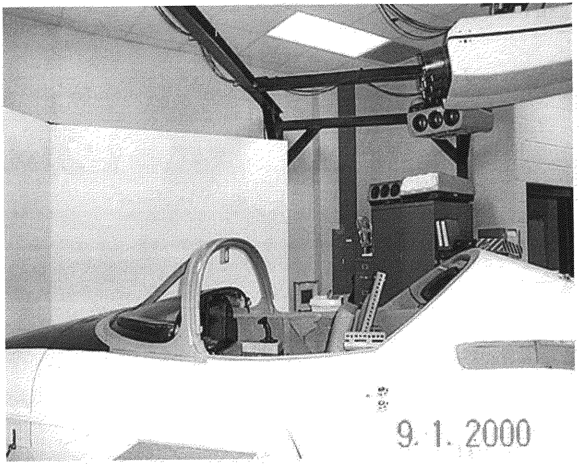


Figure 4.6 Engineering Flight Simulator Cockpit

4.3.2 Computers and Platforms

The computational engine of the EFS is a Silicon Graphics Onyx Reality II Graphics Workstation. It is a UNIX based system that has one R4400 processor chip with 12GB of hard disk space and 256 MB RAM. The Onyx Reality II handles all Equations Of Motion calculation and scene generation for the EFS. There are also three networked PCs in the lab that are responsible for driving the instrument displays, GPATS, head-down displays, and Soft Pilot/FMS Interface (SPiFI). Simulation synchronization across the multiple computers used for simulation in the EFS is accomplished using a local network (Figure 4.7).

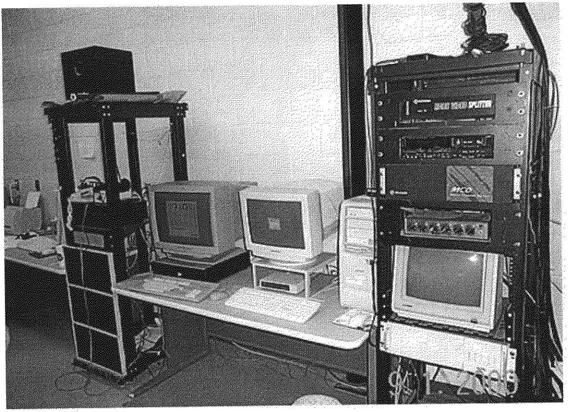


Figure 4.7 Computers in the Flight Simulation Laboratory

4.3.3 Engineering Flight Simulator Flight Visualization Environment

The EFS is capable of generating an approximately 150 degree field of view for the fixed base cockpit, utilizing three overhead rack-mounted color projectors, and the outputs from the Multi-Channel Option. Figure 4.8 shows the configuration of the flight visualization environment hardware used for the EFS. There are two computer monitors mounted inside the cockpit (Figure 4.9). On the left side is a 15" CRT used for displaying the GAPATS Head Down Display information, driven by one of the PCs running the GAPATS software. The right hand side display is a 15" LCD capacitive touchscreen for use with SPiFI and other types of HDD, such as aircraft instruments. It is driven by a PC running SPiFI software (to be described, below).

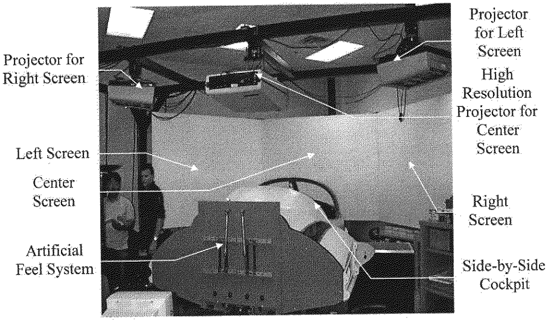


Figure 4.8 Configuration of the Engineering Flight Simulator Flight Visualization Environment Hardware

Figure 4.9 Left and Right Head Down Displays of the Engineering Flight Simulator Cockpit

4.4 Commander 700 Autopilot

The Commander 700 simulation model possesses a three axis autopilot (lateral axis, and directional axis) with gain scheduling according to three different flight phases: cruise, takeoff, and power approach.[16] The configuration for each flight phase, along with the related stability derivatives, were obtained from Engineering Flight Simulator System Description (Version 1.1) [Ref. 16]. An autopilot command generator determines appropriate inputs to the autopilot (step, ramp, etc) based on flight phase and configuration, according to desired actions of the pilot. The mode functionality of the autopilot is displayed in Table 4.1.

Table 4.1 Autopilot Mode Functionality

AUTOPILOT	**COMBINATIONS**
Pitch Axis Autopilot	Altitude Command And Hold Altitude Command And Hold With Pitch Damper Pitch Attitude Command And Hold Pitch Attitude Command And Hold With Pitch Damper Pitch Damper
Lateral Axis Autopilot	Heading Command And Hold Heading Command And Hold With Roll Damper Roll Damper
Directional Axis Autopilot	Yaw Damper Turn Coordinator With Yaw Damper

4.5 Soft Pilot/Flight Management System Interface (SPiFI)

SPiFI functions as the pilot interface to the Simplified Flight Management System (SFMS). The SFMS is avionics software developed in-house for the EFS. The modifier "Simplified" suggests that this version provides only the most basic coupled navigational functions for the autopilot.[16] SPiFI is designed and implemented using Centric Corporaton's Designer's Workbench (DWB), a complete 3-dimensional graphical modeling environment software packages. SPiFI is displayed on the right hand side head down display, where it is easily accessible from the co-pilot or navigator station (Figure 4.10). Manual of SPiFI will be soon added to the documentations of Engineering Flight Simulator.

Figure 4.10 Operation of the Soft Pilot Flight Management System Interface

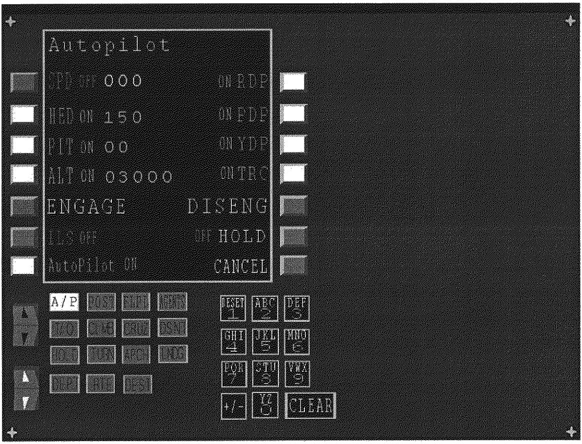
The GAPATS software module is interfaced to SPiFI within a data structure through the UDP protocol. The GAPATS PC is interfaced to the main simulation engine using the TCP/IP protocol.

4.6 User Interface

To satisfy the verification requirement of the current research, three primary user interface functionalities are required, each one is described in the subsections as follows.

4.6.1 Autopilot Input Interface

As shown in Figure 4.11, pilots may control the autopilot via the autopilot input interface page. Pilots may engage or disengage the all autopilot functions; input autopilot heading, pitch, and altitude commands; and engage or disengage individual dampers, and Instrument Landing System (ILS) and Holding pattern functions.

