AN ABSTRACT OF THE THESIS OF

Joachim C. Zaspel for the degree of Master of Science in Industrial Engineering
presented on July, 16, 1997. Title: Automating Pilot Function Performance Assessment Using Fuzzy Systems and a Genetic Algorithm.

Redacted for Privacy

Abstract approved: ______________________

Kenneth H. Funk

Modern civil commercial transport aircraft provide the means for the safest of all forms of transportation. While advanced computer technology ranging from flight management computers to warning and alerting devices contributed to flight safety significantly, it is undisputed that the flightcrew represents the most frequent primary cause factor in airline accidents. From a system perspective, machine actors such as the autopilot and human actors (the flightcrew) try to achieve goals (desired states of the aircraft). The set of activities to achieve a goal is called a function. In modern flightdecks both machine actors and human actors perform functions. Recent accident studies suggest that deficiencies in the flightcrew's ability to monitor how well either machines or themselves perform a function are a factor in many accidents and incidents. As humans are inherently bad monitors, this study proposes a method to automatically assess the status of a function in order to increase flight safety as part of an intelligent pilot aid, called the AgendaManager. The method was implemented for the *capture altitude* function: seeking to attain and maintain a target altitude. Fuzzy systems were used to compute outputs indicating how well the *capture altitude* function was performed from inputs describing the state of the aircraft. In order to conform to human expert assessments, the fuzzy systems were trained using a genetic algorithm (GA) whose objective was to minimize the discrepancy between system outputs and human expert assessments based on 72 scenarios. The resulting systems were validated by analyzing how well they conformed to new data drawn from another 32 scenarios. The results of the study indicated that even though the training procedure facilitated by the GA was able to improve conformance to human expert assessments, overall the systems performed too

poorly to be deployed in a real environment. Nevertheless, experience and insights gained from the study will be valuable in the development of future automated systems to perform function assessment.

Automating Pilot Function Performance Assessment Using Fuzzy Systems and a Genetic Algorithm

by

Joachim C. Zaspel

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented July 16, 1997
Commencement June, 1998

Master of Science thesis of Joachim C. Zaspel presented on July 16, 1997

APPROVED:

Redacted for Privacy

Major Professor, representing Industrial Engineering

Redacted for Privacy

Head of Department of Industrial and Manufacturing Engineering

Redacted for Privacy

Dean of Graduate School

I understand that my thesis will be become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Joachim C. Zaspel, Author

Acknowledgments

I would like to express my deep gratitude to Dr. Ken Funk for serving as my major professor. Without his generous help, advice, and encouragement this thesis would have not been possible. I am grateful to have participated in a NASA-funded research effort allowing me to learn so much about aviation and conducting this study. In particular, I would like to thank our technical monitor Kevin Corker and his assistant Greg Pisanich from the Ames Research Center for their support. I wish to thank Rolf Braune for sharing his expertise in aviation. Also, I would like to thank Dr. Edward McDowell and Vetle Torvik for advising me regarding questions about statistical methods. Thanks go to Robert Wilson and Woo Chang Cha with whom I had a great time working. Thanks are also due to the volunteers who participated in this study. I thank Drs. Eldon Olsen, James Nielsen, and Goran Jovanovic for serving on my committee.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF APPENDIX TABLES

LIST OF ABBREVIATIONS

ACTS	Advanced Civil Transport Simulator
ADI	Attitude Director Indicator
ATC	Air Traffic Control
CRT	Cathode Ray Tube
FAA	Federal Aviation Administration
FFA	Fuzzy Function Assessment
FHEIS	Fuzzy Hazard and Error Integration System
FOHS	Fuzzy Overspeed Hazard System
FSHS	Fuzzy Stall Hazard System
FTHS	Fuzzy Terrain Hazard System
GA	Genetic Algorithm
GPWS	Ground Proximity Warning System
MCP	Mode Control Panel
MSE	Mean of Sum of Squared Error
NTSB	National Transportation Safety Board
SHS	Selected Hazard Systems
TCAS	Traffic Collision Avoidance System
WSAS	Wind Shear Advisory System

Automating Pilot Function Performance Assessment Using Fuzzy Systems and a Genetic Algorithm

1. Introduction

Flightdecks of modern civil commercial transport aircraft are becoming more sophisticated as advanced computer equipment is installed to help pilots perform their jobs better. It appears that the cockpit of the future will resemble an office designed for "information management, communication, decision making, and supervisory control" (Wiener, 1988). In order to complete a flight mission, pilots have to achieve a number of goals in each phase of the flight ranging from attaining a target altitude to configuring the airplane for landing. While in the early days of aviation, pilots had to "manually" perform tasks to achieve their goals; today's digital on-board systems can be programmed to automatically achieve them. From a system perspective, the process of achieving a goal is called a function. So on the flightdeck, machine actors, such as the autopilot, and human actors (the flightcrew) are performing functions to accomplish goals.

While air travel is the safest of all forms of transportation (Bureau of Transportation Statistics, 1995), accidents still do occur. According to a report released by Boeing (1996), 59.8% of hull loss accidents were caused by the flightcrew while only 12.3% of the accidents were attributed to the airplane during the last ten years. Recent studies show that human deficiencies in assessing function status were a factor in past incidents and accidents (Chou et al., 1996). Why is it so important to assess how well functions are being performed? Firstly, pilots make mistakes, meaning that they may poorly perform the function manually, or may fail to program the automation correctly to perform the function for them. Secondly, the automation may fail to perform the function it is programmed for. In both cases, the problem can be recognized before it matures if one monitors if a function pursues a stated goal. So far, the flightcrew is called upon to perform this monitoring task. The dilemma is that humans are bad monitors and may fail to detect deteriorated functions. For instance, there are reported incidents and accidents in which the flightcrew failed to capture a target altitude (Chou, 1994). The reasons for failure varied from excessive workload to distraction and improper understanding of flightdeck automation. Whatever the reasons were, in most cases the flightcrew failed to respond to signs that if interpreted correctly would have indicated to the flightcrew that a function was not performed correctly.

As part of the AgendaManager, a computational aid to help pilots manage their flightdeck activities, it was sought to develop a system to automatically perform function assessment with regards to the *capture altitude* function, a function that seeks to attain and maintain a target altitude. The problem was how to devise such a system.

The objectives of this thesis were to implement function assessment for the *capture altitude* function, calibrate the implemented method to emulate human expert assessment, and validate if the method was reliable and coincided with human expert assessments. To facilitate function assessment, fuzzy systems were designed to assess the status of *capture altitude* functions. Subsequently, these systems were trained to emulate human expert assessment by the application of a genetic algorithm minimizing the discrepancy between fuzzy system and human expert assessments based on 72 scenarios. I validated the fuzzy systems by analyzing the discrepancies between them and the human experts based on another set of 32 scenarios.

The thesis is organized as follows: chapter 2 provides background information on automation in aviation technology and intelligent pilot aides. Chapter 3 lists the objectives of this thesis. Chapter 4 describes a fuzzy logic framework and computer-based fuzzy logic tools that were used in this study to create fuzzy systems facilitating function assessment. Chapter 5 talks about a genetic algorithm to train fuzzy systems. Chapter 6 details the fuzzy systems that were created to automatically perform function assessment and explains how human expert knowledge was extracted to train the fuzzy systems to conform to human assessment. Chapter 7 presents the results indicating how well the fuzzy systems conformed to human function assessment. Chapter 8 contains the conclusions and recommendations drawn from this study.

2. Background

In the coming millennium flightdeck automation is likely to change operations in the cockpit of modern civil transport aircraft and in ground based air traffic control facilities. Better ways of storing, retrieving, processing, and visualizing information and new airline and air traffic control policies and procedures as well as modified regulations by the Federal Aviation Administration (FAA) are called for as the number of commercial airplanes is estimated to triple within the next 20 years (Pélegrin, 1992). As fuzzy function assessment (FFA) described in this thesis represents a form of flightdeck automation, an overview of flightdeck automation related issues is provided here.

This chapter is organized as follows: First, various issues regarding flightdeck automation are overviewed. In particular, perceived flightdeck automation benefits and problems are summarized. Second, existing experimental intelligent pilot aid systems as a response to current automation problems are mentioned. Third, FFA for the *capture altitude* function is introduced.

2.1 Flightdeck Automation

There are many possible definitions of automation. Essentially, automation is the "automatic operation or control of a process, equipment, or a system" (American Heritage Dictionary, 1969). According to Billings, human operators use automation to "accomplish some task that would otherwise be more difficult or impossible, or ... would otherwise require increased human attention or effort" (1996, p 3). Billings divides flightdeck automation into three groups: automation of control, information, and management (1996, p 18). While control automation is concerned with controlling the aircraft in space and time, information automation handles information to be presented to the flightcrew. Management automation performs strategic planning and controlling tasks. Recently, computer-based pilot aids combine all or some forms of automation in an integrated fashion and often include AI algorithms to produce better solutions. Supplied with vast data sources from digital systems, artificial intelligence applications become feasible and are candidate systems for solving problems relating to pilot workload, situational awareness, and in flight emergency handling (Lee & Sanders, 1993). In section 2.2 those "intelligent" forms of automation are discussed in more detail.

2.1.1 Evolution of Flightdeck Automation

While early civil commercial jet airplanes such as the Boeing 707, introduced in 1958, enabled air travel at high altitude and speed, the systems the pilot operated with were simple compared with today's modern aircraft and many tasks such as navigation had to be performed manually (Billings, 1996). In those first generation jet aircraft, sensors provided pilots with essential state data such as magnetic heading, speed, altitude, and position while an analog autopilot was capable of controlling pitch, roll, and yaw as well as heading and altitude (Billings, 1996). Second generation aircraft such as the Lockheed L-1011 were equipped with more sophisticated analog autopilot offering a broader range of lateral and vertical aircraft control options and also provided autoland capabilities (Billings, 1996). Third generation aircraft such as the Airbus A-310 were equipped with digital computer systems and graphic cathode ray tube (CRT) displays, the trademark of so-called glass cockpits (Sweet, 1995). The emergence of CRT displays allowed designers to reduce the total number of displays on the flightdeck (Sexton, 1988). Also, flight management systems appeared allowing the pilot to program the airplane to follow a 4-D path defined as a sequence of longitudes, latitudes, altitudes and airspeeds (Billings, 1996). More information was made available to the crew; and pilots especially appreciated display formats allowing data integration such as weather pattern maps superimposed on graphics representing intended or alternative flight paths (Sweet, 1995). Fourth generation aircraft such as the A320 took automation a step further to a process called "envelope protection" where pilot commands are first processed by a digital "fly-by-wire" avionics system making sure that flight operating limits and boundaries are not exceeded by the pilot's control inputs (Billings, 1996).

2.1.2 Flightdeck Automation Benefits

Facilitated by rapidly improving micro-processor technology, the trend of using better computer equipment and software on the flightdeck is likely to continue. Advanced computer systems are deployed in the flightdeck in order to improve economy, customer satisfaction, and flight safety (Billings 1996; Wiener, 1989; Wiener & Curry 1980).

2.1.2.1 Economy

Today, the usage of flight management computers saves millions of dollars due to their effect of reducing fuel consumption significantly. In addition, wear and tear of tires, brakes, and engines is reduced by application of autothrottle and autobrake (Wiener, 1985b). Money is also saved by reducing the flightcrew size. With the certification of two pilot transport aircraft came a productivity boost as the flight engineer was substituted by automated systems (Wiener, 1985b). Digital systems are easy to maintain and are very reliable; in fact the fraction of time a scheduled aircraft is available for a trip is higher for more advanced aircraft than for conventional ones (Wiener, 1985b). For the airline industry the goal is "to conduct the flight as economical as possible, minimizing flight time, ground delays, fuel consumption, and wear on the equipment" (Wiener & Curry, 1980, p. 1005). The cost savings potential are likely to grow. While airspace is currently managed by Air Traffic Control (ATC) ordering aircraft to follow dedicated routes and airways, in the future, a less stringent way of managing the airspace, called "free flight", may give aircraft equipped with more powerful computer systems the opportunity to use more efficient flight paths (Billing, 1996, Sweetman, 1995). Computer systems also help utilize scarce airspace more efficiently in times of increasing air traffic (Wiener, 1985b). For instance, the FAA and NASA investigate how automation may be used to increase aircraft throughput by enabling closely-spaced parallel approaches in terminal areas (Billings, 1996).

2.1.2.2 Customer Satisfaction

In order to satisfy their customer needs, airlines have to tightly adhere to promised schedules and cater to passenger comfort. Thus, flight missions ought to progress in a timely manner and at an adequate level of comfort. Wiener and Curry describe passenger comfort as "to provide passengers with the smoothest possible flight (by weather avoidance, selection of the least turbulent altitudes, gradual turns and pitch changes, gradual altitude changes)" (1980, p. 1004). Computer systems become indispensable in devising passenger-friendly flight plans. In fact, the newest airplanes are equipped with gust alleviation algorithms and in the future "free flight" will allow pilots to select more comfortable flight paths (Billings, 1996).

2.1.2.3 Flight Safety

At least 60% of all aircraft accidents are attributed to human error (Billings, 1996; Nagel, 1988). Advanced information technology is likely to be the tool to make air-based transportation safer as evidence exists "that more highly automated aircraft have had substantially less accidents than earlier aircraft" (Billings, 1996). Various computer systems are deployed on today's modern fly-by-wire aircraft to enhance flight safety. There are automated systems such as autoflight systems that accurately fly the plane as commanded and warning devices that alert the flightcrew of imminent dangers such as system faults, stall and overspeed conditions, and environmental threats such as terrain, aircraft, and wind shear. The Ground Proximity Warning System (GPWS) detects terrain hazards and suggests climbing to a safe altitude to remove the hazard. Similarly, the traffic alert and collision avoidance system (TCAS) tracks dangerously approaching aircraft and proposes descending or climbing to alleviate the problem. The wind shear advisory system (WSAS) recognizes hazardous wind shear and suggests climbing to a safe altitude. GPWS and TCAS are proven to have increased safety (Billings, 1996). Those systems are effective because they alert the crew to hazards that they otherwise would not be aware of. Less obvious but nonetheless important is the improvement in the human-machine interface. Software generated displays (soft-displays) are very flexible and allow a great deal of data integration by visualizing data using symbols, colors, text, and multi-dimensional graphics (Wiener, 1985b). While graphics are now being used for map displays and system synoptics displays, the full potential of graphics is not utilized yet as displays visually integrating information regarding attitude and flight path in 3-D are still under development (Billings, 1996).

2.1.3 Flightdeck Automation Challenges and Problems

Accident and incident reports covering a series of fatal crashes as well as major flight abnormalities of modern civil transport aircraft show evidence that there are problems related to usage of advanced computer equipment on the flightdeck and suggest that current systems should be improved to enhance flight safety (Funk, Lyall, and Riley, 1995; Billings, 1996). Major problem areas identified so far are presented in the following section. The first problem area is related to adverse effects of flightdeck automation. The second addresses the issue of human limitations. The last points out the need for better human-machine interfaces.

2.1.3.1 Adverse Effects of Flightdeck Automation

While flightdeck automation overall contributes to safety, evidence exists that in a number of incidents automation actually was the cause of the problem or a contributing factor. As Billings summarizes:

> In some cases, automated configuration warning devices have failed or been rendered inoperative and flightcrew procedures have failed to detect by independent means an unsafe configuration for takeoff. In other cases, automation has operated in accordance with its design specifications, but in a mode incompatible with safe flight under particular circumstances. In still others, automation has not warned, or flightcrews have not detected, that the automation was operating at its limits, or was operating unreliably, or was being used beyond its limits. Finally, we have seen incidents and a few accidents in which pilots have simply not understood what automation was doing, or why, or what it was going to do next. (1996, p. 2-3).

The following sections explain problems with flightdeck automation more specifically.

2.1.3.1.1 Imperfect Automation

The programs, computers, sensors and digital devices of today's avionics systems are not error free. In fact, as the total volume of software implemented in the A340 exceeds 20 megabytes, error-free software becomes utopia. So, despite their high reliability, computer systems are prone to fail sometimes. The problem is that a failure in the computer system often remains unnoticed by the human operator until the situation turns bad (Wiener & Curry, 1980). Also, some warning devices such as the Ground Proximity Warning System (GPWS) have the tendency to give frequent false alarms. The effectiveness of those warning devices is reduced since "people will ignore an alarm if experience has shown that the alarm may be false" (Wiener & Curry, 1980, p. 1003).

2.1.3.1.2 Vulnerable Automation

Automation allows the flightcrew to make mistakes that were not possible without it and the consequences of automation induced errors tend to be severe (Kantowitz & Campbell, 1996). For instance, many errors with programming the inertial navigation system (INS) are documented (Wiener, 1980, Wiener 1988). Systems like the INS are particularly vulnerable to data entry errors. In several cases flightcrews entered an erroneous initial position or waypoint into the INS resulting in a major deviation from the intended flight path as the time displacement between error and noticeable consequences is fairly large (Wiener, 1985; Nagel, 1988).