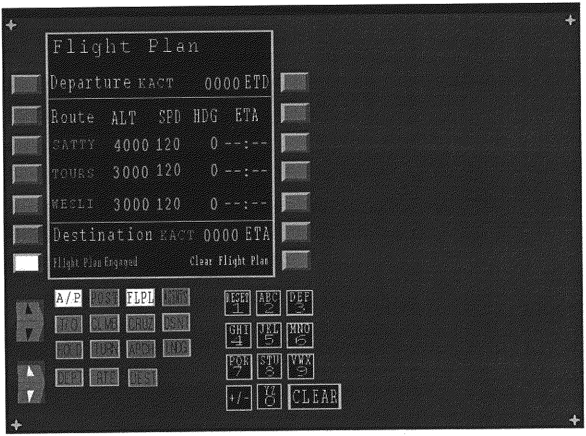

Autopilot

SPD OFF 000
HED ON 150
PIT ON 00
ALT ON 03000
ON RDP
ON PDP
ON YDP
ON TRC
ILS OFF
AutoPilot ON

Figure 4.11 Autopilot Input User Interface Page

4.6.2 Flight Planning Input Interface

As shown in Figure 4.12, pilots control the flight plan via the flight planning input interface page. During preflight, pilots input departure airport, Estimated Time of Departure (ETD), en-route waypoints, destination airport, and other flight plan information. During flight, pilots engage or disengage the flight plan, and modify the en-rout waypoints and the destination airport. Information such as names and locations of various waypoints, e.g. NDBs, VORs, intersections and airports, are provided by the Jeppesen navigation database.



Flight Plan

Departure KACT 0000 ETD

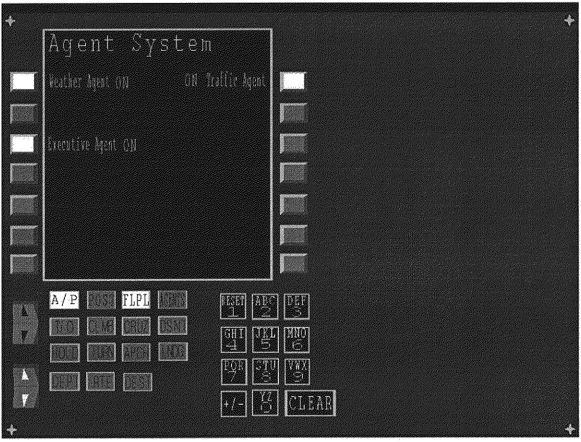
Route	ALT	SPD	HDG	ETA
SATTY	4000	120	0	--:--
TOURS	3000	120	0	--:--
WESLI	3000	120	0	--:--

Destination KACT 0000 ETA

Figure 4.12 Flight Plan Input User Interface Page

4.6.3 Agent System Control Interface

As shown in Figure 4.13, pilots control the agent system via agent system control interface page. Pilots enable/disable the agent system, and select individual weather, traffic and executive agents or combinations of them.



Agent System

Weather Agent ON

ON Traffic Agent

Executive Agent ON

Figure 4.13 Agent System Control User Interface Page

4.7 Integrated Flight Simulation and Agent System

Figure 4.14 shows the software architecture of the integrated flight simulation system consisting of the EFS and the GAPATS, SPiFI, and agent system. Since some of the required resources, such as the navigation module and the HDD moving map display, already exist in the GAPATS system, the agent system was built upon GAPATS. The combined GAPATS/Agents system functions as a simplified Flight Management System (FMS), with SPiFI serving as the pilot interface to it.

Figure 4.15 shows the software connectivity architecture of the current flight simulation system, in which the dashed lines indicate data flow. In its present form, the flight simulation system allows a pilot to operate the simulator in two modes—manual mode and autonomous mode. In manual mode, the pilot controls the aircraft using standard pilot controls and instruments. In autonomous mode, the pilot allows the autopilot to control the aircraft. The pilot enables the autopilot with a switch located on the autopilot page on SPiFI, and inputs autopilot commands and flight plan information using the menus. Input data is transferred to GAPATS through a UDP connection, and after processing by GAPATS, the input information is transferred to the autopilot command generator. The command generator determines appropriate inputs to the autopilot (step, ramp, etc) based on flight phase and configuration, according to the input commands of the pilot. The autopilot then flies the aircraft according to the commanded positions or next waypoint on the flight plan. The agent system is only used in autonomous mode in conjunction with the autopilot, which flies the aircraft along the final conflict-free flight path recommended by the agent system. Either of two types of input data is issued by the agent system to the autopilot: one is the next waypoint on the conflict-free flight path recommended by either the weather or the traffic agent, and the other is the commanded heading issued by the executive agent. In the first case, the tracker module in the autopilot computes the course towards the recommended waypoint and guides the aircraft along this course. In the second case, the commanded heading from the executive agent is sent directly to the command generator of the autopilot. Figure 4.16 illustrates the data flow between various modules as depicted above.

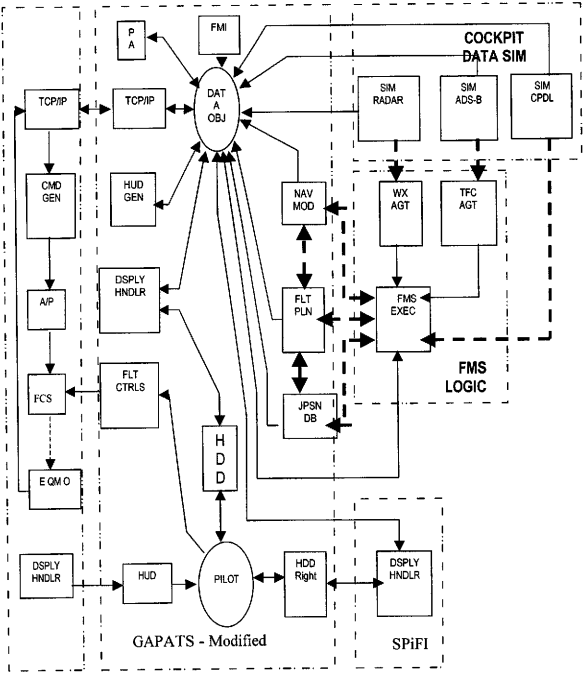


Figure 4.14 Integrated Flight Simulation Software Connectivity Architecture

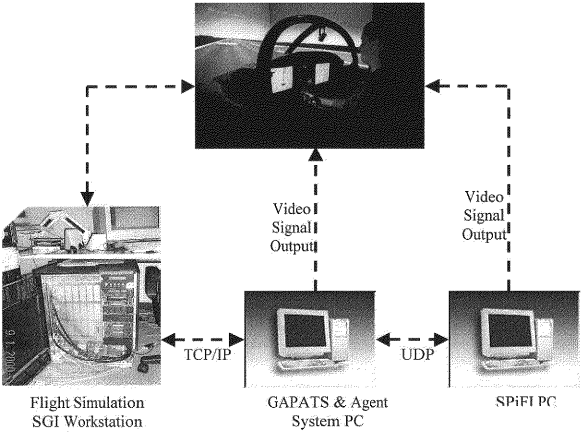


Figure 4.15 Flight Simulation System Computer Connection

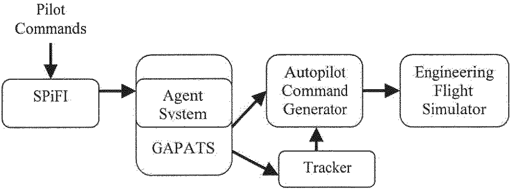


Figure 4.16 Pilot Command to Autopilot Data Flow

5 SOFTWARE IMPLEMENTATION

5.1 Introduction

This chapter details the software implementation of the agent system, and is divided into three parts the implementation of each individual agent, integration of the three individual agents into the agent system and integration of the agent system with existing GAPATS software. The agent system software is developed on three IBM-class personal computers in an object oriented programming environment

5.2 Implementing Agent System in C++

Borland C++ 5 was used to implement the agent system, since GAPATS is developed and written in Borland C++. C++ is a typical object-oriented programming language (OOP) which groups data, and operations on data, into modular units called *objects* and combines these objects into structured networks to form a complete program In an OOP language, objects and object interactions are the basic elements of design. An OOP language is an appropriate choice for a project being developed by a team of programmers. as the modularity forced upon individual programmers enhances software quality control Another advantage of an OOP language is that each programmer may conduct the localized code modifications within a module that may then be reflected across the whole application

5.3 Integration of Individual Agent Software within Agent System

Table 5.1 depicts the software and hardware configurations of these development tools.

Table 5.1 Hardware/Software Configuration of Agent system development PCs

Clock Rate	600 MHz (1 PC), 750MHz (1 PC), 800 (1 PC)
Memory	128 MB (1 PC), 256MB (2 PCs)
Hard Disk Drive Capacity	13.5GB(1 PC), 20.4GB (1 PC), 24.5 GB (1 PC)
Operating System	Windows 2000(1 PC); Windows NT (2 PCs)
Compiler/Debugger	Borland 5 C++ w/OWL (Object Window Library)/ ANSI/ISO Standard C++ Library
Video Card	All-In-Wonder video card with 32MB VRAM (1 PC) 32MB Viper 770 video card (2 PCs)
Monitor	ViewSonic PF790 (1 PC), ViewSonic 17 GS (1 PC), Micron 700 CX (1 PC)

The agent system software is composed of three agent modules, and three supporting modules (Figure 5.1). The *weather radar* module functions similarly to an on-board weather radar, providing simulated radar images of weather and also other weather information required by the weather agent. The *ADS-B system* module emulates the ADS-B device employed on many aircraft. The module receives and transmits simulated ADS-B messages and collects other necessary traffic data. The *AgDataObject* is a module that serves as the central depository for all data shared by the various modules in the agent system. To avoid forcing developers to write and modify parts of one large body of code, creation of an AgDataObject standardizes and isolates the data interfaces between the various functional modules The word, "object," refers to the underlying programming method used, which is Object-Oriented programming.[6] Functionally, the AgDataObject operates as a data communication switch, routing required data between the various modules and it helps facilitate a distributed, parallel implementation process.[6] The structure and functionality of the AgDataObject is similar to that of the DataObject used in GAPATS

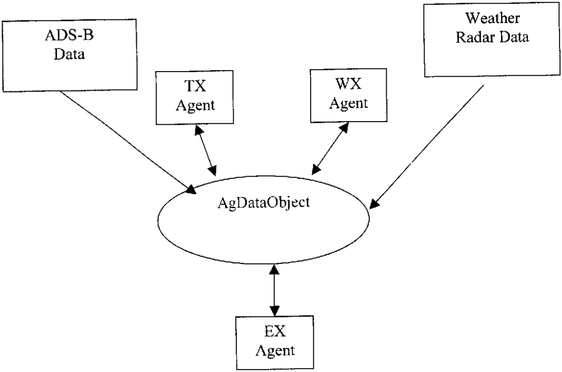


Figure 5.1 Data Flow within Agent System

The AgDataObject is implemented as a C++ class and contains several data members of following data types:

- StormIntensity: Defines thunder storm intensity, which is used to generate the simulated squall lines and weather conflicts for the weather agent.

- RadarData. Defines weather radar image data. which is used to generate simulated on-board weather radar image for the weather agent.

- WxAgentData: Defines output data from the weather agent, including the modified. weather conflict free flight path calculated by the weather agent

- AdsbData: Defines ADS-B data, which is used to generate simulated air traffic scenarios and traffic conflicts for the traffic agent.

- TxData: Defines output data from the traffic agent. including the modified, traffic conflict free flight path calculated by the traffic agent.

- ExData Defines output data from the executive agent. including the ultimate, conflict free flight path calculated by the executive agent.

- GapatsData Defines necessary information provided by GAPATS, including GPS data, raw flight data, and aircraft controllers configuration data.

The AgDataObject also contains several function members that may be divided into two categories: data assessors, and data depositors. The former is used to obtain the data stored in the AgDataObject, and the latter is used to update the data For example, the current StormIntensity data saved in AgDataObject is returned by calling the function *stormData()* and is renewed by calling the function *SetData(StormIntensity *data).*

5.4 Integration of Agent System Software with GAPAT

As described in Chapter 4, the agent system is integrated into GAPATS. Data communication between GAPATS and the agent system is realized through connection of DataObject and AgDataObject, illustrated in the Figure 5.2.

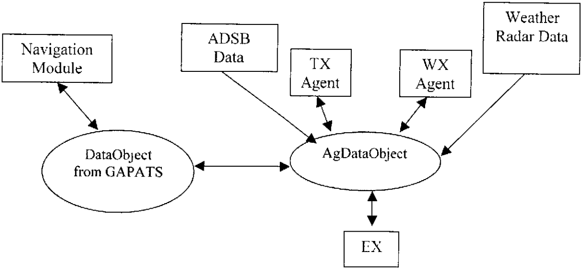


Figure 5.2 Data Flow between GAPATS and Agent System

There are several reasons for generating a new AgDataObject instead of directly connecting the individual agents to the existing DataObject. First, from a research point of the view, though closely related to and requiring considerable data from GAPATS, the agent system is designed and developed as an independent system rather than a supplementat to the existing GAPATS system. Second, from a software development point of view, considering the convenience of integration, creating a new AgDataObject is a better way than adding more data structures to the existing huge and complicated

DataOjbect and in addition, it may guarantee a clear and neat software structure. At last, constructing a separate AgDataObject is as well in the interest of the future maintenance and development of agent system

Three new push buttons are added to the current GAPATS executive window

- Weather Agent button
- Traffic Agent button
- Executive Agent button.

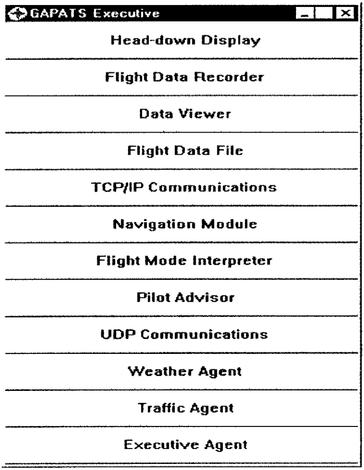


Figure 5.3 GAPATS Executive with Agent System

The modified GAPATS executive window is shown in Figure 5.3. Each of the three agents can be enabled or disabled by the user through the SPiFI interface. This functionality requires that GAPATS and SPiFI run simultaneously. However, during the software development stage of this research, enable/disable push buttons for each of the three agents were added to the GAPATS executive window so that SPiFI does not need to be running. This makes the software debugging process more convenient. The modified GAPATS executive window is shown in Figure 5.4.

For each agent, a simple timer algorithm is used to control its update rate. As and example, the timer for the weather agent is

```
if (wx_run)
        {
                wxagent->Update(&agdata);
                wx_run = 0;
                wx_start_t = time(NULL);
        }

        wx_current_t = time(NULL);
        if (difftime(wx_current_t,wx_start_t)>= WXRUNNINGRATE) wx_run = 1;
```

where if wx_run is equal to 1, then the weather agent runs and updates the AgDataobject, which is denoted "agdata" here. Once the update function of the weather agent is executing, wx_run is immediately set to 0, and the current time is recorded in variable wx_start_t. Then, in each subsequent cycle of GAPATS, the difference between the current time and the last execution time of the weather agent is checked. Note that wx_run is reset to 1 only when the the elapsed time exceeds the defined update rate of the weather agent, denoted by WXRUNNINGRATE here. Finally, after a certain specified period of time, the weather agent executes again, when wx_run becomes 1. The timers for the other two agents are similarly implemented and the users as required can easily set the update rate for each agent.