2.1.3.1.3 Clumsy Automation

While automation is installed to decrease workload, Wiener suggests that although automation reduces workload in relaxed phases of flight such as cruise, it actually increases workload in busy phases such as final approach (1983, 1985a, 1985b). As the pilot has to monitor or scan the automation periodically, the potential for automation to actually increase workload is explained by its constant demand for attention. With the advent of automation such as the flight management system "pilots complain about more programming, planning, sequencing, and alternative selection, and more thinking" (Wiener 1985b).

2.1.3.1.4 Opaque Automation

A study performed by Sarter and Woods showed that pilots fail to understand the behavior of advanced automation under certain circumstances and revealed that the human-machine interface falls short of providing feedback as to what the automation is doing and why it is doing it (1994). The devices are opaque in that they operate silently and lack transparency. The focus of the researchers' criticism is the "proliferation of modes" allowing the autoflight system in combination with the flight management system to provide the flightcrew with a large number of functions and options for achieving their goals under different circumstances (1995, p. 17). The downside of mode rich systems is that "a human user can commit an erroneous action by executing an intention in a way that is appropriate to one mode when the device is actually in another mode." (1995, p. 6). Since advanced automation is gradually taking on the role of a semi-autonomous actor, transparency and predictability of the automation is more and more important. Lack of automation transparency may cause goal conflicts between flightcrew and machine as happened at Nagoya, Japan, in April 1994, when a Taiwanese A300-600 was programmed to perform go-around, a flight maneuver in which the airplane is commanded to climb and accelerate, while the pilot tried to land the airplane. The airplane got out of trim and crashed.

2.1.3.2 Human Limitations

Wiener concludes that the human is an unreliable monitor (1985a). Parasuraman found that "monitoring computers to make sure they are doing their job properly can be as burdensome as doing the same job manually, and can impose considerable mental workload on the human operator" (1996). In light of these findings it is surprising to see that the pilot's role has changed from a continuous in-the-loop controller to a system supervisor and monitor. The large number of warning and alerting systems installed in today's jets indicates that backup systems are necessary in case the flightcrew fails to perform its monitoring task correctly. As there are fewer and fewer meaningful functions to be performed manually, problems related to loss of flying skills, complacency, and boredom become an issue (Wiener, 1983, 1985a, 1988). Humans have limited short-term memories; thus, maintaining a correct mental model of the complex automation becomes a formidable task (Wickens and Flach, 1988). Today's flightdeck automation fails to address human limitations in many aspects. Billings proposes "human-centered aviation automation" tailored to human needs (1996). Other researchers argue that adaptive automation allowing the modification of the level of automation in real time may "represent an optimal coupling of the level of automation to the level of operator workload" (Scerpo, 1996).

2.1.3.3 Human-machine Interface

The human-machine interface provides the flightcrew with information regarding the state of the airplane and systems. According to Wiener, most cockpits overload the flightcrew with data (Wiener, 1983); partly because "most warning and alerting systems have grown up piecemeal in the cockpit, often being added one-by-one as the result of accidents" (Wiener, 1989, p 4). In addition, automated devices such as the flight management system are not very transparent to the human, causing the flightcrew to lack situational awareness about the status of the automatic device (Wiener, 1988). There is still room for improving the human-machine interface: existing information can be better integrated and more meaningful information about the state of the automation is needed.

2.2 Intelligent Pilot Aids

Faced with the deficiencies of current cockpit automation researchers go new ways. Promising is the area of intelligent assistant systems (Boy, 1991). Such systems, when configured as pilot aids, integrate functions such as monitoring, assessing, planning, and plan execution into one cohesive unit. This section reviews experimental pilot aids and the functions they perform.

Woods (1991) distinguishes between intelligent advisory, subordinate, and information systems. An important dimension of intelligent systems is their style of collaboration with the human operator and their level of autonomy. Autonomous systems operate on an underlying process silently without much interaction with the operator, whereas purely dependent systems are not authorized to manipulate the monitored process at all. According to Woods, advisory systems are dependent on the human as problem holder, while in contrast, intelligent subordinate systems, supervised by the human operator, act on the monitored process autonomously (1991). Information systems are designed to facilitate the information exchange between the human and the machine and provide better information for the human problem solver. The pilot aids reviewed here fit Wood's categorization scheme well.

Hammer (1984) developed a rule-based procedural aid capable of identifying pilot errors in procedure execution. Similarly, Search Technology's Hazard Monitor detects flight management system programming errors resulting in inappropriate airplane configuration (Search Technology, 1995). Hammer's system as well as the Hazard Monitor are intelligent advisors fitting Woods description of a critiquing advisor that "analyzes a user's decision, solution or plan of action in order to detect errors, verify the adequacy of the human's decision, or suggest improvements" (p. 153). Diverter, developed by Lockheed Aeronautical Systems Company, and Finder, developed by Sextant Avionique are advisors performing high level planning functions. Planning on the flightdeck is performed less frequently compared with continuous system monitoring, and flight control tasks and involves aggregating vast amounts of data. Typical planning tasks on the flightdeck have the objective to devise or revise a flight path. During initial flight planning a flight path is created that best meets multiple criteria such as economy and flight time. During flight, the flightcrew may have to revise the flight path to react to an emergency situation or to avoid bad weather. In case of a diversion that is selecting an alternate destination airport in an emergency situation, the flight-crew has to quickly make an assessment of which airport fits best given a number of constraints. Diverter as well as Finder support real-time flight planning taking into account all available data such as fuel capacity, weather forecasts, ground facilities, regulation rules, and passenger-based regulations (Bittermann et al., 1992; Rudolph et al., 1990).

The Cockpit Assistant System (CASSY) is more comprehensive than the systems mentioned above as it integrates flight planning, system monitoring, situation assessment, pilot intent and error recognition, plan execution, and an advanced human-machine interface featuring graphic displays in combination with a speech recognition and speech synthesis into one intelligent pilot aid (Gerlach et al., 1995; Onken, 1992, 1995). Advisory aiding is performed by continually checking pilot conformance to ATC instructions. If CASSY detects a significant deviation the pilot is alerted. CASSY functions like a semi-autonomous subordinate when the pilot authorizes CASSY to execute a flight plan previously generated by CASSY or the flightcrew. The advanced interface implemented in CASSY can be regarded as an intelligent information system. The Pilot's Associate (PA) uses expert system technology to aid fighter pilots in tasks required for air combat (Rouse et al., 1990; Banks & Lizza, 1991). Since the PA combines monitoring, planning, pilot intent and error recognition, plan execution, and display management, it exhibits characteristics of intelligent advisory, subordinate and information systems.

The AgendaManager (AM) checks flightcrew compliance to ATC instructions, monitors system status, detects goal conflicts between the flightcrew and the auto flight system, and prioritizes functions (Funk & Braune, 1997, Funk et al., 1997). Pilot goals are declared verbally mostly as a response to ATC clearances and progress towards achieving those goals is continuously monitored. If progress is determined to be unsatisfactory, warning messages are presented visually to alert the crew. Given these capabilities, the AM is an intelligent advisory and information system. The AgendaManager evolved as a mean to facilitate Agenda Management (Funk & McCoy, 1996). In the context of Agenda Management, a function is a set of activities to achieve a goal, a desired state of the aircraft. Functions are performed by human actors, the flightcrew, or by machine actors such as the autopilot. For instance, if the goal is to climb and maintain 10,000 feet, the flightcrew has the choice to hand-fly the airplane to the target altitude or program the autopilot to perform this function automatically. The AM is decomposed into software objects representing entities such as actors (pilot, automation), aircraft systems, goals, and functions. Software objects representing functions are called Function Agents and monitor whether their corresponding goals are being achieved in a satisfactory and timely manner. The method called fuzzy function assessment (FFA) described in this thesis is aimed at facilitating the logic, Function Agents use to determine how satisfactorily goals are being pursued.

2.3 Fuzzy Function Assessment (FFA)

Assessing a function is a process that unveils abnormal behavior and detects problems before they mature; thus, it is an operation that is likely to increase safety (Wiener, 1989). Assessing a function involves determining how close the current state is with regards to the goal and analyzing the magnitude of the deviation from the goal in a continous manner in order to recognize if progress is being made. In addition, a function is deemed unsatisfactory if safety constraints are violated. Current system monitors do not take the flightcrew's goals into account and therefore lack the capability to indicate if the airplane is heading in the wrong direction. As function assessment is embedded as part of the AM, the flightcrew's goal is known at any point of time; thus, function assessment with regards to pilot goals is feasible. What motivation is there to automate function assessment? Function assessment is a repetitive monitoring process that when performed incorrectly or not performed at all may threaten mission safety. For example, the flightcrew of Eastern Air Lines L-1011, distracted by a minor landing gear indicator malfunction, failed to monitor the function to maintain 2,000 feet. Starting at 2,000 feet, the airplane gradually descended and crashed into ground, killing 99 passengers. According to the National Transportation Safety Board the probable cause of this accident was "the failure of the flightcrew to monitor the flight instruments during the final 4 minutes of flight" (NTSB, 1973). Since humans are inherently bad monitors (Wiener, 1985, 1988; Parasuraman, 1996) automating functions, which monitor a process, is justified as long as "the detection algorithms and associated software are reliable" (Parasuraman, 1996. p. 96). Humans tend to perform salient tasks first in a serial manner (Chou, 1992). Aviate functions involve controlling the aircraft in space and attitude and are considered the most important ones. Due to the lack of cues alerting the crew of high priority aviate functions deviating from declared goals, less important functions such as diagnosing a malfunctioning gear indicator light may receive more attention than necessary. By delivering feedback indicating if functions are in agreement with declared goals, aviate functions are likely to receive the right level of attention and therefore, function prioritization errors are mitigated.

In this thesis FFA is implemented for the *capture altitude* function. Capturing in this context means to attain and maintain a target altitude. A *capture altitude* function is performed satisfactorily if and only if progress is made towards attaining and maintaining a target altitude in a safe manner. The following two cases show that automation can be used to climb to an altitude in a manner that is not safe. In 1979, Aeromexico, flight 945, a DC-10 cleared to climb to 31,000 ft. stalled at 29,800 ft. and was recovered at 18,900 ft. Heavily damaged, the airplane landed at its destination. According to the NTSB, the flightcrew programmed the autothrust system to maintain an airspeed of 320 kts and the autopilot was set to maintain a vertical speed of 1,200 ft per minute. Once maximum continuous thrust was commanded from the autothrust system to maintain 320 kts at high altitude, the autopilot increased pitch attitude to reach the selected vertical speed. Eventually, the autopilot increased pitch beyond operating limits and caused the airplane to stall. The flightcrew failed to "follow standard climb procedures and to adequately monitor the aircraft's flight instruments" (NTSB, 1979). In 1994, Airbus A330-322 crashed during a test flight at Toulouse Blagnac Airport, France. According to Billings, "it was found that the aircraft autopilot had gone into altitude acquisition (ALT) mode. In this mode, there was no maximum pitch limitation in the autoflight system software. As a consequence, ... the autopilot can induce irrelevant pitch attitudes since it is still trying to follow an altitude acquisition path which it cannot achieve" (Billings, 1996, p. 181).

In light of potentially hazardous function execution by automation or the flightcrew, FFA is designed to evaluate if a function is being carried out with respect to the goal and without creating a hazardous situation. The hazard assessment part of FFA can be seen as an "electronic cocoon" (Wiener, 1989, p. 5) that senses to what degree the aircraft violates the safety envelope. As mentioned earlier, many automated warning and alerting devices issue nuisance warnings that impede their total effectiveness. A type of FFA is desirable that matches human expert assessment capabilities to alleviate the problem of nuisance warnings and as suggested by Wiener indicates the degree of emergency (1980).

3. Research Objectives

The primary objectives of this study were to 1: implement FFA for the capture altitude function; 2: calibrate FFA to emulate human expert assessment; 3: evaluate if the implemented FFA is reliable and coincides with human expert assessment. A secondary but important objective was to develop software making it easy to rapidly implement FFA.

The methods to reach the above objectives are described in chapters 4 - 6 and are organized as follows: chapter 4 reviews fuzzy systems as the underlying technology for FFA. Chapter 5 describes the use of a genetic algorithm to train fuzzy systems to match human assessment. Chapter 6 deals with how fuzzy systems were developed to facilitate FFA.

4. Fuzzy Systems

FFA relies on fuzzy systems using fuzzy if-then rules to transform system inputs to system outputs. Fuzzy systems are based on fuzzy logic theory (Munakata & Jani, 1994). The way fuzzy logic is used in this study is presented in the following section for the reader's convenience. First, background information about fuzzy logic in general is provided. Next, it is shown how fuzzy logic helps overcome the shortcomings of traditional logic with regards to representing the reality, as humans perceive it. Finally, basic concepts of fuzzy logic pertaining to fuzzy systems are described in more detail in order to lay a solid foundation for chapter 6 in which the development of fuzzy systems for FFA is described.

4.1 The History of Fuzzy Logic

Fuzzy logic provides a method for computers to represent vague or imprecise concepts such as warm, tall, and expensive. While in terms of Boolean logic an object for example is either warm or not warm, fuzzy logic introduces the degree of membership that quantifies to what degrees an object belongs to various sets such as the set of cold, and hot objects. So, a warm object may belong to the set of cold objects to degree 0.1 and the set of hot objects to degree 0.9. The degree of membership ranges from 0 to 1, where 0 means no membership and 1 full membership.

Since its invention in 1965, fuzzy logic has gained in importance and many successful applications of fuzzy logic, ranging from industrial process control to consumer products to aerospace and bioengineering, provide empirical evidence that fuzzy logic is a practical tool to solve real problems (Langari & Yen, 1994). Fuzzy logic went on a journey around the globe, which is well documented by von Altrock (1995). Lotfi A. Zadeh devised fuzzy logic at the University of Berkeley in California in 1965.

Initially, fuzzy logic faced harsh criticism from statisticians who believed that probability theory already provided the tool set to model human-like decision making and therefore argued that fuzzy logic was unnecessary. Given the controversy about fuzzy logic in the U.S., it was not surprising to see the first technical application to come from Europe. Mamdani and Assilian (1975) developed the first fuzzy rule-based control system implemented on a laboratory-scale steam engine in England. Encouraged by the good result, more and more applications of fuzzy logic followed in Europe as well in the United States. However, the applications in Europe were mostly limited to process control automation while in the U.S. the focus was on military applications. A large-scale breakthrough of fuzzy logic did not occur until Japanese industry became interested in the technology. First, Japanese companies applied fuzzy logic for various control applications such as in the case of the Sapporo subway control system (Yasunobu, Miyamoto, and Ihara, 1983). The subsequent "history of industrial applications of fuzzy logic in Japan," (p. 43) is well documented by Hirota (1995). Starting in 1987, through a collaborative effort of universities, companies and the government, fuzzy logic experienced a phenomenal boom in Japan. The merits of this collaboration became visible when Japanese consumer goods manufacturers incorporated fuzzy logic in all kinds of home electronics ranging from camcorders to washing machines. The proliferation of fuzzy logic in the consumer goods sector was accompanied by strong marketing campaigns promoting the idea of intelligent consumer goods through fuzzy logic. In fact, fuzzy logic became one of the buzzwords of the nineties. A new concept called "softcomputing", which combines neural networks, fuzzy logic, genetic algorithms and other artificial intelligence methods, is studied by many researchers to solve complicated problems and make consumer products smarter (Dote, 1995; Linkens & Nyongesa, 1996; Munakata & Jani, 1994; Zadeh, 1994). Fuzzy logic is still booming in Japan and is supported by huge governmental budgets, while in the U.S. and Europe, in light of the strategic role of fuzzy logic, measures are under way to catch up with the Japanese competition.

4.2 Boolean vs. Fuzzy Logic

In this section the fundamental principle of fuzzy logic is illustrated by first showing how Boolean logic fails to address the way humans process information and perceive reality and then, by describing how fuzzy logic extends Boolean logic in order to handle imprecision and vagueness.