6 EXPERIMENT DESIGN

6.1 Introduction

This chapter develops the test plans for evaluation of the executive agent in the real-time engineering flight simulator. It also describes the constraints and considerations to be taken into account in implementation and evaluation of the executive agent

6.2 Multi-Agent Simulation System

The test cases designed for the evaluation of the executive agent should include both weather and traffic conflicts. Therefore, the simulated scenarios where one or several thunderstorms and aircraft conflicts concurrently exist must be generated The test cases for the executive agent should be realistic such that the weather-constrained airspace is taken into account in the scenarios. In the test case scenarios for evaluating the traffic agent, several simulated "bogey" aircraft are created and fly along their designated courses. In these scenarios. the aircraft must maintain their headings and airspeeds, because the traffic agent algorithm only permits the subject aircraft to change its course to avoid traffic conflicts.[3] However, when thunderstorms are added into the test cases, how the simulated bogey aircraft are supposed to behave corresponding to the existence of a squall line must be addressed. It makes no sense to permit these aircraft to maintain their original courses and ignore the existence of the squall lines while the subject aircraft may not. Thus, the bogey aircraft should also undertake certain intelligent and reasonable reactions to the thunderstorms, which means the simulated flight paths for these bogey aircraft should be constrained to accessible regions only.

A simple multi-agent simulation system (MASS) composed of all the simulated bogey aircraft is used to solve these problems Each simulated bogey aircraft is regarded as one agent of this MASS, which possesses a weather agent similar to that currently implemented in the hierarchical agent system. Using the MASS, reasonable test cases are created where all of the bogey aircraft have the capability to detect and resolve weather conflicts Consequently, the airspace between two thunderstorms in a squall line will become crowded with traffic as desired, as all of the bogey aircraft will be guided by the weather agent to pass between the thunderstorms. Therefore, in the airspace between two thunderstorms, the possibility of violating airborne safe separation increases, and traffic conflicts are more likely to occur In concept, usage of the MASS fulfills the purpose of providing realistic weather avoidance behaviors for evaluating the executive agent.

For reasonable test cases, not only weather conflicts but also safe separation assurance between any two of the bogey aircraft should be considered For simplicity, this aspect is not addressed in the current research. However, the MASS may be extended in future research, such that each bogey aircraft has the ability to solve traffic CD&R problems as well. Full development of the MASS is one of the primary tasks for future research

6.3 Test Matrix and Conditions

Different from the test cases for the weather and the traffic agent, which are developed both in Matlab and C++, those for the whole agent system are only written in Borland C++ and implemented directly in real-time simulator. A test case will be regarded as successful if none of the constraints in Section 6.3.2 is violated

6.3.1 Test Considerations

The primary objective of running the test cases is to examine the feasibility, effectiveness and efficiency of the agent system. Therefore, the first item to be examined of the test results is if the agent system provides an alternate route at all in case conflicts exist Next is whether the flight path suggested by the agent system endeavors to avoid both the weather and the traffic conflicts

6.3.2 Constraints

The new path must satisfy all of the constraints stated in Table 6.1. These constraints should be similar to or the sum of those used for evaluation of the independent weather and traffic agents. Considering weather conflicts. on a radar image. the yellow and red regions should be avoided, or in other words. the flight path of the aircraft should avoid regions of intensity greater than 30dBZ.[4] The minimal horizontal and vertical distances of safe separation between two aircraft are set according to the definition of the protected zone Considering the comfort of passengers and pilots. large, rapid and frequent heading changes are undesirable. However, in some extreme situations that contain both weather and traffic conflicts, ensuring the safety of the aircraft has a higher priority than ensuring passengers comfort. Therefore, the constraints such as number of turns, segment length, heading change limitation, etc. can be relaxed, if desired.

Table 6 1 Test Constraints

Constraints	**Values**
Thunderstorm Intensity to be avoided	> 30 dBZ
Number of Turns	As low as possible
Minimum horizontal distance of separation assurance	3 nautical miles
Minimum vertical distance of separation assurance	500 feet
Minimum distance from the inaccessible region	3 nautical miles

6 3 3 Update Rate for Each Agent

In the test cases for the weather agent, its update rate is set as no more than 20 seconds for moving squall lines.[4] However, 20 seconds update rate of the weather agent is unacceptable for the test cases of the agent system. As mentioned in the Chapter 3, the agent system employs a sequential conflict resolution process, in which the traffic agent performs with regard to the weather conflict-free flight path provided by the weather agent Rapid change of the new weather conflict-free path will cause the performance of the conflict detection and resolution module in the traffic agent unstable and inaccurate Therefore, the update rate of the weather agent is set as 1 minute in the case of the moving thunderstorms. The update rate of the traffic agent should be relatively small, so that it may handle the frequently changed traffic scenarios. It is set as 3 seconds in the test cases merely for the traffic agent,[3] and 10 seconds in the test cases for the whole agent system, which accounts for the existence of the executive agent. Acting as the arbitrator of the weather and traffic agents, the executive agent should update no less frequently than either of the above two agents, and its update rate is set as 5 seconds The one minute interval of the weather agent and the 10 seconds interval of the traffic agent give ample time for the executive agent to resolve the conflicts between the traffic agent and the weather agent, thereby facilitating integration with the other agents[4] The interval of 5 seconds of the executive agent provides sufficient time for it to complete the calculation of a new weather and traffic conflict free course.

6.3.4 Simulated Radar Images

The onboard radar for a general aviation aircraft has adjustable range and a horizontal scan angle of 100 degrees. As the update rate of the weather agent is set as 5 minutes in the agent system, in order to guarantee the detection and resolution of weather conflict in most cases, in the algorithm, the radar range is set at 60 nautical miles It is one half longer than the radar range of 40 nautical miles set in the test cases for the weather agent, alone.

6.3.5 Simulated Weather Condition and Traffic Scenarios

Appropriate simulated weather condition and traffic scenarios are created to fully examine the capacity of the agent system to solve the multiple conflicts. Initial positions of the thunderstorms and initial states of all the aircraft have been carefully selected so that all the aircraft may avoid the thunderstorms on the same side, and the subject aircraft may encounter the bogey aircraft near the thunderstorms. Section 6 3.6 details the selection of the test cases and creation of the simulated test scenarios.

6.3.6 Data and Information Inputs

Table 6.2 Data and Information Inputs for Agent System

Input Variable	**Units**
Subject Aircraft Current Position (latitude, longitude)	degrees
Subject Aircraft Current Heading	degrees
Subject Aircraft Current Vertical Airspeed	feet/sec
Subject Aircraft Current Horizontal Airspeed	Knots
Radar Data	dBZ
Previous Waypoint (latitude, longitude)	degrees
Current Waypoint (latitude, longitude)	degrees
ADS-B Messages (described in the following rows)	
Aircraft Identification	
Bogey Aircraft Current Position (latitude, longitude)	degrees
Bogey Aircraft Current Heading	degrees
Bogey Aircraft Current Vertical Airspeed	feet/sec
Bogey Aircraft Current Horizontal Airspeed	Knots

The inputs to the weather and the traffic agent may be found in Bokadia's thesis [Ref. 4] and Shandy's thesis [Ref. 3] respectively. The primary inputs to the executive agent are the new waypoints on the flight path that is recommended separately by the weather and traffic agents. Other inputs include the latest information on weather conditions, which is provided by a radar image generated by the onboard radar, the current traffic situation received by the onboard ADS-B receiver, and the current states of the aircraft. Table 6.2 describes completely all the external inputs required for the agent system, which excludes the internal inputs passed from one agent to another one.

6 3 7 Selection of test cases

In all test cases, the initial altitude and airspeed of aircraft are set as 3000ft and 160 knots In the first two cases, the agent system will be evaluated in situations where only weather or traffic conflicts exist In the third case, the aircraft will encounter one other "bogey" aircraft when all of the two endeavor to avoid a stationary single thunderstorm that cuts both two aircraft's original flight paths The initial flight plan of the subject aircraft is from KCLL (College Station, TX) to KACT (Waco, TX), while that of the bogey aircraft is from KACT to KCLL. In the forth case, there are a line of three thunderstorms moving at 20 knots and a bogey aircraft, in which the two aircraft have the same initial flight plans as in the third case and the squall line crosses two aircraft's original flight paths. The fifth case is similar to the forth one, except that the moving thunderstorms are at the speed of 30 knots. The sixth case is similar to the fifth one, except that there are two "bogey" aircraft. Table 6.3 summaries all the test cases for the evaluation of the agent system in the flight simulator

Table 6.3 Test Matrices for Engineering Flight Simulator

Test Case	Description
I	A line of three moving thunderstorms moving at the speed of 30 knots and crossing the original flight path
II	Four bogey aircraft
III	Single stationary thunderstorm & one bogey aircraft
IV	A line of three thunderstorms moving at the speed of 20 miles per hour, cutting the original flight path and one bogey aircraft
V	A line of three thunderstorms moving at the speed of 30 miles per hour, cutting the original flight path and one bogey aircraft
VI	A line of three thunderstorms moving at the speed of 30 miles per hour, crossing the original flight path and two bogey aircraft

7 EVALUATION OF THE AGENT SYSTEM

7.1 Introduction

In this chapter, some experiment scenarios designed for the evaluation will be depicted and then the author will describe and analyze the evaluation results

7.2 Results from Real-time Engineering Flight Simulator

The test cases for the evaluation of the agent system on the real-time EFS were discussed in Table 6.3 In all test cases, the aircraft is flying from KCLL (College Station) to KCNW (Waco) As it is impossible to show the whole flight here, in the following sections, copies of the left Head Down Display at some critical points in the flight illustrates the results for all of the cases and shows the performance of the agent system Note that in all the figures, the brown line shows the straight-line path from the previous waypoint (i e KCLL) of the aircraft to the current waypoint (i e KCNW). The alternate path generated by the agent system is shown in blue.

7 2 1 Case I – One Moving Squall Line

In this case, a moving squall line was located on the flight path between KCLL and KCNW, with the speed of 30 knots and the course of 220 degree. Figure 7 1 to Figure 7.4 show the position of aircraft and storm at different points of the flight, along with the flight path generated by the agent system.

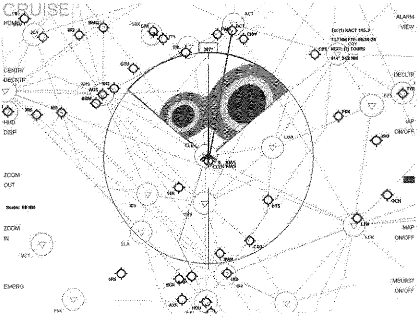


Figure 7.1 Case I – (a) Aircraft Flied along a New Path to Avoid Squall Line

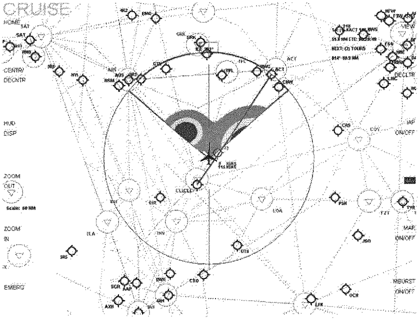


Figure 7.2 Case I – (b) Aircraft Flied between Two Thunderstorms

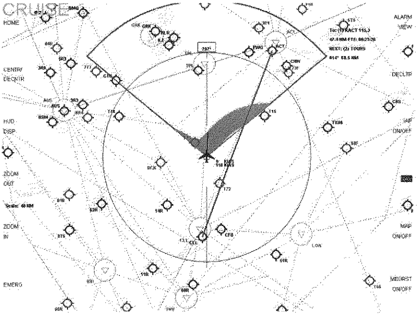


Figure 7.3 Case I – (c) Aircraft Almost Flied out of the Squall Line

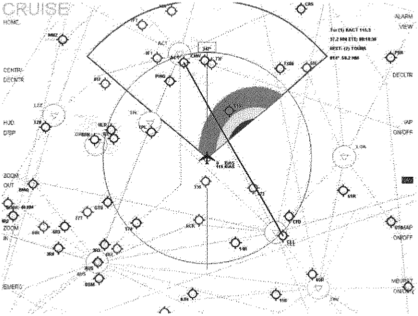


Figure 7.4 Case I – (d) Aircraft Returned to the Original Flight Path after the Weather Conflict Was Resolved

In this test case, the aircraft kept flying outside the forbidden zone (with storm intensity greater than 30 dBZ) of the squall line. The result for this test case is similar to that in Ref. 4 for only the weather agent.

7.2.2 Case II – Four Bogey Aircraft

In this case, the aircraft encountered four bogey aircraft, which Figure 7.5 to Figure 7.7 show the position of aircraft and storm at different points of the flight, along with the flight path generated by the agent system.

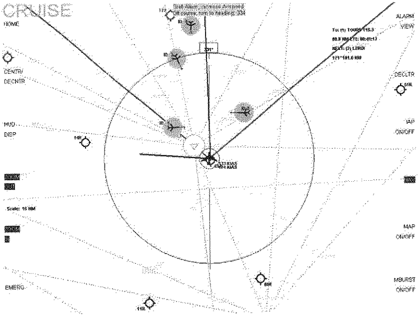


Figure 7.5 Case II – (a) Four Traffic Collisions at the Beginning

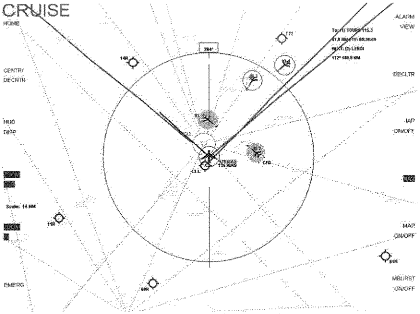


Figure 7.6 Case II – (b) Aircraft Flied to a New Path to Avoid All Traffic Collisions

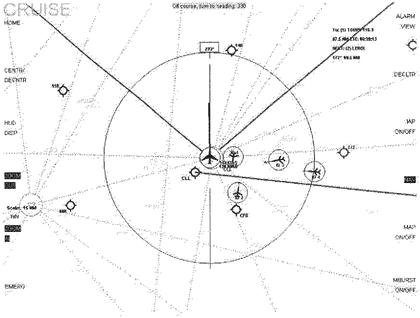


Figure 7.7 Case II – (c) All Traffic Collisions Were Resolved

In this test case, the aircraft kept flying outside the protected zones of the bogey aircraft. The result for this test case is similar to that in Ref. 3 for only the traffic agent.

7.2.3 Case III – One Stationary Thunderstorm and One Bogey Aircraft

In this case, a stationary thunderstorm was located on the flight path between KCLL and KCNW, the aircraft encountered one bogey aircraft that flied towards KCLL and endeavored to avoid the same thunderstorm. As the thunderstorm was on the left side of the aircraft, the agent system selected to avoid the coming bogey aircraft from the right side so that it would not be cornered by the thunderstorm and the bogey aircraft. Figure 7.8 to Figure 7.12 show the position of aircraft and storm at different points of the flight, along with the flight path generated by the agent system.