4.2.1 The Weakness of Boolean Logic

In a world modeled in Boolean logic, questions are answered in a yes-or-no rather than in a more-or-less fashion. In the same way, statements are true or false and objects belong to a set or not. In well defined domains without imprecision and vagueness, Boolean logic works well. For example, consider you want to define A as a set containing all numbers that are greater than 10. A priori, it is known which numbers belong to A and which ones do not. Thus, defining $A := \{x|\, x > 10\}$ is legitimate. But now consider the task of defining B as a set containing all *old* humans. Here, fuzziness is introduced because it is no longer evident which humans would be in B and which ones would not. An attempt to define the set in crisp terms such as $B := \{x \text{ is human}|\ \text{age}(x) \geq 60 \text{ years}\}$ would be counterintuitive and also introduces an artificial and arbitrary threshold. Classifying a 60 year-old person as *old* while considering a 59.99 year-old not *old* seems to misrepresent the true state of reality. A sharp boundary is counterintuitive in most real situations, since an insignificant and almost unidentifiable difference between two objects results in a different classification of those two. Conventional logic bears the problem that whenever the attempt is made to model a system, whose elements are imprecise and vague, precision has to be artificially injected. The implication for a rule-based system is that more rules are needed to model the system behavior in crisp terms than the fuzzy formulation of the system would require. A large number of rules are needed to cover all possible situations and to prevent the system from reacting too sensitively under varying system inputs. I explain the complications of Boolean logic regarding rule-based systems using the following example. Consider the rule-based system with *airspeed* and output *stall*:

IF *airspeed* is *low* THEN *stall* is *likely.*

IF *airspeed* is *high* THEN *stall* is *unlikely.*

low(*airspeed*):= {

true, for *airspeed* ≤ 140 kts;

false, for *airspeed* > 140 kts;}

high(*airspeed*):= {

true, for *airspeed* > 140 kts;

false, for *airspeed* ≤ 140 kts}

When increasing *airspeed, stall* abruptly changes from *likely* to *unlikely* at the defined threshold of 140 kts, which separates *low* from *high* airspeed. In order to reduce sensitivity, the system has to be modified in the following way: 1. increase the granularity of the output *stall* into say, *likely, possible*, and *unlikely*; 2. increase the granularity of input *airspeed* into say, *low*, *medium,* and *high*; 3. add another rule resulting in the modified rule base:

IF *airspeed* is *medium* THEN *stall* is *possible*

IF *airspeed* is *low* THEN *stall* is *likely*

IF *airspeed* is *high* THEN *stall* is *unlikely*

Still, even after this modification the system behavior changes abruptly at the threshold values of *medium*, *low*, and *high* of the system input *airspeed.* An adequately smooth behavior can only be obtained by increasing the granularity of *stall* and *airspeed* significantly and adding a considerable number of rules.

To summarize, crisp logic faces the following two major complications as described by Zimmermann (1991):

1. Real situations are very often not crisp and deterministic and they cannot be described precisely.
2. The complete description of a real system often would require far more detailed data than human being could ever recognize simultaneously, process and understand. (p. 3)

4.2.2 The Fuzzy Logic Benefit

While fuzzy logic is based on a strict mathematical framework, it allows one to model imprecise and vague information at a high level of abstraction. One of the main benefits of fuzzy logic is its capability to model human logic (von Altrock, 1997). Humans do not devise a rule for each possible situation; instead humans formulate rules for a few typical situations and arrive at a conclusion through approximation. In that respect, fuzzy logic is very similar to human logic. Whenever a decision process for certain typical cases can be formulated in rules, fuzzy logic will use the limited knowledge contained in the rules to approximate a solution for any given case by selecting rules whose if-part match the given situation to some degree, and computing an aggregated result that combines the net effects of those selected rules. For expert system designers fuzzy logic has one very desirable characteristic: by applying fuzzy logic, the number of rules can be usually reduced significantly; thus drastically simplifying the design and reducing design time (Linkens & Nyongesa, 1996).

4.3 Fuzzy Logic Framework

The material presented here is a mathematical framework, which is used for the implementation of fuzzy systems in this study. The technology involved is based on the work of von Altrock (1993, 1997), Dote (1996), Langari & Yen (1993), Mendel (1995), Sugeno (1985), Zadeh (1965, 1975a, 1975b, 1975c), and Zimmermann (1991). For further information about fuzzy logic, the reader is referred to the above sources.

4.3.1 Fuzzy Set

A fuzzy set A is characterized by its membership function $f_A: X \rightarrow [0,1]$ that assigns a real number between 0 and 1 to any object in the universe of discourse X on which A is defined. The value of the membership function $f_A(x)$ for a specific element $x \in X$ represents the degree of membership of x in A. Important operations on fuzzy sets are union and intersection, which are defined as follows:

$$f_{A \cap B}(x) = \min(f_A(x), f_B(x))$$

$$f_{A \cup B}(x) = \max(f_A(x), f_B(x))$$

In this example the fuzzy set *low* is defined on the universe of discourse X which contains all possible altitude values. Figure 4.1 depicts the membership function values $f_{low}(x)$ for certain altitude values.

$$f_{low}(x) = \begin{Bmatrix} 0 & , x \leq 250 \\ -\frac{1}{3} + \frac{1}{750}x & , 250 < x \leq 1000 \\ \frac{5}{3} - \frac{1}{1500}x & , 1000 < x \leq 2500 \\ 0 & , x > 2500 \end{Bmatrix}$$

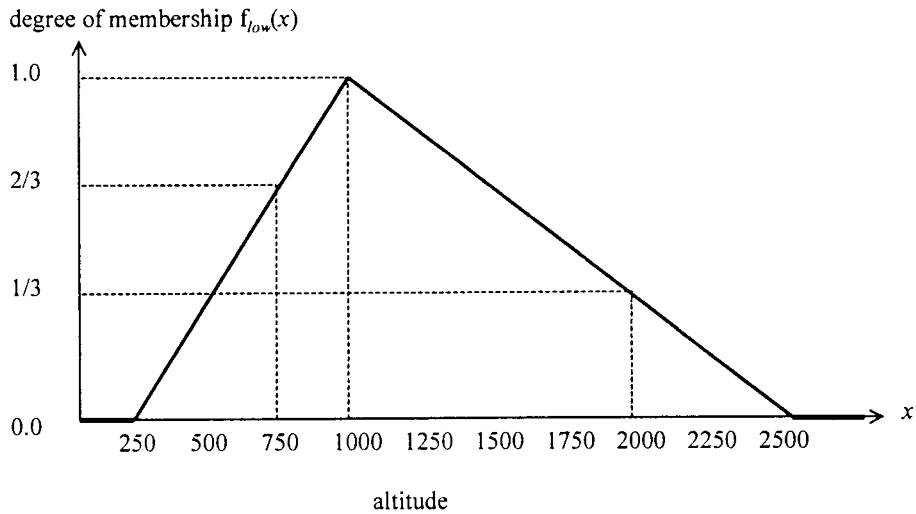


Figure 4.1 Membership function f_{low}

While an altitude of 750 ft. is considered low to degree of 2/3; an altitude of 1000 ft. is considered low to degree of 1.0, and an altitude of 2000 ft. is considered low to degree of 1/3.

4.3.2 Membership Function

In this study four different kinds of membership functions, the so-called linear standard membership functions, are used. Standard membership functions are normalized, that is, their maximum is always 1, and their minimum is 0. The four different types are described as follows:

4.3.2.1 Z-Type Standard Membership Function

The Z-type membership function is defined as follows:

$$f_A(x) = \begin{cases} 1 & ,x \le t \\ \dfrac{r}{r-t} + \dfrac{1}{t-r}x & ,t < x \le r \\ 0 & ,x > r \end{cases}$$

where r and t define the shape of the function as can be seen from figure 4.2.

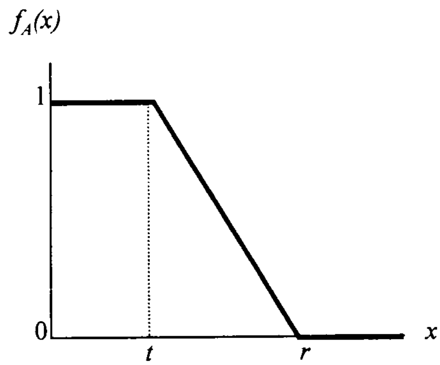


Figure 4.2 Z-type membership function

The typical value $p(A)$ of fuzzy set A whose membership function is Z-type is t. Thus, $p(A) = t$.

4.3.2.2 Λ-Type Standard Membership Function

The Λ-type membership function is defined as follows:

$$f_A(x) = \begin{cases} 0 & ,x < l \\ \dfrac{l}{l-t} + \dfrac{1}{t-l}x & ,l \le x \le t \\ \dfrac{r}{r-t} + \dfrac{1}{t-r}x & ,t < x \le r \\ 0 & ,x > r \end{cases}$$

where l, t, and r define the shape of the function as can be seen from figure 4.3.

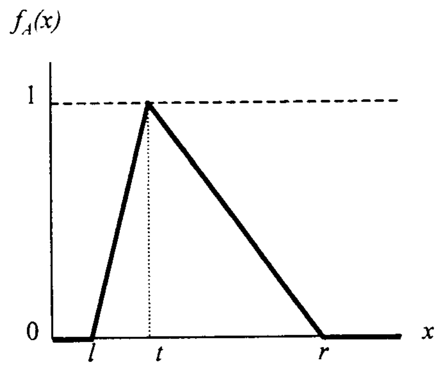


Figure 4.3 Λ-Type Membership Function

The typical value $p(A)$ of fuzzy set A whose membership function is Λ-type is t. Thus, $p(A) = t$.

4.3.2.3 ∏-Type Standard Membership Function

The ∏-type membership function is defined as follows:

$$f_A(x) = \begin{Bmatrix} 0 & ,x \leq l \\ \dfrac{l}{l-t} + \dfrac{1}{t-l}x & ,l < x \leq t \\ 1 & ,t < x \leq e \\ \dfrac{r}{r-e} + \dfrac{1}{e-r}x & ,e < x \leq r \\ 0 & ,x > r \end{Bmatrix}$$

where l, t, e, and r define the shape of the function as can be seen from figure 4.4.

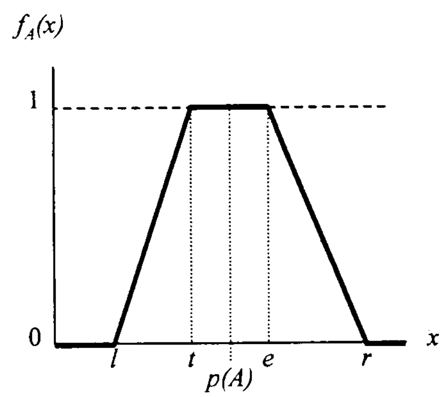


Figure 4.4 ∏-Type Membership Function

The typical value $p(A)$ for the fuzzy set A whose membership function is ∏-type is the median of the maximizing interval for which $f_A(x) = 1$, thus $p(A) = \dfrac{e-t}{2}$.

4.3.2.4 S-Type Standard Membership Function

The S-type membership function is defined as follows:

$$f_A(x) = \begin{cases} 0 & ,x \leq l \\ \dfrac{l}{l-t} + \dfrac{1}{t-l}x & ,l < x \leq t \\ 1 & ,x > t \end{cases}$$

where l and t define the shape of the function as can be seen from figure 4.5.

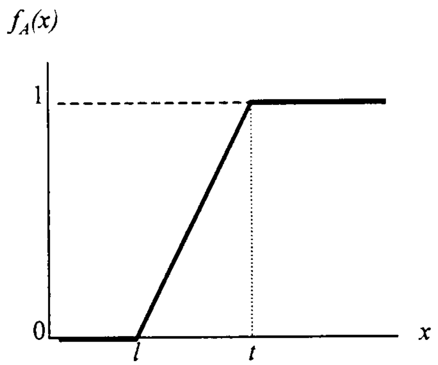


Figure 4.5 S-Type membership function

The typical value $p(A)$ of fuzzy set A whose membership function is S-type is t. Thus, $p(A) = t$.

4.3.3 Linguistic Variable

Within a fuzzy system imprecise knowledge is represented by linguistic variables "whose values are words or sentences in a natural or artificial language" (Zadeh, 1975a, p 199). The motivation for using linguistic variables instead of numerical variables is based on the fact that linguistic variables introduce a high level of abstraction that helps us describe the state of the world in simple linguistic terms. When asked how the weather is, we seldom respond in a precise manner such as "the temperature is exactly 20.5 degrees Celsius and it is raining at a rate of 1.2 mm per hour"; instead we rather give a more rudimentary description such as "it is warm and it is raining very hard". The words "warm" and "very hard" can be interpreted as terms of the linguistic variables "temperature" and "rainfall" respectively. Also, linguistic variables are very helpful when formulating rules such as "IF altitude is *very low* and vertical speed is *slightly negative* THEN flight phase is *landing*". The variables *altitude*, *vertical speed*, and *flight phase* can be associated with linguistic variables; *very low*, *slightly negative*, and *landing* are terms for the respective linguistic variables. For the purposes of this study the concept of a linguistic variable is augmented from the original form defined by Zadeh (1975b) and is described as follows:

Let l denote the name of a linguistic variable. A linguistic variable is decomposed into a set of terms, $T(l) = \bigcup_{k \in Q} \{t_k\}$, where Q = {1, 2, ...,p}; note that $t_k \neq t_t$ for all $(k,t) \in \{(k,t) \in Q \times Q \mid k \neq t\}$. We define a bijective function g_l that maps a term $t \in T(l)$ to an index $k \in Q$. Thus, g_l: $T(l) \to Q$; g_l^{-1} maps an index $k \in Q$ back to the corresponding term $t \in T(l)$; thus, g_l^{-1}: $Q \to T(l)$, and $g_l^{-1}(g_l(t)) = t$. The linguistic value of a linguistic variable can be expressed as a vector of truth values that quantify the degrees to which the terms are valid. So the linguistic value of l is:

$$V(l) = \begin{bmatrix} v_1^l \\ v_2^l \\ \bullet \\ \bullet \\ \bullet \\ v_p^l \end{bmatrix}$$

where $v_k^l \in [0, 1]$ ($k \in$ Q) quantifies the degree to which the term $t = g_l^{-1}(k)$ holds true for variable l. We differentiate between system input linguistic variables, intermediate linguistic variables, and system output linguistic variables. When we do not want to differentiate between the different variable types we simply refer to a linguistic variable l. A system input linguistic variable h and a system output linguistic variable o have the numerical values $x \in X$ and $y \in Y$ respectively while an intermediate linguistic variable i has no numerical value. Any term $t \in T(h)$ of a system input linguistic variable h has a membership function f_t associated with it, likewise any term $r \in T(o)$ of a system output linguistic variable o has a membership function f_r associated with it. None of the terms of an intermediate linguistic variables i relates to a membership function. The following example helps clarify the concept of a linguistic variable.

This example defines the linguistic variables *altitude*, *vertical speed*, and *terrain hazard*. These linguistic variables are used in subsequent examples where they describe properties of an airplane. The system input linguistic variable named *altitude* is decomposed into the terms *very low, low*, and *medium*. Therefore, *T*(*altitude*) = {*very low, low, medium*}. The function $g_{altitude}$ is defined as follows:

$$g_{alitude}(t) = \begin{cases} 1, & t = very\ low \\ 2, & t = low \\ 3, & t = medium \end{cases}$$

The universe of discourse $X = [0, 5000]$ is the set of altitude values (in ft.) under consideration. The membership functions of $t \in T(altitude)$ are graphed in figure 4.6.

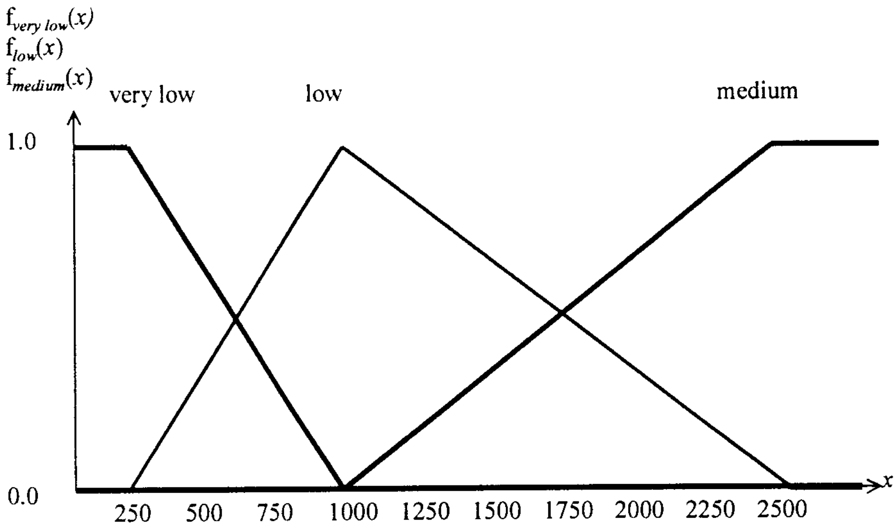


Figure 4.6 Terms and membership functions of *altitude*

The system input linguistic variable named *vertical speed* is decomposed into the terms *negative large,* and *negative*. Therefore, *T*(*vertical speed*) = { *negative large*, *negative*}. The universe of discourse X = [-6000, 0] is the set of vertical speed values (in ft. per minute) under consideration. The function $g_{vertical\ speed}$ is defined as follows:

$$g_{vertical\ speed}(t) = \begin{cases} 1, & t = negative\ large \\ 2, & t = negative \end{cases}$$

The membership functions of $t \in T(vertical\ speed)$ are graphed in figure 4.7.