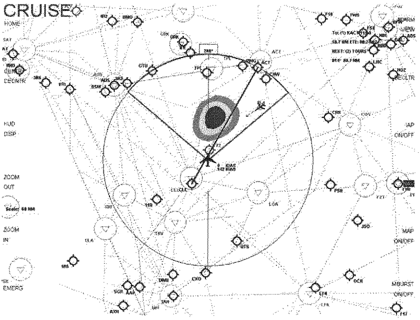


Figure 7.8 Case III – (a) Test Case Begun

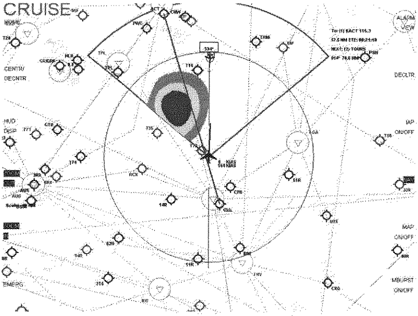


Figure 7.9 Case III – (b) Aircraft Flied to a New Path to Avoid the Thunderstorm

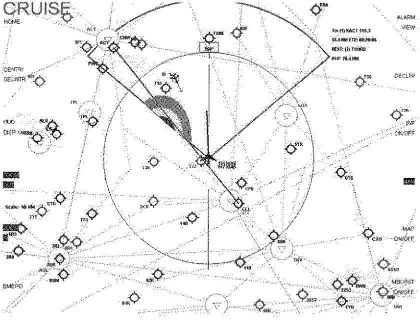


Figure 7.10 Case III – (c) Aircraft Deviated from the Weather Conflict Free Path to Avoid the New Traffic Threat

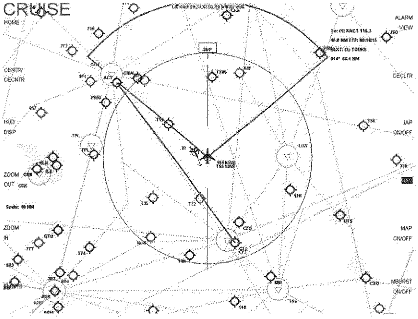


Figure 7.11 Case III – (d) Aircraft Approached to the Bogey Aircraft

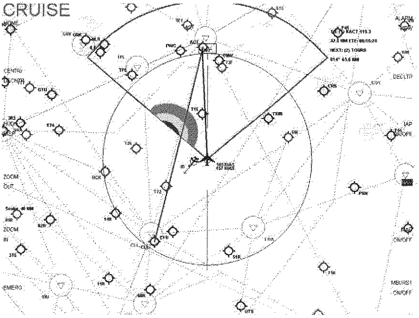
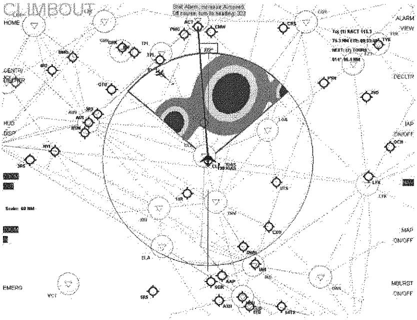


Figure 7.12 Case III – (e) Aircraft Again Flied along the Weather Conflict Free Path after the Traffic Conflict Was Resolved

7.2.4 Case IV – One Moving Squalline with Low Speed and One Bogey Aircraft

In this case, a moving squall line with three thunderstorms was located on the flight path between KCLL and KCNW, with the speed of 20 knots and the course of 220 degree. The aircraft encountered one bogey aircraft that flied towards KCLL and endeavored to fly between the two thunderstorms to avoid the same squall line. Figure 7.13 to Figure 7.17 show the position of aircraft and storm at different points of the flight, along with the flight path generated by the agent system.



Stall Alarm: Increase Airspeed.
Off course, turn to heading: 323

Figure 7.13 Case IV – (a) Test Case Begun

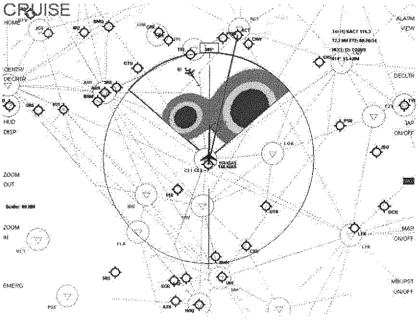


Figure 7.14 Case IV – (b) Aircraft Flied along a New Path to Avoid the Squall Line

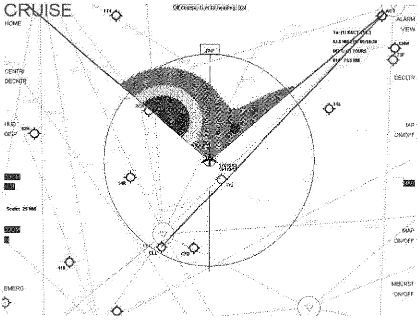


Figure 7.15 Case IV – (c) The Agent System Issued a New Flight Path to Avoid the Coming Bogey Aircraft and the Thunderstorm at the Same Time

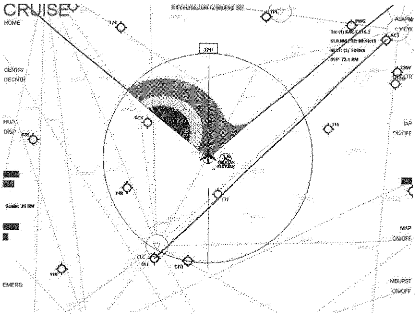


Figure 7.16 Case IV – (d) Aircraft Approached to the Bogey Aircraft

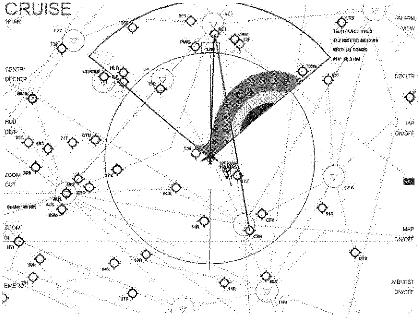


Figure 7.17 Case IV – (e) Aircraft Again Flied along the Weather Conflict Free Path after the Traffic Conflict Was Resolved

7.2.5 Case V – One Moving Squalline with High Speed and One Bogey Aircraft

In this case, a moving squall line with three thunderstorms was located on the flight path between KCLL and KCNW, with the speed of 30 knots and the course of 220 degree. The aircraft encountered one bogey aircraft that flied towards KCLL and tried to fly between the two thunderstorms to avoid the same squall line. At first, the agent system issued a flight path to avoid the bogey aircraft from the right side. However, as the thunderstorm was on the right side of the aircraft and it moved rapidly towards the aircraft, the agent system soon discovered that the original flight path would lead the aircraft into the forbidden zone of the aircraft. Therefore, the agent system issued another flight path to avoid the bogey aircraft from the left side, which, at the same time, reduced the risk of being trapped into the thunderstorm. Figure 7.18 to Figure 7.23 show the position of aircraft and storm at different points of the flight, along with the flight path generated by the agent system.

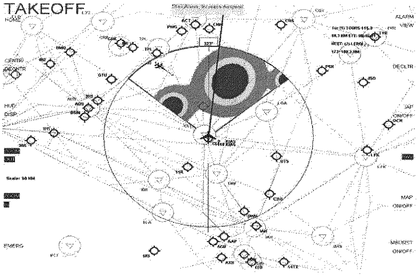


Figure 7.18 Case V – (a) Test Case Begun

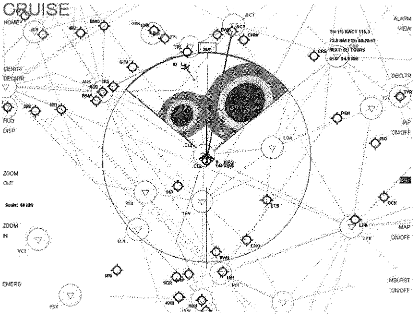


Figure 7.19 Case V – (b) Aircraft Flied along a New Path to Avoid the Squall Line

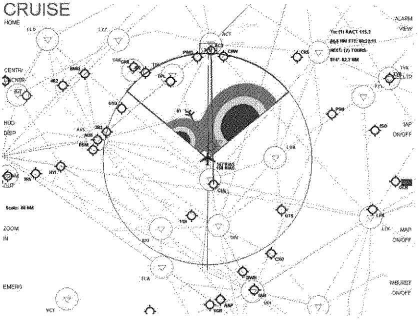


Figure 7.20 Case V – (c) The Agent System Issued a New Flight Path to Avoid the Coming Bogey Aircraft from the Right Side and the Thunderstorm at the Same Time

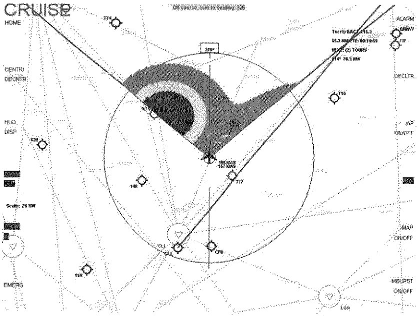


Figure 7.21 Case V – (d) the Agent System Issued a New Flight Path to Avoid the Coming Bogey Aircraft from the Left Side

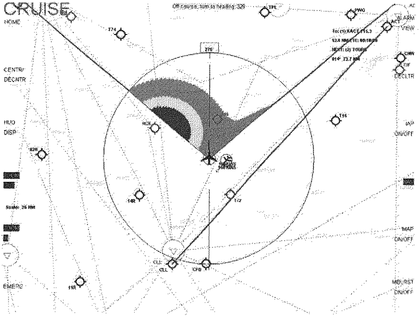


Figure 7.22 Case V – (e) Aircraft Approached to the Bogey Aircraft

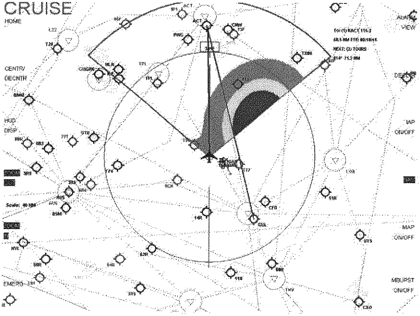


Figure 7.23 Case V – (f) Aircraft Again Flied along the Weather Conflict Free Path after the Traffic Conflict Was Resolved

7.2.6 Case VI – One Moving Squalline with High Speed and Two Bogey Aircraft

In this case, a moving squall line with three thunderstorms was located on the flight path between KCLL and KCNW, with the speed of 30 knots and the course of 220 degree. The aircraft encountered two bogey aircraft that flied towards KCLL and endeavored to fly between the two thunderstorms to avoid the same squall line. During the flight, the agent system changed its flight path to avoid the coming bogey aircraft from the right side to the left, due to the same reason described in the previous section. Figure 7.24 to Figure 7.29 show the position of aircraft and storm at different points of the flight, along with the flight path generated by the agent system.

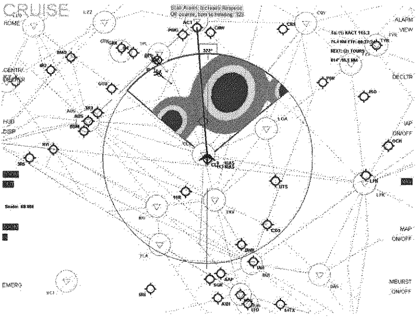


Figure 7.24 Case VI – (a) Test Case Begun

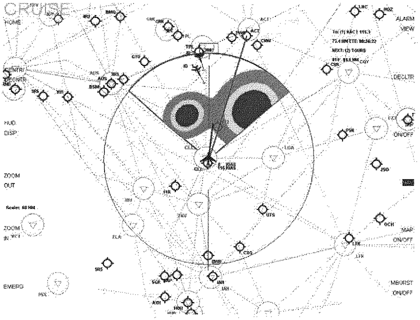


Figure 7.25 Case VI – (b) Aircraft Flied along a New Path to Avoid the Squall Line

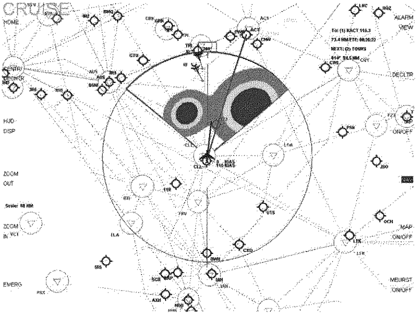


Figure 7.26 Case VI – (c) The Agent System Issued a New Flight Path to Avoid the Coming Bogey Aircraft from the Right Side and the Thunderstorm at the Same Time

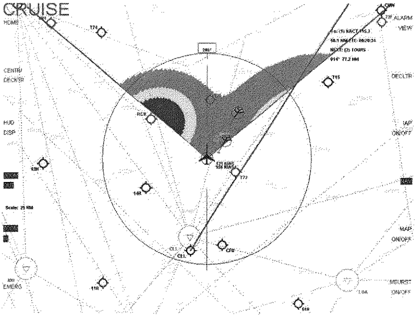


Figure 7.27 Case VI – (d) The Agent System Issued a New Flight Path to Avoid the Coming Bogey Aircraft from the Left Side

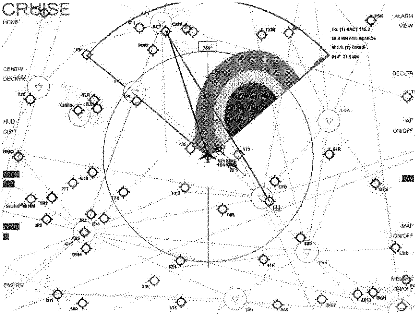


Figure 7.28 Case VI – (e) Aircraft Passed the Two Bogey Aircraft

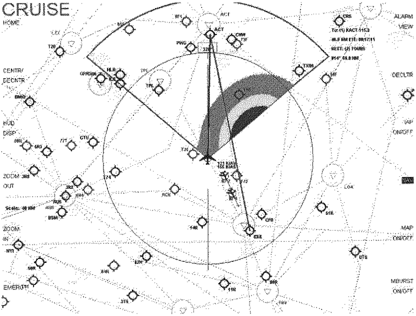


Figure 7.29 Case VI – (f) Aircraft Again Flied along the Weather Conflict Free Path after the Traffic Conflict Were Resolved

The flight path recommended by the agent system in all the test cases shown above satisfied the constraints listed in Table 6.1, which the aircraft avoided the forbidden areas of the squall line and protected zones of other aircraft. More test cases were run to testify the robustness of the agent system, which showed that the agent system failed to give resolution in one kind of extreme case. In this case, the aircraft was flying very closely to the dangerous area of the thunderstorms that are on its left side. Meanwhile, it was ready to pass a bogey aircraft that is on its right side. Without any expectation, the bogey aircraft made a sharp turn and flied towards the aircraft and the aircraft had to fly into the thunderstorm to avoid the bogey aircraft. In this case, both the traffic and weather conflict were violated at last. In the future research, the executive agent will be extended to a three dimensional agent. Therefore, in the extreme cases as described above, it may recommend a vertical maneuver to avoid the coming bogey aircraft while keep the aircraft out of the dangerous zone of the thunderstorm at the same time.