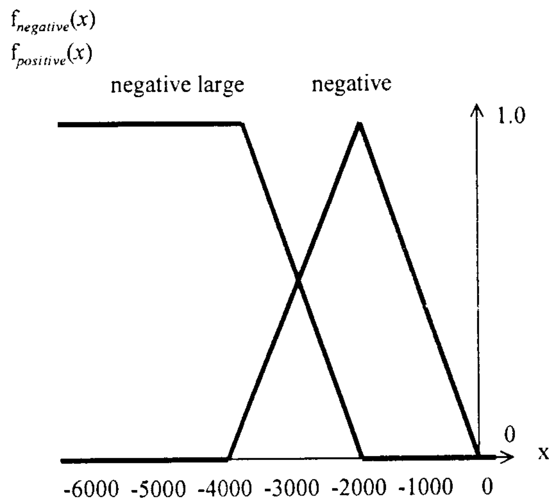


Figure 4.7 Terms and membership functions of *vertical speed*

The system output linguistic variable named *terrain hazard* is decomposed into *low*, *medium*, and *high*. Therefore, *T(terrain Hazard)* = {*low*, *medium*, *high*}. I define the universe of discourse $X = [0, 10]$. X is the set of crisp terrain hazard values quantifying terrain hazards numerically. An output value close to 10 indicates a high terrain hazard whereas a value close to 0 indicates a low terrain hazard. The function $g_{terrain\ hazard}$ is defined as follows:

$$g_{terrain\ hazard}(t) = \begin{cases} 1, & t = low \\ 2, & t = medium \\ 3, & t = high \end{cases}$$

The membership functions of $t \in T(terrain\ hazard)$ are graphed in figure 4.8.

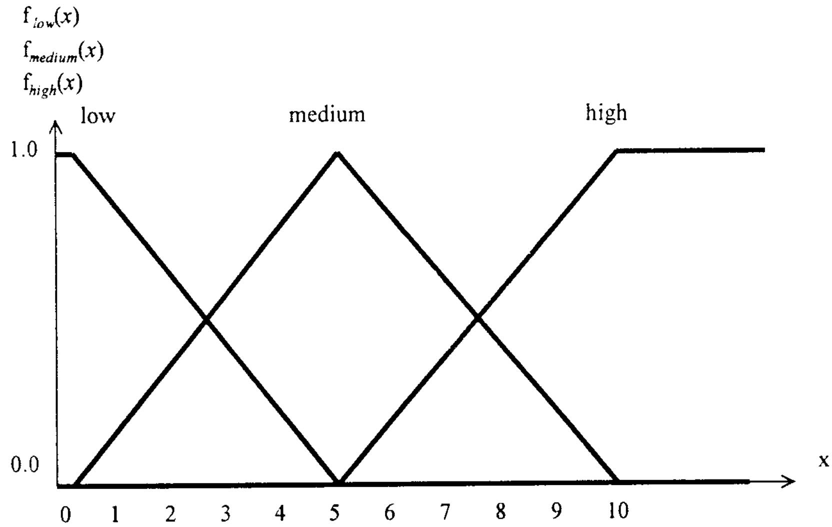


Figure 4.8 Terms and membership functions of *terrain hazard*

4.3.4 Fuzzy Rule Block

Fuzzy reasoning is the mechanism used to define system behavior for fuzzy systems. The knowledge that governs the computation of system outputs from system inputs is described in fuzzy rules. A fuzzy rule block B is a set of fuzzy rules. B can be expressed as $B = \{R_1, R_2, \ldots R_n\}$, where R_i is a fuzzy rule, $i \in I = \{1, 2, \ldots, n\}$. In turn R_i is of the form:

$R_i = (d_i, a^i)$ IF (l_1 is $t_{i,1}$) and (l_2 is $t_{i,2}$) and ... (l_m is $t_{i,m}$) THEN (l_{m+1} is $t_{i,m+1}$), (l_{m+2} is $t_{i,m+2}$), ..., (l_{m+z} is $t_{i,m+z}$)

where $d_i \in [0,1]$ is the degree of support of rule R_i, a^i denotes the aggregation operator used to compute the weight of R_i during fuzzy inferencing, $t_{i,j} \in T(l_j), j \in J = \{1, 2, \ldots m+z\}$, is a term of the linguistic variable l_j in fuzzy rule R_i. Note that $l_j \neq l_k$ for all $(j,k) \in \left\{(j,k) \in J \times J \middle| \, j \neq k\right\}$. Further, we denote the set of linguistic variables appearing in the if-part of B as $F(B)=\{l_1, l_2, \ldots, l_m\}$ and call the variables within $F(B)$

block input variables, and the set of linguistic variables appearing in the then-part of B as $E(B)=\{l_{m+1}, l_{m+2}, \ldots, l_{m+z}\}$ and call the variables within $E(B)$ block output variables. The purpose of the degree of support d_i of rule R_i is to differentiate between strong and weak rules. The higher the degree of support the stronger the impact of the rule is on the solution. Likewise, the lower the degree of support the weaker the impact of the rule is on the result.

Table 4.1 represents a rule block computing the terrain hazard value given the altitude and vertical speed. Rule 1 for instance states "If altitude is very low and vertical speed is negative large then terrain hazard is high."

Table 4.1 Rule block example

			IF		**THEN**
Rule #	degree of support	aggregation operator	altitude	vertical speed	terrain hazard
1	1	min	very low	negative large	high
2	1	min	very low	negative	medium
3	1	min	low	negative large	high
4	1	$\mu_{\gamma=0.4}$	low	negative	medium
5	1	min	medium	negative large	medium
6	1	min	medium	negative	low

In the above rule block *altitude*, *vertical speed*, and *terrain hazard* are linguistic variables. The terms *very low, low,* and *medium* belong to *altitude*; *negative large*, and *negative* are linguistic terms of *vertical speed*; *low*, *medium* and *high* are linguistic terms of *terrain hazard*. The linguistic variables *altitude* and *vertical speed* are block inputs while *terrain hazard* is a block output.

4.3.5 Fuzzy System

A fuzzy system S is comprised of a queue of rule blocks denoted by $C = \{B_1, B_2, ..., B_w\}$, a set of system input linguistic variables denoted by $H = \{h_1, h_2, ..., h_u\}$, and a set of system output linguistic variables denoted by $O = \{o_1, o_2, ..., o_v\}$. Essentially, S implicitly describes a function that maps the numerical input vector $NH = (x_1, x_2, ..., x_u)$ to the numerical output vector $NO = (y_1, y_2, ..., y_v)$; where $x_1, x_2, ..., x_u$ are the numerical values of $h_1, h_2, ..., h_u$ and $y_1, y_2, ..., y_v$ are the numerical values of $o_1, o_2, ..., o_v$. Therefore:

$$NO = S(NH)$$

The set of linguistic variables contained in S is

$$V = \bigcup_{q=1}^{w} E(B_q) \cup F(B_q) = \{l_1, l_2, \ldots, l_h\}.$$

Further S fulfills the following constraints:

1. $\forall B_q \in C: \quad (E(B_q) \cap F(B_q) = \varnothing)$
2. $\forall B_q \in C : (\forall l_j \in F(B_q) : (l_j \in H \vee l_j \in E(B_x), x \in W - \{q\}))$
3. $\forall B_q \in C : (\forall l_j \in E(B_q) : (l_j \in O \vee l_j \in F(B_x), x \in W - \{q\}))$
4. $\forall (B_q, B_t) \in \{(B_q, B_t) \in C \times C | q \neq t\} : (E(B_q) \cap E(B_t) = \varnothing)$
5. $\forall (B_q, B_t) \in \{(B_q, B_t) \in C \times C | q \neq t\} : (F(B_q) \cap E(B_t) \neq \varnothing \Rightarrow t < q)$

where $W = \{1, 2, ..., w\}$.

The first constraint states that the set of block input variables and the set of block output variables of any rule block are disjunct. The second constraint states that any block input variable of a rule block is an element of H or an element of the set of block output variables of another rule block. The third constraint states that any block output variable of a rule block is an element of O or an element of the set of block input variables of another rule block. The fourth constraint prevents the manipulation of any variable by multiple rule blocks. The fifth constraint establishes an order within C such that any particular block in C does not contain a block input variable that is also a block output variable of any of the subsequent blocks.

4.3.6 Fuzzy Logic Algorithm

While the previous sections introduced fuzzy sets, linguistic variables, fuzzy rules, fuzzy rule blocks, and fuzzy system this section explains now the process computing system outputs from system inputs. The algorithm can be divided into three procedures as follows: 1. fuzzification (calculation of linguistic values for the system input linguistic variables); 2. fuzzy Inference (calculation of linguistic values for the system output linguistic variables); 3: defuzzification (translation of system output linguistic variable values into corresponding numerical system outputs). Figure 4.9 shows an example of a fuzzy system consisting of four inputs, two outputs, and three rule blocks. Note the usage of intermediate linguistic variables i_1 and i_2.

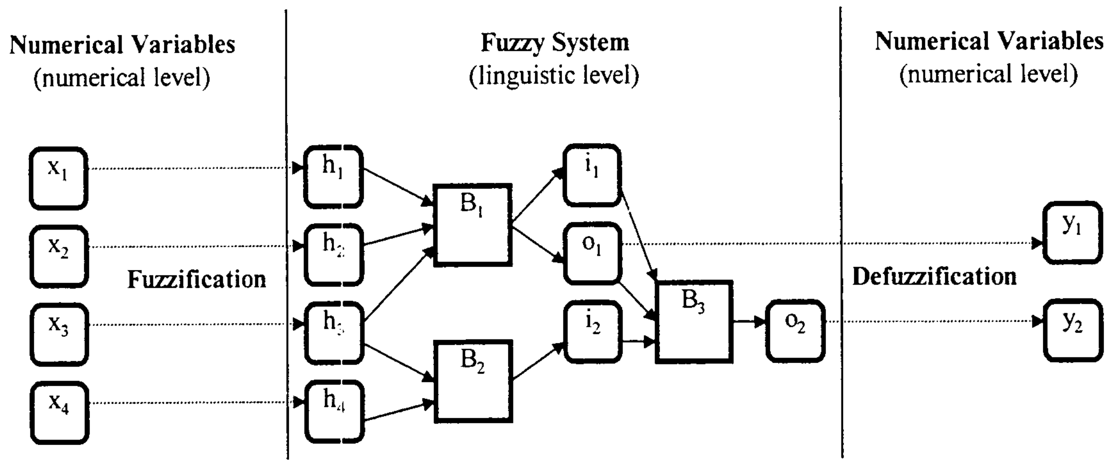


Figure 4.9 Example of fuzzy system

The first procedure, fuzzification, is performed in order to obtain a level of abstraction needed for the application of fuzzy if-then rules. In general, fuzzification is the process of computing the membership function values associated with the linguistic terms of system input linguistic variables given the numerical system inputs. After the fuzzification step, fuzzy inference evaluates the solution in terms of the system output linguistic variables. The latter are translated into numerical system outputs in the third step, called defuzzification.

Fuzzification, fuzzy inference, and defuzzification are performed on a fuzzy system. Let S be the fuzzy system under study, $C = \{B_1, B_2, ..., B_w\}$ be a queue of rule blocks of S, $H = \{h_1, h_2, ..., h_u\}$ be a set of system input linguistic variables of S, $O = \{o_1, o_2, ..., o_v\}$ be a set of system output linguistic variables of S, $x_1, x_2, ..., x_u$ be the numerical inputs for the respective system input linguistic variables $h_1, h_2, ..., h_u$, $y_1, y_2, ..., y_v$ be the numerical outputs for the respective system output linguistic variables $o_1, o_2, ..., o_v$.

4.3.6.1 Fuzzification

Fuzzification is the process of calculating the linguistic value for each system input linguistic variable $h_j \in H, j \in J = \{1, 2, ..., u\}$ as follows:

$$V(h_j) = \begin{bmatrix} v_1^{h_j} \\ v_2^{h_j} \\ \bullet \\ \bullet \\ \bullet \\ v_p^{h_j} \end{bmatrix} = \begin{bmatrix} f_{g_{h_j}^{-1}(1)}(x_j) \\ f_{g_{h_j}^{-1}(2)}(x_j) \\ \bullet \\ \bullet \\ \bullet \\ f_{g_{h_j}^{-1}(p)}(x_j) \end{bmatrix}$$

where $v_k^{h_j}$ ($k \in K = \{1, 2, ..., p\}, p = |T(h_j)|$) is the truth value relating to the term $g_{h_j}^{-1}(k) \in T(h_j)$ of system input linguistic variable h_j, and $f_{g_{h_j}^{-1}(k)}$ is the corresponding membership function.

4.3.6.2 Fuzzy Inference

Within S, fuzzy inference is performed for each rule block in a sequential order imposed by C. For each rule block an aggregation and composition process is performed. The aggregation process of a rule block involves the computation of rule weights from the truth values of the block input variables. The composition process determines the values of the block output variables. A rule weight quantifies to what extent the premises of the if-part of the rule hold true for a given situation.

Fuzzy inference is performed in an order imposed by C, thus first B_1, then B_2, ..., and finally B_W. Fuzzy inference of a generic fuzzy rule block $B \in C$ works as follows:

Let $B = \{R_1, R_2, \ldots R_n\}$, a^i ($i = 1, 2, \ldots, n$) be the aggregation operators, d_i ($i = 1, 2, \ldots, n$) be the degrees of support, $F(B) = \{l_1, l_2, \ldots, l_m\}$ be the set of block input variables of B, $E(B) = \{l_{m+1}, l_{m+2}, \ldots, l_{m+z}\}$ be the set of block output variables of B, $t_{i,j} \in T(l_j)$ ($j = 1, 2, \ldots m+z$, $i = 1, 2, \ldots, n$) are terms. Further, we define:

(4.1) $I^j(t) = \{\, i \in Q \mid t_{i,j} = t \}$

where $I^j(t)$ is a subset of $Q = \{1, 2, \ldots, n\}$, $n = |B|$, and $t \in T(l_j)$. $I^j(t)$ is the set of indices corresponding to rules that contain the premise "l_j is t", where l_j is a linguistic variable, and t is a term of l_j.

4.3.6.2.1 Rule Block Aggregation

Then aggregation via aggregation operator a^i yields the weight w_i representing the conjunction of m premises of the if-part of rule R_i:

$$w_i = a^i\left(v^{l_1}_{g_{l_1}(t_{i,1})}, v^{l_2}_{g_{l_2}(t_{i,2})}, \ldots, v^{l_m}_{g_{l_m}(t_{i,m})}\right)$$

where $v^{l_j}_{g_{l_j}(t_{i,m})}$ is the truth value associated with term $t_{i,m}$ of variable l_j.

In this study a^i is either the min or the γ-operator. The min-operator simply yields the smallest of the m truth values, while the γ-operator is more complex and is defined as follows:

$$\mu_\gamma(\mu_1, \mu_2, \ldots, \mu_m) = \left(\prod_{i=1}^{m} \mu_i\right)^{(1-\gamma)} \cdot \left(1 - \prod_{i=1}^{m} (1 - \mu_i)\right)^{\gamma}$$

4.3.6.2.2 Rule Block Composition

We compute the linguistic values for all block output variables $l_j \in E(B)$ of B. The linguistic value of block output variable $l_j \in E(B)$ is computed as follows:

$$V(l_j) = \begin{bmatrix} v_1^{l_j} \\ v_2^{l_j} \\ \cdot \\ \cdot \\ \cdot \\ v_p^{l_j} \end{bmatrix} \qquad (4.2)$$

where a truth value $v_k^{l_j}$, k∈{1, 2, ..., p}, is given as:

$$v_k^{l_j} = \begin{cases} 0, & g_{l_j}^{-1}(k) \notin \bigcup_{i \in Q} \{t_{i,j}\} \\ \max\limits_{i \in I^j(g_{l_j}^{-1}(k))} (w_i d_i), & g_{l_j}^{-1}(k) \in \bigcup_{i \in Q} \{t_{i,j}\} \end{cases} \qquad (4.3)$$

4.3.6.3 Defuzzification

Defuzzification is the computation of the crisp values $y_1, y_2, \ldots, y_v$ from the linguistic values of the output linguistic variables $o_1, o_2, \ldots, o_v$. Let o be an output linguistic variable, and $r \in T(o)$ be a term of o. $p(r)$ is the typical value of the fuzzy set characterized by the membership function associated with term r. The Center of

Maximum defuzzification method (Altrock, 1997, p 45) computes the crisp value y from of the linguistic value of o as follows:

$$(4.4) \qquad y = \frac{\sum_{r \in T(o)} v^{o}_{g_o(r)}\, p(r)}{\sum_{r \in T(o)} v^{o}_{g_o(r)}}$$

where $v^{o}_{g_o(r)}$ is the truth value associated with term r of variable o.

4.3.6.4 Example Fuzzy Logic Algorithm

The whole algorithm can be easily illustrated using the rule block B of illustrated in table 4.1 and the linguistic variables depicted in figures 4.6, 4.7, and 4.8. Thus the fuzzy system S is characterized by $C=\{B\}$, $H=\{$*altitude*, *vertical speed*$\}$, $O=\{$*terrain hazard*$\}$. We set:

$l_1 = h_1 =$ *altitude*, $x_1 = 750$

$l_2 = h_2 =$ *vertical speed*, $x_2 = -3000$

$l_3 = o =$ *terrain hazard*

The objective is to compute y, the crisp value that corresponds to the linguistic value of o. By means of fuzzification, the numerical values quantifying *altitude* and *vertical speed* respectively are translated into their corresponding linguistic variable values. Once the values of the linguistic variables *altitude* and *vertical speed* are obtained, fuzzy inference determines the value of the output linguistic variable *terrain hazard*. The result is translated into a numerical value in the defuzzification step.

4.3.6.4.1 Fuzzification Example

Given an altitude of 750 ft. and a vertical speed of −3000 ft. per minute, the membership functions as described in figure 4.6 yield the following results:

$$V(h_1 = altitude) = \begin{bmatrix} v_1^{h_1} \\ v_2^{h_1} \\ v_3^{h_1} \end{bmatrix} = \begin{bmatrix} f_{g_{h_1}^{-1}(1)}(750) \\ f_{g_{h_1}^{-1}(2)}(750) \\ f_{g_{h_1}^{-1}(3)}(750) \end{bmatrix} = \begin{bmatrix} f_{very\ low}(750) \\ f_{low}(750) \\ f_{medium}(750) \end{bmatrix} = \begin{bmatrix} 0.33 \\ 0.67 \\ 0 \end{bmatrix}$$

Regarding the vertical speed the values of the membership functions as illustrated in figure 4.7 are given as follows:

$$V(h_2 = vertical\ speed) = \begin{bmatrix} v_1^{h_2} \\ v_2^{h_2} \end{bmatrix} = \begin{bmatrix} f_{g_{h_2}^{-1}(1)}(-3000) \\ f_{g_{h_2}^{-1}(1)}(-3000) \end{bmatrix} = \begin{bmatrix} f_{negative\ large}(-3000) \\ f_{negative}(-3000) \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

4.3.6.4.2 Fuzzy Logic Inference Example

After the fuzzification step the system inputs are given in terms of linguistic variables. The fuzzy inference step computes the values of the output linguistic variables. Fuzzy inference consists of an aggregation and a composition process for rule block *B*. The if-part of a fuzzy rule defines the extent to which the rule is applicable for a given case. In this case ($x_1 = 750, x_2 = -3000$), *altitude* is *very low* to the degree 0.33, and *vertical speed* is *negative large* to degree 0.5. Consider the if-part of rule 1:

R_1: (1, min) IF *altitude* is *very low* <u>and</u> *vertical speed* is *negative large*

The if-part of rule 1 combines the two conditions "*altitude* is *very low*" and "*vertical speed* is *negative large*". The aggregation operator is the minimum operator, thus, the weight of the rule or the extent to which rule 1 is valid is $w_1 = \min(0.33, 0.5) = 0.33$. Table 4.2 summarizes the result of the aggregation.

Table 4.2 Aggregation example

Rule i	$t_{i,1}$ *(altitude)*	$v^{l_1}_{g_{l_1}}(t_{i,1})$		$t_{i,2}$ *(vertical speed)*	$v^{l_2}_{g_{l2}}(t_{i,2})$		**Result of Aggregation** w_i
1	*very low*	$v^{l_1}_1$	= 0.33	*negative large*	$v^{l_2}_1$	= 0.5	min(0.33; 0.5) = 0.33
2	*very low*	$v^{l_1}_1$	= 0.33	*negative*	$v^{l_2}_2$	= 0.5	min(0.33; 0.5) = 0.33
3	*low*	$v^{l_1}_2$	= 0.67	*negative large*	$v^{l_2}_1$	= 0.5	min(0.67; 0.5) = 0.5
4	*low*	$v^{l_1}_2$	= 0.67	*negative*	$v^{l_2}_2$	= 0.5	$\mu_{0.4}$ (0.67; 0.5) = 0.48
5	*medium*	$v^{l_1}_3$	= 0.0	*negative large*	$v^{l_2}_1$	= 0.5	min(0.0; 0.5) = 0.0
6	*medium*	$v^{l_1}_3$	= 0.0	*negative*	$v^{l_2}_2$	= 0.5	min(0.0; 0.5) = 0.0

To perform the composition we first compute the index sets $I^3(t)$ for each $t \in T(l_3)$ according to equation 4.1:

$I^3(low) = \{6\}$

$I^3(medium) = \{2, 4, 5\}$

$I^3(high) = \{1, 3\}$

This means that premise "*terrain hazard* is *low*" occurs in rule 6, premise "*terrain hazard* is *medium*" occurs in rules 2, 4, and 5, and premise "*terrain hazard* is *high*" occurs in rules 1 and 3.

Note that $\bigcup_{i=1}^{n=|B|} \{t_{i,3}\} = \{low, medium, high\}$. We compute the value of o according to equation 4.2 and 4.3:

$$V(l_3 = terrain\,hazard) = \begin{bmatrix} v_1^o \\ v_2^o \\ v_3^o \end{bmatrix} = \begin{bmatrix} \max\limits_{i \in I^3(low)} (w_i \cdot d_i) \\ \max\limits_{i \in I^3(medium)} (w_i \cdot d_i) \\ \max\limits_{i \in I^3(high)} (w_i \cdot d_i) \end{bmatrix} = \begin{bmatrix} \max(0) \\ \max(0.33, 0.48, 0) \\ \max(0.33, 0.5) \end{bmatrix} = \begin{bmatrix} 0 \\ 0.48 \\ 0.5 \end{bmatrix}$$

The truth values of the terms associated with the output linguistic variable are used in the defuzzification step.

4.3.6.4.3 Defuzzification Example

Defuzzification is performed according to equation 4.4. Figure 4.10 illustrates the fact that the Center of Maximum method computes a crisp value that represents the best compromise between the typical values of the terms.

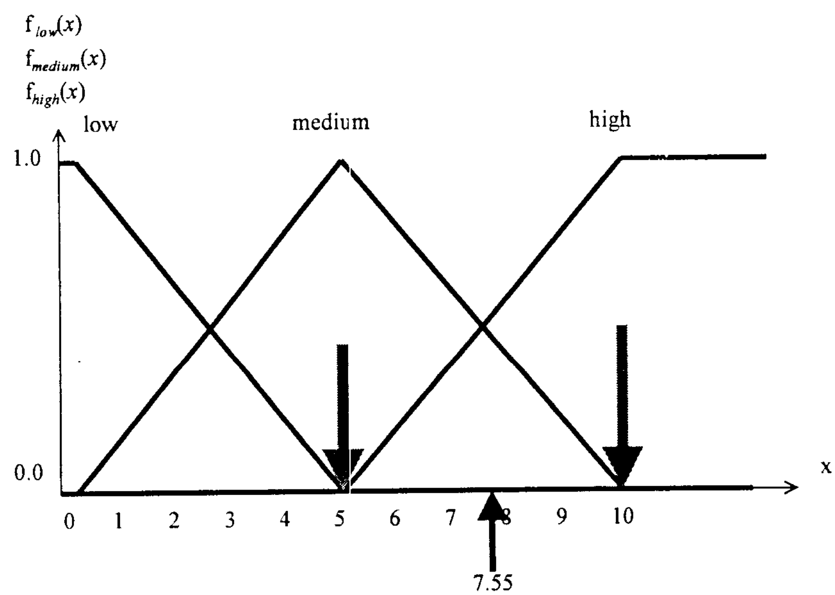


Figure 4.10 Defuzzification example

$$y = \frac{\sum_{r \in T(o)} v^o_{g_o(r)} P(r)}{\sum_{r \in T(o)} v^o_{g_o(r)}} = \frac{v^o_{g_o(low)} \cdot p(low) + v^o_{g_o(medium)} \cdot p(medium) + v^o_{g_o(high)} \cdot p(high)}{v^o_{g_o(low)} + v^o_{g_o(medium)} + v^o_{g_o(high)}}$$

$$= \frac{0 \cdot 0 + 0.48 \cdot 5 + 0.5 \cdot 10}{0 + 0.48 + 0.5} = 7.55$$

Thus, at an altitude of 750 ft. and a vertical speed of –3000 ft. per minute, the terrain hazard is 7.55.

4.4 Fuzzy Logic Tools

Commercial tools such as fuzzyTech, CubiCalc, TILSHELL, and FIDE are available for fuzzy logic implementation (Chiu, 1995). While the commercial packages are very powerful regarding computation speed, the integration of the AgendaManager, implemented in Smalltalk, with those is not as seamless as is possible with a fuzzy logic class library implemented in Smalltalk. Therefore, a fuzzy logic class library and graphical tools were implemented in Smalltalk for representation of membership functions, linguistic variables, rules, rule blocks, and fuzzy systems as defined in section 4.3. The development environment is composed of a fuzzy system editor to define fuzzy system structures, a rule block editor to specify rule blocks, a linguistic variable editor to manipulate linguistic terms and their associated membership functions, and a debugger for system tuning.

4.4.1 Fuzzy System Editor

This section describes the fuzzy system editor tool. The fuzzy system editor allows the system designer to load a previously saved system, create a new system, import linguistic variables and rule bases from saved systems into the currently active system, and save the currently active system, as well as add, remove, and rename linguistic variables and rule blocks within the currently active system. Figure 4.11 shows the graphical user interface of the fuzzy system editor. The currently active system consists of one rule block with inputs altitude and vertical speed, and output terrain hazard. The system designer manipulates variables and rule blocks with the linguistic variable editor and rule block editor respectively.

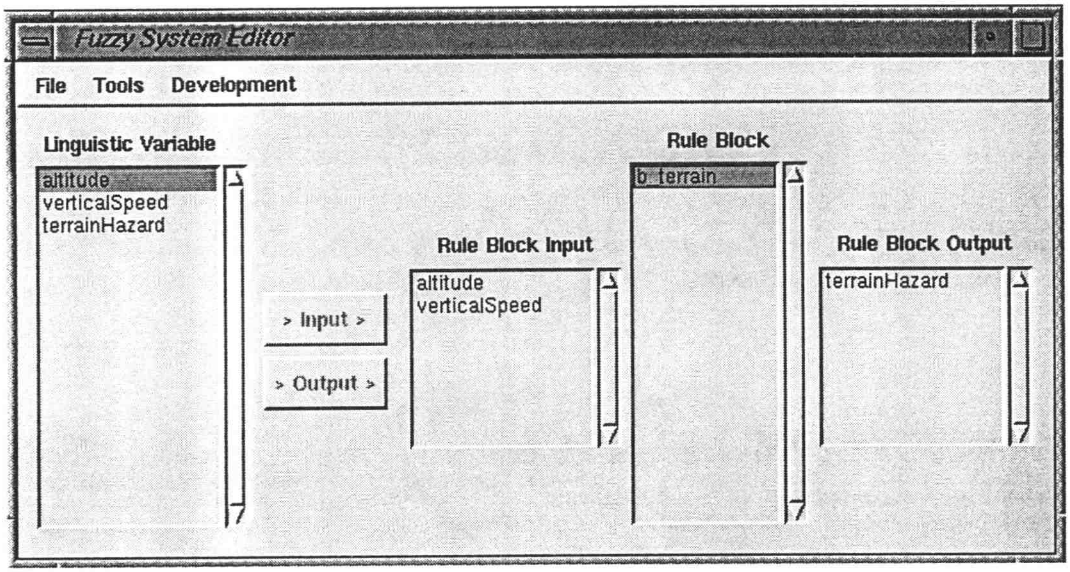


Figure 4.11 Fuzzy system editor interface

4.4.2 Linguistic Variable Editor

The purpose of this tool is to provide an interface for a linguistic variable to manipulate its type, terms, and membership functions. The variable type is system input, system output, or intermediate. The system designer can rename, add, remove, or copy terms at will and manipulate the corresponding membership functions. The membership functions employed are standard linear membership functions as described in section 4.3.2. Figure 4.12 shows the graphical user interface of the linguistic variable editor displaying the properties of the linguistic variable altitude. Altitude is an output variable comprised of the terms very low, low, and medium. The membership functions are plotted on the lower panel of the editor.

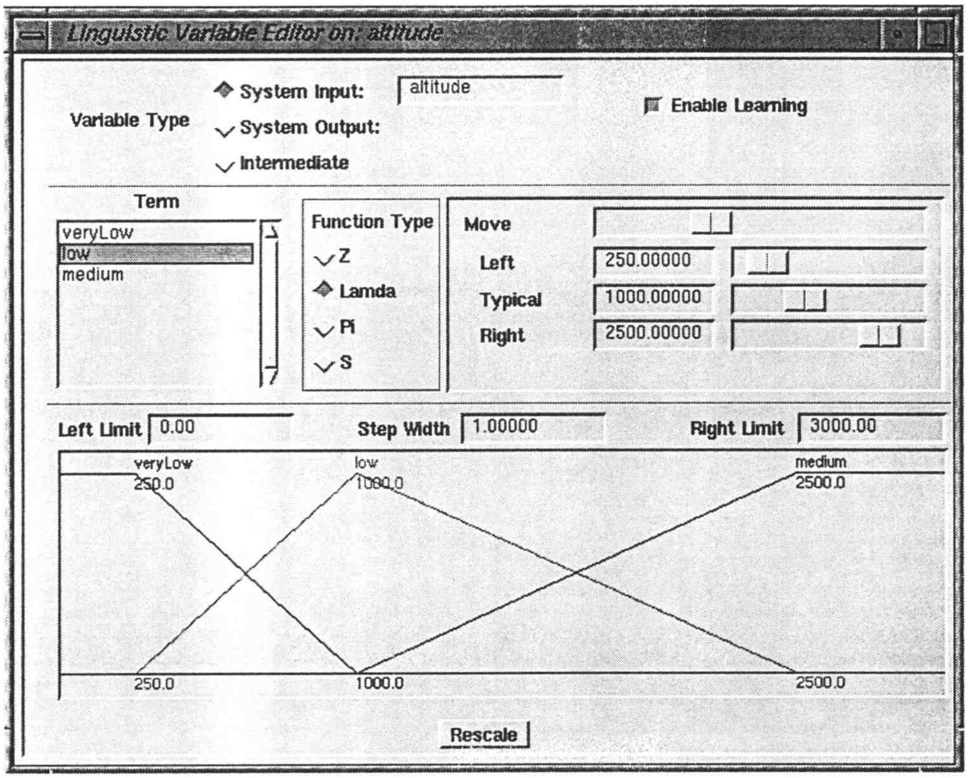
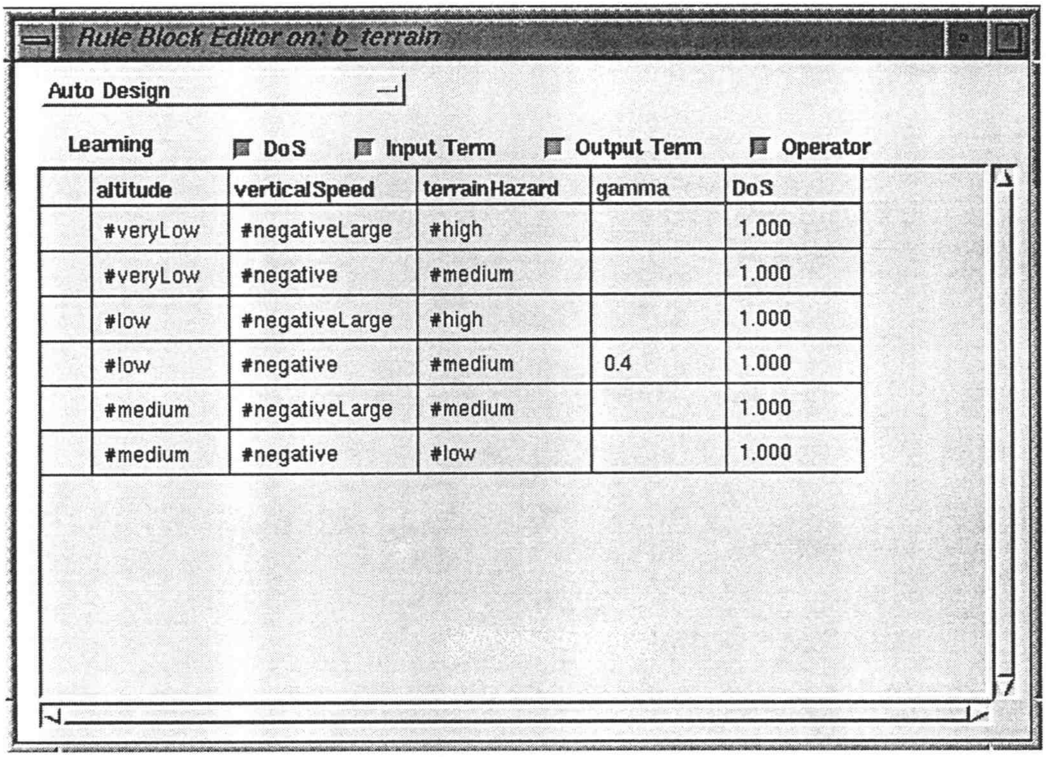