8 CONCLUSIONS

An agent based hierarchical system was developed for conflict detection and resolution in a Free Flight environment. An intelligent executive guidance agent was developed as a high-level arbitrator to resolve multiple traffic and weather conflicts from lower level weather and traffic agents. Within the whole aircraft system, the agent-based system actually plays the role of advisor or consultant rather than an executive. The flight path it recommends must be examined and verified by the flight management system and confirmed by the pilot, since he is ultimately responsibility for ensuring the airborne safe separation required in Free Flight. Test cases consisting of simultaneous weather and traffic conflicts, in addition to individual weather and traffic conflicts, were used to exercise the combined agent based hierarchical system. Based on the results presented in this Thesis, the following conclusions are drawn:

1. The concept of an intelligent agent based hierarchical system, composed of a high level executive intelligent agent, and lower level intelligent weather and traffic agents, has been shown to be an effective candidate for producing conflict-free flight path guidance in airspace subjected to co-existing individual weather and traffic conflicts.

2. Sequential prioritization of co-existing weather and traffic conflicts according to a pre-defined fuzzy rule-base can successfully resolve conflicts of different spatial and temporal types (tactical and strategic).

3. The executive intelligent agent demonstrated its capacity for detecting conflicts between the two lower-level agents, reasoning the severity and determining priority, and planning the resolution. These desirable characteristics are attributed to the fuzzy rule-based system approach used to develop the executive intelligent agent.

4. The overall agent based hierarchical system appears to be a promising candidate for the resolution of multiple co-existing weather and traffic conflicts. When integrated

with a simplified flight management system coupled with a heading command and hold autopilot, it can provide safe and reliable alternate flight paths in conditions of severe weather and multiple traffic conflicts.

9 RECOMMENDATIONS

The results and conclusions of this research indicate several areas in which the research can be extended:

1. **Trajectory Negotiation.** In the current implementation, only the subject aircraft possesses the agent system and the capacity to detect and resolve the traffic conflicts. In situations where one or more bogey aircraft also possess conflict detection and resolution capabilities, coordination and cooperation between all neighboring aircraft in the local airspace is required to reach mutually acceptable global resolution of conflicts. For the situation of an agent system in each aircraft, negotiation with other aircraft can be handled by either adding a new agent or enhancing the current capacity of the executive agent. In fact, several aircraft involved with one or more conflicts may themselves be regarded as a multi-agent system, in which each aircraft may be thought of as an independent agent. However, this multi-agent system is fundamentally different from the hierarchical agent system developed in the thesis, since all of the agents are on the same level. A negotiation protocol is needed in this case to search for a multilaterally acceptable solution.

2. **Multi-Agent Simulation System (MASS).** Further development of this research places a higher demand on validation of the method and experiment design. The Multi-Agent Simulation System introduced in Chapter 6 should be enhanced to create situations where the simulated bogey aircraft are more intelligent and reasonable, such that they have the same capacities of conflict detection and resolution as the subject aircraft. The ground air traffic controller, acting as high-level supervisor and coordinator, could be added into the Multi-Agent Simulation System as well.

3. **Soft Pilot/FMS Interface (SPiFI).** The Agent System Page on SPiFI and the Moving Map on the HDD should be further developed to facilitate the interaction between the agent system and the pilot. More information about the present conflicts could be displayed to increase the pilots' situational awareness, and the resolved flight path recommended by the agent system could be displayed in an appropriate way to aid a pilot's judgment and verification. Additionally, whenever a pilot desires to disable the agent system and fly manually, the agent system should be disable immediately and gracefully without distraction or an increase in workload.

4. **Weather Agent.** To satisfy the requirements of a sequential conflict resolution process, the weather conflicts addressed in the current research were a-priori assumed to be large scale and relatively slow moving. However, there exist many other types of weather phenomena which are small, local, and rapidly changeable such as microbusts and localized thunderstorms. These weather conflicts should be considered as tactical instead of strategic, and resolved in the second step of the sequential conflict resolution process. Therefore, instead of always resolving weather conflicts first and traffic conflicts second, conflicts should be categorized as either strategic or tactical. Besides some large-scale weather conditions, the strategic conflicts may also be used to handle Special Usage Airspace (SUA), while tactical conflicts could include the localized weather phenomena, traffic conflicts, and terrain conflicts.

REFERENCES

[1] Burdun, I., and Pardentyev, O., "AI Knowledge Model for Self-Organizing Conflict Prevention/Resolution in Close Free-Flight Air Space," *IEEE Aerospace Applications Conference Proceedings*, Vol. 2, 1999, pp. 409-428.

[2] Federal Aviation Administration, "FREE FLIGHT-Introduction" http://www.faa.gov/freeflight/ff_ov.htm, accessed in May, 2001

[3] Shandy, S. and Valasek, J., "Intelligent Agent for Aircraft Collision Avoidance", *AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, 6-9 August 2001.

[4] Bokadia, S., "Intelligent Weather Agent for Aircraft Severe Weather Avoidance", M.S. Thesis, Department of Aerospace Engineering, Texas A&M University, College Station, TX, May 2002.

[5] Painter, J., Glass, E., Economides, G., and Russell, P., "Knowledge-Based Processing for Aircraft Flight Control", *NASA Grand NAG1-1066*, NASA Langley Research Center, Hampton, VA, October 1994.

[6] T.H. Nguyen and D.T. Ward, "A Nerual-Network Based Inference Engine for a General Aviation Pilot Advisor", *35th Aerospace Sciences Meeting & Exhibit*, Reno, NV, January 6-10, 1997.

[7] Painter, J.H., Kelly, W.E., Trang, J.A., Lee, K.A., Branham, P.A., et al, "Decision Support for the General Aviation Pilot", *Proceedings of the 1997 IEEE International Conference on System, Man, and Cybernetics*, Orlando, FL, October 12-15, 1997.

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

[9] Bauser, P. Nouak, S. and Winkler, R. "A brief course in Fuzzy Logic and Fuzzy Control", http://www.flll.uni-linz.ac.at/pdw/fuzzy/fuzzy.html, accessed in August 2000.

[10] Kelly, W. III, "Dimensionality in Fuzzy Systems," Ph.D. Dissertation, Department of Electrical Engineering, Texas A&M University, August 1997.

[11] Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, 1995.

[12] Varaiya, P. Djaferis, T. and Schick, I. "A Question about Hierarchical Systems", *System Theory: Modeling, Analysis and Control*, Kluwer, New York, NY, 2000.

[13] Tomlin, C., Pappas, G., and Sastry, S., "Noncooperative Conflict Resolution [air traffic management]," *Proceedings of the 36th IEEE Conference on Decision and Control*, Vol. 2, 1997, pp. 1816 –1821.

[14] Lass, S.L., "An Object –Oriented, Knowledge-based Aircraft Approach Controller," M.S. Thesis, Department of Electrical Engineering, Texas A&M University, May, 1995.

[15] Ross, T., *Fuzzy Logic with Engineering Application*, McGraw-Hill, Inc., Columbus, OH, 1995, pp. 315.

[16] Engineering Flight Simulator System Description (Version 1.1), Aerospace Engineering Department, Texas A&M University, 2000.

VITA

Jie Rong received his B.S. degree in Mechanics and Engineering Science from Peking University, Beijing, P. R. China in July 1999. His graduate work has concentrated on the application of artificial intelligence methods to air traffic control problems. He will receive his M.S. Degree in Aerospace Engineering from Texas A&M University in August 2002 and will continue for his Ph.D Degree in Aerospace Engineering, Texas A&M University.

Jie's current mailing address is 1 Hensel Dr. T3E, College Station, TX, 77840.