Figure 4.12 Linguistic variable editor

4.4.3 Fuzzy Rule Block Editor

The rule block editor is a tool for manipulating the rules of a rule block. In particular the rule block editor allows the system designer to alter rules, add rules, and delete rules from the rule base. Figure 4.13 shows the rule block editor displaying the rule base of rule block "b_terrain". Rule block inputs are altitude and vertical speed; rule block output is terrain hazard. Each row within figure 4.13 represents a rule. For example the first row translates into the rule "IF altitude is very low and vertical speed is very negative THEN terrain hazard is high". The columns named "gamma" and "DoS" refer to the parameter for the γ-operator, and the degree of support of the rule respectively. If the entry for "gamma" remains empty the aggregation for the corresponding rule is performed with the min-operator, if an entry for "gamma" is made, the aggregation is performed with the γ-operator.



altitude	verticalSpeed	terrainHazard	gamma	DoS
#veryLow	#negativeLarge	#high		1.000
#veryLow	#negative	#medium		1.000
#low	#negativeLarge	#high		1.000
#low	#negative	#medium	0.4	1.000
#medium	#negativeLarge	#medium		1.000
#medium	#negative	#low		1.000

Figure 4.13 Rule block editor

4.4.4 Debugger

The debugger allows the system developer to feed the fuzzy system with input data and obtain the resulting outputs, as well as find the weights of rules and linguistic values of linguistic variables. Figure 4.14 shows the user interface of the debugger. Given the input values 750 ft. for altitude and –3000 ft. per minute for vertical speed, the fuzzy system yields the output value 7.549 for terrain hazard. Information about the state of rule block "b_terrain" is presented in the lower part of the graphical user interface. For instance the following information about the rule "IF altitude is very low and vertical speed is negative large THEN terrain hazard is high" can be obtained: the premise "altitude is very low" is true to degree 0.333, the premise "vertical speed is negative large" is true to degree 0.5. Consequently, the weight of the rule is 0.33.

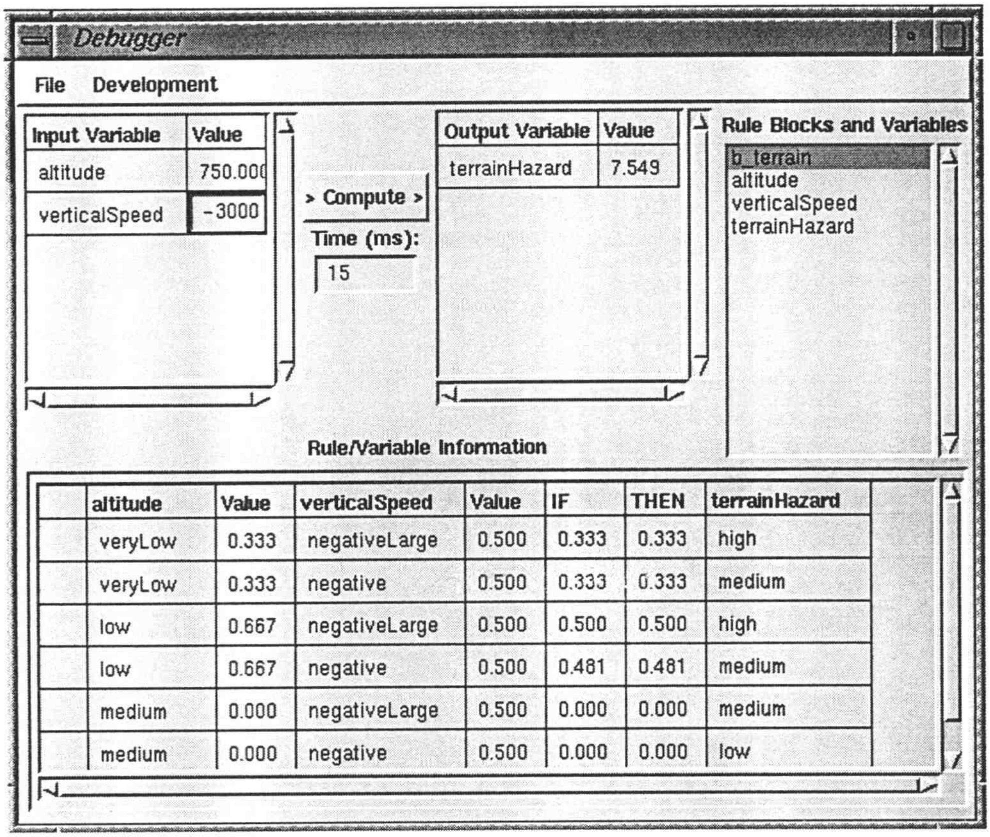


Figure 4.14 Debugger interface

4.5 Fuzzy System Development

Before going into the details of explaining the development process of a fuzzy system, I address the issue of when a fuzzy system should be employed. Generally, if human expert knowledge plays an important role in supervising, monitoring, or controlling the system, fuzzy logic technology is suitable (Munakata and Jani, 1994; Sugeno, 1985). Fuzzy Logic is also commendable for very complex processes, when there is no simple mathematical model. The process of building a fuzzy system starts with the identification of system inputs and system outputs and their ranges. Thereafter, linguistic input and output variables are designed whose terms are associated with membership functions. Selecting the number of terms and the types of membership functions is not a science but rather an art practiced by the system designer. Subsequently, a rule base is formulated and again, the system designer decides the magnitude of the rule base governing the behavior of the system. Finally, the system designer has to verify that some sample system inputs actually yield the expected system outputs and also validate the system as to whether it serves the purpose it was made for. Three methods for developing fuzzy systems are described here.

4.5.1 Human Expert-Based Development

This kind of development is based on human expert knowledge formulated in rules (Munakata and Jani). The expert is interviewed by the system designer and asked to formulate the underlying reasoning process when making decisions or assessments. The observation is that human decision making is often expressed in rules such as "If airspeed is too low then increase thrust". This rule can be easily mapped into a fuzzy rule with the input *airspeed* and output *thrust*. An iterative tuning process may follow optionally.

4.5.2 Iterative Tuning

This approach is characterized by the fact that the system designer devises a system to his best knowledge and thereafter tunes the system in subsequent steps. Tuning involves feeding the fuzzy system with inputs and checking the consistency and correctness of the corresponding outputs. If the results are not satisfactory, changes are made in the rule base and/or membership functions of the linguistic variables to obtain the desired system outputs. When the system behaves satisfactorily, the system designer terminates the tuning iterations.

4.5.3 Adaptive Approach

The adaptive approach can be used if sample data mapping inputs to outputs is available. The adaptive approach is a two step process. First, the designer creates a tentative fuzzy system. Second, an algorithm manipulates the tentative fuzzy system to approximate the sample data.

The fuzzy systems facilitating FFA were developed according to this approach. A Genetic Algorithm (GA) was used to minimize the discrepancy between actual FFA system outputs and desired FFA system outputs based on the sample data obtained from experiments discussed in chapter 6. In a sense, through the application of the GA the fuzzy system "learned" from the examples provided by the sample data.

The sample data for fuzzy system S was a collection of vector pairs. The first vector was the system input; the second vector was the desired system output. In short, sample data was denoted as $D = \{(X_1, Z_1), (X_2, Z_2), \ldots (X_n, Z_n)\}$; where (X_k, Z_k) was the k-th vector pair of D ($k \in \{1, 2, \ldots n\}$). Vector $Z_k = (z_{1,k}, z_{2,k}, \ldots, z_{v,k})$, contained the desired numerical outputs of the output linguistic variables $o_1, o_2, \ldots, o_v$, as response to vector $X_k = (x_{1,k}, x_{2,k}, \ldots, x_{u,k})$, containing the numerical inputs to the input linguistic variables $h_1, h_2, \ldots, h_u$ of fuzzy system S. The GA was used to minimize the function:

$$(4.5) \quad SQE(S,D) = \sum_{k=1}^{n} \left(S(X_k) - Z_k\right)' \left(S(Xk) - Z_k\right)$$

where $S(X_k)$ was the vector of actual numerical outputs of fuzzy system S as response to system inputs X_k. The value of SQE represented the sum of squared error of fuzzy system S based on sample data D.

5. Genetic Algorithms for Fuzzy Systems

GAs are general purpose multi-dimensional optimization algorithms that are modeled after mechanisms of evolution in nature (Takagi, 1993). GAs are defined in terms of population, reproduction, crossover, and mutation (Linkens & Nyongesa). The population is a pool of individuals, represented by their genomes. A genome is a string of genes and is implemented as a binary string. Thus, the genome is a sequence of zeros and ones in which the properties and characteristics of the corresponding individual are encoded. A fitness function specifies each individual's fitness. Reproduction involves selecting two parent genomes from the current population to produce offspring that can then replace members of the old generation. An Offspring is constructed by copying portions of its parents' genome as specified by crossover points. Figure 5.1 illustrates this concept as two crossover points partition the genome of two parents and define what portions of the parent genome is passed on to which offspring.

Parent 1

0	0	1	0	0	0	1	0	0	0

Parent 2

1	1	1	1	1	0	0	1	1	0

Offspring 1

0	0	1	0	1	0	0	0	0	0

Offspring 2

1	1	1	1	0	0	1	1	1	0

Figure 5.1 Genetic crossover operation

Mutation involves inverting randomly selected bits from the affected individual. Inverting a bit is setting the bit to zero if it is one or one if it is zero. Figure 5.2 shows the state of the individual's genome before and after mutation. The shaded areas mark the mutated genes.

Before Mutation

0	0	1	1	1	1	1	1	0	0

After Mutation

0	1	1	1	0	1	1	1	0	0

Figure 5.2 Genetic mutation operation

A GA is easily described in pseudo code:

initialize a random population P
while termination condition is not fulfilled do:
    produce offspring O from certain members of P
    select members of O to replace members of P
    mutate population P

The following sections deal with the implementation of a particular GA used in this study to train fuzzy systems.

5.1 Population

Given the initial fuzzy system S', an initial population P of size n is derived as follows:

Create genome $G(S')$ corresponding to fuzzy system S'
Set $P = \{G(S')\}$
for $j = 1$ to n-1 do:

Set G_j = copy of $G(S')$

mutate G_j at rate r

Set $P = P \cup \{G_j\}$

After n-1 iterations P = $\{G_1, G_2, \ldots, G_n\}$ and $n = |P|$.

The fitness value of an individual G_k is $F(G_k) = \dfrac{1}{SQE(FS(G_k), D)}$, where $FS(G_k)$, or short S_k, is the fuzzy system represented by its genome G_k. Note that $SQE(S_k, D)$ is the sum of squared error defined in equation 4.5 and quantifies how well S_k fits the sample data D. The higher $F(G_k)$, the better does S_k fit sample data D. The following section explains how a genome G is derived from a fuzzy system S, and how a fuzzy system S is derived from a genome G for fitness function evaluation.

A fuzzy system is encoded into a genome by encoding its linguistic variables and rules. New fuzzy systems are constructed using instructions encoded in modified genomes according to a coding scheme common to all individuals. Within population P and across all generations, the lengths of all individuals' genomes are the same.

The initial fuzzy system S' defines static parameters for fuzzy system structure, linguistic variables and rule blocks. Static parameters for fuzzy system structure are: 1. number of linguistic variables; 2: number of rule blocks; 3: input-output relationships between linguistic variables and rule blocks. Static parameters for linguistic variables are: 1. variable type (input, output, intermediate); 2. variable name; 3. name of each linguistic term; 4. universe of discourse of each membership function. Static parameters for rule blocks are: 1. number of rules; 2. names of block input and block output variables. In other words, the GA may only modify the shape of membership functions, premises of rules, the choice of aggregation operators, and the setting of the degree of support. The GA does not manipulate the overall structure of the fuzzy system by any means.

The genome of a fuzzy system S' as defined in section 4.3.5 is the composition of genes of variables contained in set $V' = \{l'_1, l'_2, \ldots, l'_h\}$ and genes of rule blocks contained in queue $C' = \{B'_1, B'_2, \ldots, B'_w\}$. So, the genome of S' is:

$G(S') = \sum_{i=1}^{h} G(l'_i) + \sum_{q=1}^{w} G(B'_q)$, where "plus" means the concatenation of binary strings. $|G(S')|$ is the size of $G(S')$ and equals the number of binary digits $G(S')$ is composed of. A new fuzzy system S can be constructed from any genome G, if $|G| = |G(S')|$, by decoding the genes corresponding to linguistic variables and rule blocks according to the coding scheme provided by S'.

In particular, G is partitioned into: $G = \sum_{i=1}^{h} g_i + \sum_{q=1}^{w} b_q$, where $|g_i| = |G(l'_i)|$ and $|b_q| = G(B'_q)$. Then, g_i and b_q represent gene strings containing instructions on how to construct new versions of the initial linguistic variable l'_i, and the initial rule block B'_q respectively. The new versions of l'_i and B'_q are lv_i and B_q respectively. The resulting new fuzzy system S is defined by its variables $V = \{lv_1, lv_2, \ldots, lv_h\}$ and rule blocks $C = \{B_1, B_2, \ldots, B_w\}$.

A linguistic variable l'_i is encoded by encoding its membership functions. Given a queue of membership functions $L' = \{f'_1, f'_2, \ldots, f'_n\}$, fulfilling the constraint: $\forall (f'_p, f'_r) \in \{(f'_p, f'_r) \in L' \times L' \mid p \neq r\} : t'_p < t'_r$, where t'_x, $x = 1, 2, \ldots, n$, is the t-parameter of membership function $f'_x : [l,r] \rightarrow [0,1]$ explained in section 4.3.2, I define the following parameters:

$$d'_x = \frac{t'_x - l}{l - r}$$

$$b'_x = \frac{e'_x - t'_x}{r'_x - t'_x}$$

$$p'_i = \begin{Bmatrix} 0 & \text{, type of } f'_i \text{ is } \lambda \\ 1 & \text{, type of } f'_i \text{ is } \pi \end{Bmatrix}$$

$$a'_1 = \frac{l'_1 - l}{t'_1 - l}$$

$$p'_1 = \begin{cases} 0 & \text{, type of } f'_1 \text{ is } \lambda \\ 1 & \text{, type of } f'_1 \text{ is } \pi \\ 2 & \text{, type of } f'_1 \text{ is } z \end{cases}, \; i = 2, 3, \ldots, n\text{-}1$$

$$c'_n = \frac{r'_n - t'_n}{r - t'_n}$$

$$p'_n = \begin{cases} 0 & \text{, type of } f'_n \text{ is } \lambda \\ 1 & \text{, type of } f'_n \text{ is } \pi \\ 3 & \text{, type of } f'_n \text{ is } z \end{cases}$$

where, e'_x, l'_x, and r'_x are e-, l-, and r-parameters respectively of membership function f'_x as described in section 4.3.2. The genome $G(l'_i)$ is the composition of the parameters $d'_1, d'_2, d'_3, \ldots, d'_n, a'_1, b'_1, p'_1, b'_2, p'_2, b'_3, p'_3, \ldots, b'_{n-1}, t'_{n-1}, c'_n, b'_n, p'_n$ in binary form. Each parameter is encoded into a fixed length binary code. Table 5.1 summarizes the digit lengths of binary codes for linguistic variable related parameters.

Table 5.1 Lengths of linguistic variable related codes

Parameter	a'_1	p'_1	d'_x	b'_x	p'_i	p'_n	c'_n
Binary code length	10	2	10	10	1	2	10

A new linguistic variable lv_i with n membership functions defined on the universe of discourse $X = [l, r]$ is constructed corresponding to the parameters $d_1, d_2, d_3, \ldots, d_n, a_1, b_1, p_1, b_2, p_2, b_3, p_3, \ldots, b_{n-1}, t_{n-1}, c_n, b_n, p_n$ decoded from gene string g_i as follows:

1. Set t-parameters $t_x = d_x \cdot (l - r) + l$, $x = 1, 2, \ldots, n$ and sort them in increasing order. The sorted collection of t-parameters is $T = \{t_{1'}, t_{2'}, \ldots, t_{n'}\}$.

2. If $p_1 \neq 2$ then set $l_1 = a_1 \cdot (t_1 - l) + l$. Set $r_1 = t_{2'}$. If $p_1 = 1$ then set $e_1 = b_1 \cdot (r_1 - t_{1'}) + t_{1'}$.

   Associate $f_1 = \begin{cases} \lambda \text{ membership function}(l_1, t_{1'}, r_1) & , p_1 = 0 \\ \pi \text{ membership function}(l_1, t_{1'}, e_1, r_1) & , p_1 = 1 \\ z \text{ membership function}(t_{1'}, r_1) & , p_1 = 2 \end{cases}$ with the corresponding linguistic term of membership function f'_1.

3. For $i = 2, 3, \ldots, n\text{-}1$: Set $l_i = t_{(i-1)'}$. Set $r_i = t_{(i+1)'}$. If $p_i = 1$ then set $e_i = b_i \cdot (r_i - t_{i'}) + t_{i'}$. Associate $f_i = \begin{cases} \lambda \text{ membership function}(l_i, t_{i'}, r_i) & , p_i = 0 \\ \pi \text{ membership function}(l_i, t_{i'}, e_i, r_i) & , p_i = 1 \end{cases}$ with the corresponding linguistic term of membership function f'_i.

4. If $p_n \neq 3$ then set $r_n = c_n \cdot (r - t_{n'}) + t_{n'}$. Set $l_n = t_{(n-1)'}$. If $p_n = 1$ then set $e_n = b_n \cdot (r_n - t_{n'}) + t_{i'}$.

   Associate $f_n = \begin{cases} \lambda \text{ membership function}(l_n, t_{n'}, r_n) & , p_n = 0 \\ \pi \text{ membership function}(l_n, t_{n'}, e_n, r_n) & , p_n = 1 \\ s \text{ membership function}(l_n, t_{n'}) & , p_n = 3 \end{cases}$ with the corresponding linguistic term of membership function f'_n.

The genome of rule block $B'_q = \{R'_1, R'_2, \ldots, R'_n\}$ is obtained by concatenating the gene strings of its rules. Therefore, $G(B'_q) = \sum_{i=1}^{n} G(R'_i)$. Thus, $b_q = \sum_{i=1}^{n} r_i$ is a concatination of sub-gene strings, where $|r_i| = |G(R'_i)|$. Rule R'_i within rule block B'_q is of the form (section 4.3.4):

$R'_i = (d'_i, a''_i)$ IF (l'_1 is $t'_{i,1}$) and (l'_2 is $t'_{i,2}$) and ... (l'_m is $t'_{i,m}$) THEN (l'_{m+1} is $t'_{i,m+1}$), (l'_{m+2} is $t'_{i,m+2}$), ..., (l'_{m+z} is $t'_{i,m+z}$)

The genome G(R'_i) is the composition of the parameters $g_{l_1}(t'_{i,1}), g_{l_2}(t'_{i,2}), \ldots, g_{l_m}(t'_{i,m}), g_{l_{m+1}}(t'_{i,m+1}), g_{l_{m+2}}(t'_{i,m+2}), \ldots, g_{l_{m+z}}(t'_{i,m+z}), d'_i, \gamma'_i, b'_i$ in binary form. Note that $g_{l_j}(t'_{i,j})$ is function maping term $t'_{i,j}$ of linguistic variable l_j to an index, as described in section 4.3.3 and

$$\gamma'_i = \begin{cases} \gamma \text{ parameter of } a'^i & \text{,if } a'^i \text{ is } \gamma \text{ operator} \\ 0 & \text{,if } a'^i \text{ is min operator} \end{cases},$$

$$b'_i = \begin{cases} 1 & \text{,if } a'^i \text{ is } \gamma \text{ operator} \\ 0 & \text{,if } a'^i \text{ is min operator} \end{cases}$$

Table 5.2 summarizes the digit lengths of binary codes for rule related parameters. $|T(l'_j)|$ is the number of terms variable l'_j is decomposed into.

Table 5.2 Lengths of rule related codes

Parameter	$t'_{i,j}$	d'_i	γ'_i	b'_i
Binary code length	$\left\lceil \frac{\ln(\lvert T(l'_j)\rvert + 1)}{\ln 2} \right\rceil$	7	7	1

A new rule block B_q with n rules is constructed corresponding to the gene string $b_q = \sum_{i=1}^{n} r_i$. Each sub-gene string r_i is used to construct a new rule R_i. The new rule bock is defined as $B_q = \{R_1, R_2, \ldots, R_n\}$. A new rule R_i is constructed according to the parameters $i_{i,1}, i_{i,2}, \ldots, i_{i,m}, i_{i,m+1}, i_{i,m+2}, \ldots, i_{i,m+z}, d_i, \gamma_i, b_i$ decoded from gene string r_i as follows:

Set $t_{i,j} = g^{-1}{}_{lj}(i_{i,j})$, j = 1, 2, …, $m+z$, where $i_{i,j}$ is the index being mapped to $t_{i,j}$.

If $b_i = 1$ then set $a^i = \gamma_{\gamma i}$-operator.

If $b_i = 0$ then set $a^i = min$-operator.

Set $R_i = (d_i, a^i)$ IF (lv_1 is $t_{i,1}$) and (lv_2 is $t_{i,2}$) and ... (lv_m is $t_{i,m}$) THEN (lv_{m+1} is $t_{i,m+1}$), (lv_{m+2} is $t_{i,m+2}$), …, (lv_{m+z} is $t_{i,m+z}$)

Note that lv_j is the new linguistic variable corresponding to l'_j.

5.2 Reproduction

Reproduction involves selecting parent genomes from the current population P and creating offspring by copying portions of parent genes. The resulting offspring replaces members of the old population and a new generation of individuals is formed. The probability for an individual G_k to be selected as parent is $p(G_k) = \frac{F(G_k)}{\sum_{i=1}^{n} F(G_i)}$.

The higher an individual's fitness value the more likely it is selected as parent. The reproduction method employed is called "selective breeding" and works as follows (Linkens & Nyongesa, 1995):

P = given initial population

P' = next generation population

O = set containing offspring

A_1 = genome of parent 1

A_2 = genome of parent 2

G_1 = genome of offspring 1

G_2 = genome of offspring 2

initialize $O = \phi$

repeat $\left\lceil \frac{|P|}{2} \right\rceil$ times:

select parent A_1

select parent A_2

create two children G_1 and G_2 by performing crossover operation with A_1 and A_2

Set $O = O \cup \{G_1, G_2\}$

set $P' =$ best $|P|$ individuals from $O \cup P$

5.3 Crossover

The crossover points partition the genome of the parents selected for reproduction and determine which part of the parents genome is passed on the which offspring. The crossover operator returns the genome of two children (G_1, G_2). It works as follows:

k	=	given number of crossover points
A_1	=	given genome of parent 1
A_2	=	given genome of parent 2
l	=	given length of individual's genome
G_1	=	genome of offspring 1
G_2	=	genome of offspring 2
C	=	set/collection of crossover points

initialize $C = \phi$, $G_1 = \phi$, $G_2 = \phi$

k times repeat:

    create random integer number c between 1 and l-1

    set $C = C \cup \{c\}$ (C never contains duplicate elements)

set $C = C \cup \{0, l\}$

set $C = C$ sorted in ascending order according to element's magnitude = $\{c_1, c_2, \ldots, c_p\}$

for j = 1 to p-1 do:

    if j is odd then

    $G_1 = G_1$ + (copy of A_1 between $c_j + 1$ and c_{j+1})

    $G2 = G_2$ + (copy of A_2 between $c_j + 1$ and c_{j+1})

    else

    $G_1 = G_1$ + (copy of A_2 between $c_j + 1$ and c_{j+1})

    $G2 = G_2$ + (copy of A_1 between $c_j + 1$ and c_{j+1})

So, G_1 and G_2 are the resulting genomes of the crossover operation performed on A_1 and A_2.

5.4 Mutation

Mutation involves inverting a number of genes. Inverting is changing 0 to 1 and 1 to 0. The algorithm works as follows given r, the mutation rate, and G, the genome to mutate:

$\lceil r\,|G| \rceil$ times repeat:

invert a randomly selected gene within G

5.5 Termination Condition and Quality of Resulting Fuzzy System

The questions of when to terminate the GA and how to validate the quality of the resulting fuzzy system are addressed in this section. The GA terminates after a certain number of generations. At each generation the fittest and median fit individual of the current population are determined to be the one with the highest fitness and the median fitness value respectively. When the GA terminates, the individual that best fits the sample data is the fittest individual of the latest generation. It is found in the latest population since the fittest members of the population always survive and are passed on to the next generation. How is the quality of the fittest individual validated? Minimizing the sum of squared errors on the sample data does not mean that the error on new cases is minimized. In fact, the longer the fuzzy system is trained the better it fits the data it is being trained on. Ultimately, it will "memorize" the data almost exactly. However, by doing so the fuzzy system may lose its generality to fit new data. To counteract the effect of overfitting, the sample data is divided into sets of training data, overtraining test data, and validation test data. Instead of training the GA on the whole sample data set, the GA trains the fuzzy system on the training data only while the error with regards to the overtraining test data is a predictor of how well the fittest individual fits new data. The fittest individual at the generation at which the error with regards to the overtraining test data is the smallest is the best guess of an individual likely to fit new data well. This individual is denoted as *Best*. The quality of *Best* is then evaluated according to the validation test data. The lower $SQE(FS(Best), VD)$, where VD is the validation test data set, the better the quality of the fuzzy system $FS(Best)$.

The GA determines $Best$ as follows:

OD	=	Overtraining test data
j	=	generation
$maxGen$	=	maximum number of generations
P_j	=	population at generation j
G_j	=	fittest individual of population Pj

set P_0 = randomly generated population

set $Best = G_0$

for j = 1 to maxGen - 1 do:

create population P_j

if $SQE(FS(G_j),OD) < SQE(FS(Best),OD)$ then set $Best = G_j$

5.6 Genetic Algorithm Tool

A tool providing the interface for specifying parameters of an underlying genetic algorithm for fuzzy system manipulation is discussed in this section. The tool allows the user to input the population size, number of crossover points, an initial mutation rate for creation of the initial population, a regular mutation rate, the number of generations to be run, and the file name that contains the training data. Optionally, the name of a file containing the overtraining test data to check for overfitting can be input as well. The current generation number is provided along with the time since algorithm activation. Figure 5.3 shows the interface of the tool. In this example, the example fuzzy system composed of the rule block of table 4.1 and linguistic variables of figures 4.6, 4.7, and 4.8 undergoes training with regards to data contained in the file "terrainTraining". The overtaining test data is stored in the file "terrainOvertraining". The population size, the number of crossover points, the initial mutation rate, the regular mutation rate, and the maximum number of generations are set to 50, 5, 0.2, 0.01, and 100 respectively. It takes 0.318 hours to complete 100 generations.

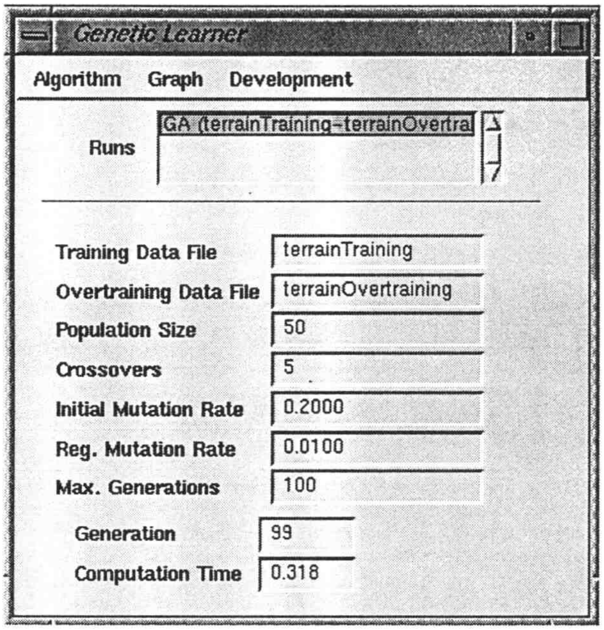


Figure 5.3 Genetic algorithm tool

Figure 5.4 represents a graphical tool showing the mean of sum of squared error (MSE) of the fittest and the median individual with regards to the training data and the MSE of the fittest with regards to the overtraining test data. Overfitting is reached around generation 10 from which onwards the MSE of the fittest regarding the overtraining test data set is increasing. Around generation 35 the population is converging as the median and the fittest individual have about the same MSE regarding the training data.

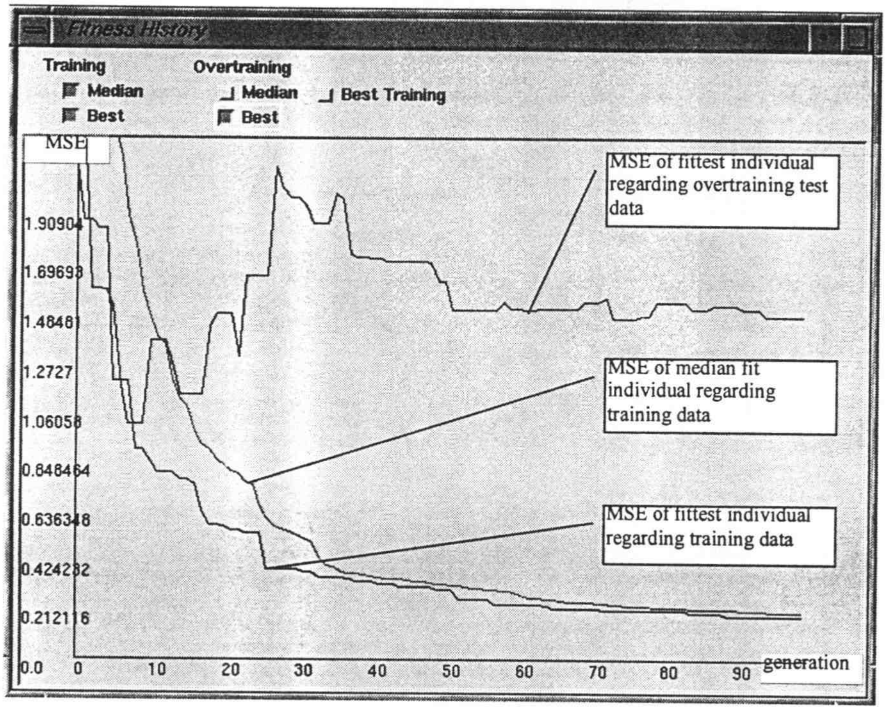


Figure 5.4 Graphical tool showing MSE over generation

At run time of the GA, the user can open a fuzzy system editor on the best fuzzy system found so far. Figure 5.5 shows the optimized rule block “b_terrain”.

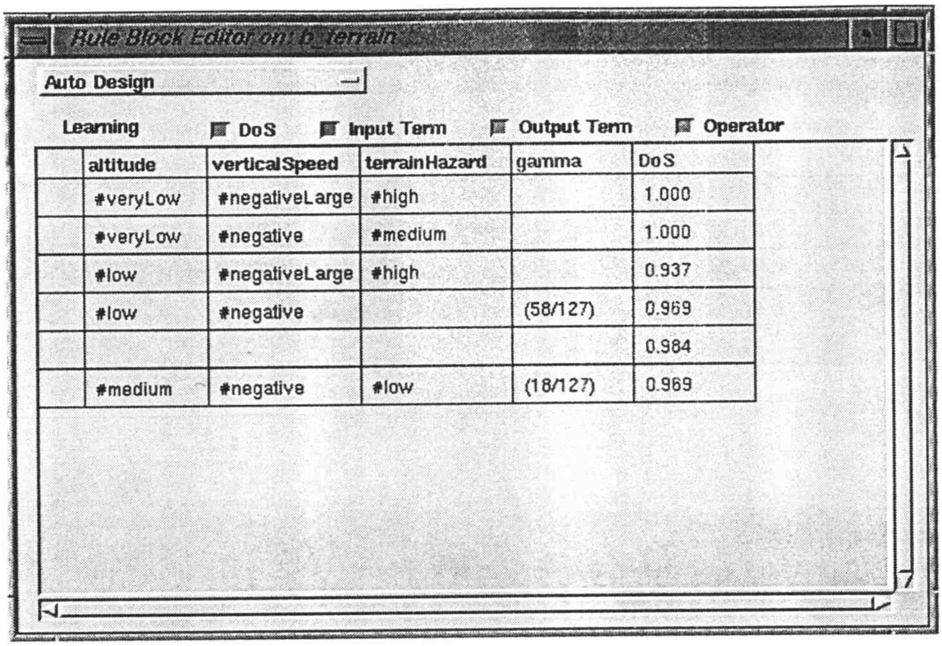


Figure 5.5 Rule block editor showing modified rule base

6. Fuzzy Systems for FFA

This chapter deals with the implementation of FFA for the *capture altitude* function. A thorough explanation of FFA is given in section 6.1. As stated earlier, fuzzy systems, denoted as FFA systems, were used to facilitate FFA and are introduced in section 6.2. Calibrating FFA to emulate human expert assessment was arranged by training initial FFA systems through the application of a GA as described in chapter 5. The data required for training the FFA systems was extracted from real aviation experts. This knowledge acquisition process was conducted by letting human experts watch simulated altitude capture functions and rate how well the functions were performed; this is the topic of section 6.3. Section 6.4 discusses the training process of the initial FFA systems in more detail. I proceed with a more formal explanation of how FFA for the *capture altitude* function works.

6.1 Functioning of FFA

FFA for the *capture altitude* function is a process that continuously outputs a vector NO indicating how well a *capture altitude* function, say I, characterized by its target altitude a is being performed. When function I is created by declaring target altitude a, state information of the airplane in form of a state vector is collected periodically, every d milliseconds, but only the most recent x vectors are retained in memory $M = \{V_1, V_2, V_3, \ldots V_x\}$. Thus M describes the history of the flight during the last $x \cdot d$ milliseconds. Vector $V_j \in M$ is of the form:

(6.1) $V_j = (s_{j,1}, s_{j,2}, s_{j,3}, s_{j,4}, s_{j,5})$;

where $s_{j,1}$ is the altitude, $s_{j,2}$ is the indicated airspeed, $s_{j,3}$ is the vertical speed, $s_{j,4}$ is the flight path angle, and $s_{j,5}$ is the thrust setting representing the state of the airplane $(x - j) \cdot d$ milliseconds ago. A function H transforms memory M to vector X as follows:

$$(6.2) \qquad H(M) = X = \begin{pmatrix} x_1 = & \frac{2}{d \cdot n \cdot (n-1)} \cdot \sum_{j=1}^{n-1} \left(s_{x-j-1,3} - s_{x-j,3}\right) \cdot (n-j) \\ x_2 = & s_{x,3} \\ x_3 = & s_{x,1} \\ x_4 = & s_{x,1} - a \\ x_5 = & s_{x,5} \\ x_6 = & s_{x,4} \\ x_7 = & s_{x,2} \\ x_8 = & \frac{2}{d \cdot n \cdot (n-1)} \cdot \sum_{j=1}^{n-1} \left(s_{x-j+1,2} - s_{x-j,2}\right) \cdot (n-j) \\ x_9 = & s_{1,1} - a \\ x_{10} = & s_{1,3} \end{pmatrix}$$

Table 6.1 explains the meaning of the variables. Weighted averages were calculated for acceleration measures in order to reduce noise.

Table 6.1 Input variables

Variable	**Meaning**	**Explanation**	**Dimension**
x_1	current vertical speed acceleration	calculated as the weighted average of the last n-1 vertical speed rate of change values of memory M; where the most current, second most current, ..., and the (n-1)th most current vertical speed rate of change value received a weight of n-1, (n-2), ..., and 1 respectively	feet per minute per second
x_2	current vertical speed	latest recorded vertical speed value in memory M	feet per minute
x_3	current altitude	latest recorded altitude value in memory M	feet
x_4	current altitude error	calculated as current altitude – target altitude	feet
x_5	current thrust	latest recorded thrust value in memory M	fraction of maximum thrust
x_6	current flight path angle	latest recorded flight path angle recorded in memory M	degrees
x_7	current airspeed	latest recorded airspeed value in memory M	knots
x_8	current airspeed acceleration	calculated as the weighted average of the last n-1 airspeed rate of change values of memory M; where the most current, second most current, ..., and the (n-1)th most current airspeed rate of change value received a weight of n-1, (n-2), ..., and 1 respectively	knots per minute per second
x_9	past altitude error	calculated as earliest recorded altitude in memory M – target altitude	feet
x_{10}	past vertical speed	earliest recorded vertical speed in memory M	feet per minute

The resulting vector X is comprised of variables functioning as numerical input variables of fuzzy systems outputting vector NO, the result of FFA. Vector NO is:

$$NO = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

where the variables y_j ($j = 1,2,\ldots,4$) are defined as described in table 6.2.

Table 6.2 Output variables

Variable	**Meaning**	**Value Range**	**Explanation**
y_1	overspeed hazard	[0, 10]	The variables quantify the extent to which the corresponding hazard is present. The extent ranges from 0 to 10. At 0 no hazard is present while at value 10 the hazard is fully developed. The higher the value the more developed the hazard.
y_2	stall hazard	[0, 10]	
y_3	terrain hazard	[0, 10]	
y_4	function performance	[0, 10]	The variable quantifies how well the function is being performed considering safety, and compliance to the specified target altitude.

For this study, the variables x, d, and n were set to 150, 50, and 11 respectively. Therefore, memory was confined to 7.5 seconds worth of flying time, data was collected in real-time at 20 Hz, and acceleration measures aggregated 0.55 seconds worth of flying time. While basing function assessment on just 7.5 seconds of observed flight is controversial the reader should note that the length was chosen to be rather short for the following reason. As the number of flight scenes to be played to the human experts was to be maximized in order to collect as much sample data as possible and time was scarce, the flight scene length had to be kept as short as reasonably possible.

The initial fuzzy systems used to compute overspeed-, stall-, and terrain hazard as well as function performance are discussed in the next section.

6.2 Initial FFA Systems

As the input and output variables of the FFA systems are identified, the next step is to explain properties and the architecture of FFA systems in this section. The FFA systems were designed as a combination of fuzzy hazard systems assessing the level of emergency for different hazards, and a fuzzy system integrating hazards and level of goal compliance to compute the function performance. As described before, the fuzzy hazard systems output a value between 0 and 10. A high output value was interpreted as a highly developed hazard. Function performance reflected how well the airplane was being operated considering safety constraints and goal compliance. The greater the value the better the performance. A detailed description of the fuzzy systems regarding definition of linguistic variables and rules is included in appendix A.

Using fuzzy logic as the underlying technology was justified because it provided the flexibility to integrate many sensor readings to assess hazards or evaluate function performance in real time at any given situation. A very important feature of these fuzzy systems was that they represented knowledge captured in comprehensible if-then rules. The FFA systems described in this thesis were envisioned to become the underlying data source for warning and alerting devices indicating a degree of emergency. Section 6.3 discusses how the initial FFA systems were trained on the average assessments taken from an expert panel.

6.2.1 Fuzzy Terrain Hazard System

The function of the fuzzy terrain hazard system (FTHS) is similar to the logic of a Ground Proximity Warning System (GPWS) alerting the flight crew to potential terrain-related hazards. The GPWS checks for hazards such as excessive descent rate and unsafe terrain clearance (Boeing, 1988).

Two alternatives were under consideration. Fuzzy Terrain Hazard System 1 (FTHS 1) determined the terrain hazard based on current altitude and current vertical speed only.

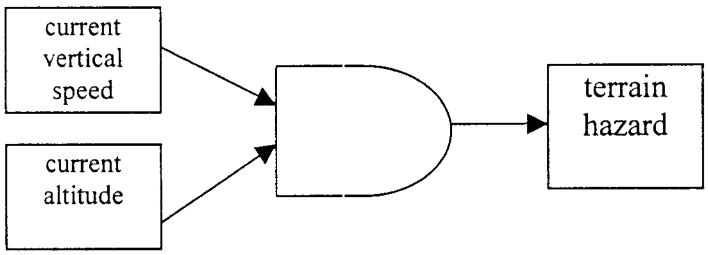


Figure 6.1 FTHS 1 architecture

Figure 6.1 represents the architecture of FTHS 1. A rule block computed the output terrain hazard given the inputs current vertical speed and current altitude. The rule base containing 32 rules reasoned that with increasingly negative current vertical speed and decreasing current altitude, the terrain hazard increased (see appendix table A.5).

Fuzzy Terrain Hazard System 2 (FTHS 2) determined the terrain hazard based on current altitude, current flight path angle, current airspeed, current thrust, and current altitude error. One rule block computed the intermediate variable attitude describing the vertical direction of the airplane in linguistic terms by integrating current thrust, current flight path angle, and current airspeed. Terrain hazard was the result of combining attitude, current altitude error, and current altitude. Figure 6.2 represents the structure of (FTHS 2). The knowledge base containing 224 rules in appendix table A.6 worked the following way: the more extreme the downward trend of the airplane expressed by attitude, and the lower the current altitude, the higher was the terrain hazard. However, if the attitude indicated an extreme upward trend at low airspeed, a stall condition might have been present. A stall at a low altitude implied a terrain hazard since a recovery maneuver would call for pitching down the airplane. Current altitude error in combination with attitude indicated if the airplane was going in the right direction. If the goal was to climb, but the airplane descended, then the terrain hazard was considered to be more developed.

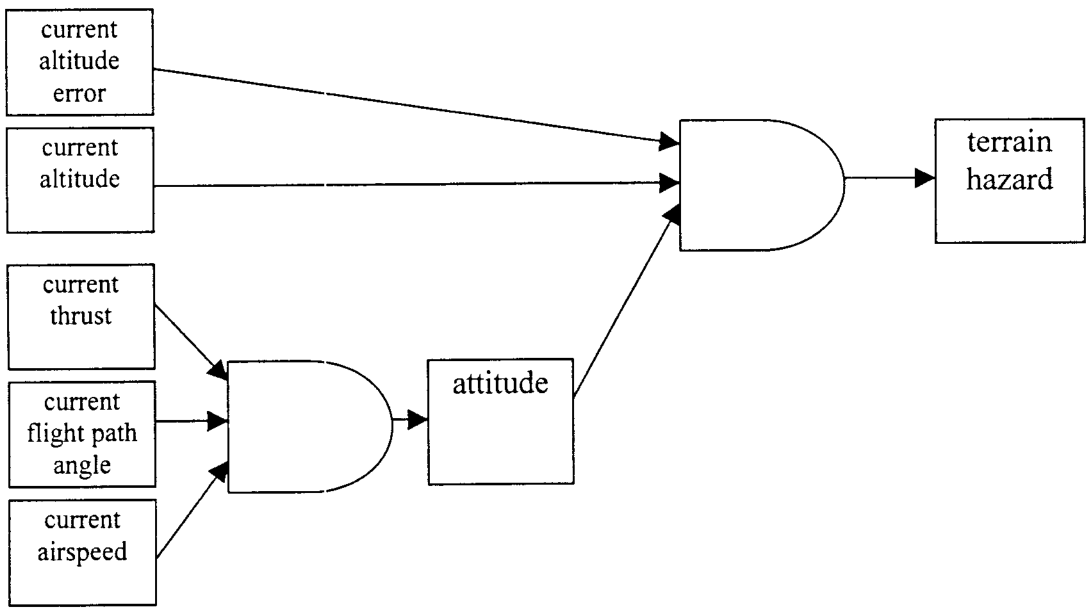


Figure 6.2 FTHS 2 architecture

6.2.2 Fuzzy Stall Hazard System

The fuzzy stall hazard system (FSHS) is similar to the logic that activates the stick shaker, a warning system to identify potential stall hazards. Two different fuzzy systems, one based on current flight path angle and current airspeed, the other based on current flight path angle, current airspeed, current airspeed acceleration, and current thrust, were explored. Figure 6.3 and 6.4 the knowledge base architecture of Fuzzy Stall Hazard System 1 (FSHS 1) and 2 (FSHS 2) respectively. A stall is caused by excessive angle of attack that is usually accompanied by low airspeed. Due to limitations in the flight simulator software the decision logic of both systems (appendix table A.3 and A.4) was based on lack of current airspeed, a symptom, and not excessive angle of attack, the true cause. The logic of both systems reasoned that with increasing current flight path angle and decreasing current airspeed/future airspeed, the stall hazard increased. While FSHS 1, consisting of 32 rules, used current airspeed, FSHS 2, consisting of 104 rules, made a prediction about the future airspeed based on current airspeed, current airspeed acceleration, and current thrust.

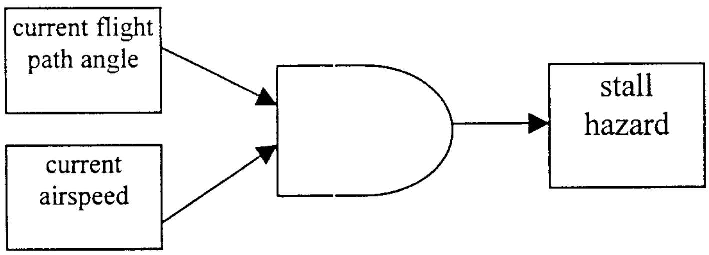


Figure 6.3 FSHS 1 architecture

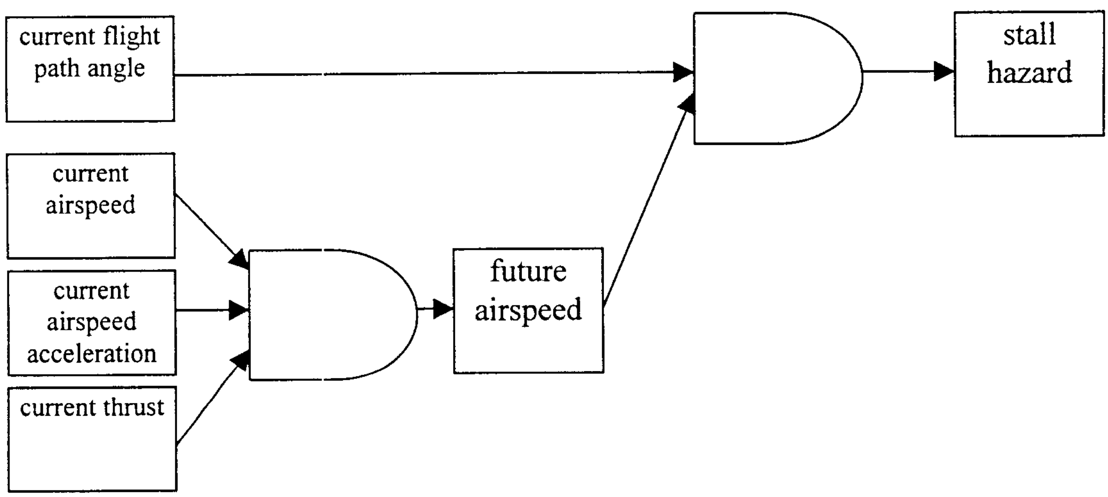


Figure 6.4 FSHS 2 architecture

6.2.3 Fuzzy Overspeed Hazard System

The function of the desired fuzzy system is comparable with that of the logic controlling the "clacker", a warning system for excessive airspeed. The logic controlling the "clacker" of a Boeing 757 simply checks if the current airspeed is beyond the maximum allowable indicated airspeed (Boeing, 1988). Two fuzzy systems were under consideration. Fuzzy Overspeed Hazard System 1 (FOHS 1) tracked current airspeed and current airspeed acceleration while Fuzzy Overspeed Hazard System 2 (FOHS 2) tracked current airspeed, current airspeed acceleration, current thrust, and current flight path angle. Figure 6.5 and 6.6 represent the knowledge base architecture of the respective fuzzy systems. FOHS 1, consisting of 24 rules, reasoned that with increasing current airspeed and current airspeed acceleration the overspeed hazard increased (see appendix table A.1). After predicting the future airspeed, FOHS 2, consisting of 104 rules, reasoned that with increasing future airspeed and decreasing current flight path angle the overspeed hazard increased (see appendix table A.2).

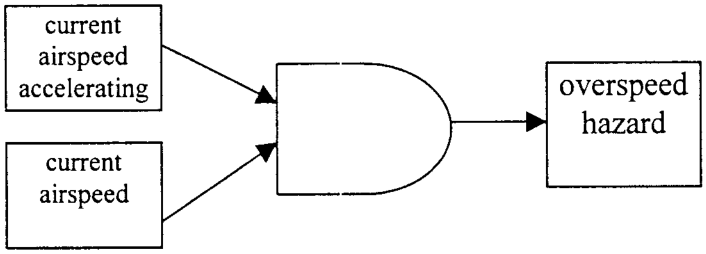


Figure 6.5 FOHS 1 architecture

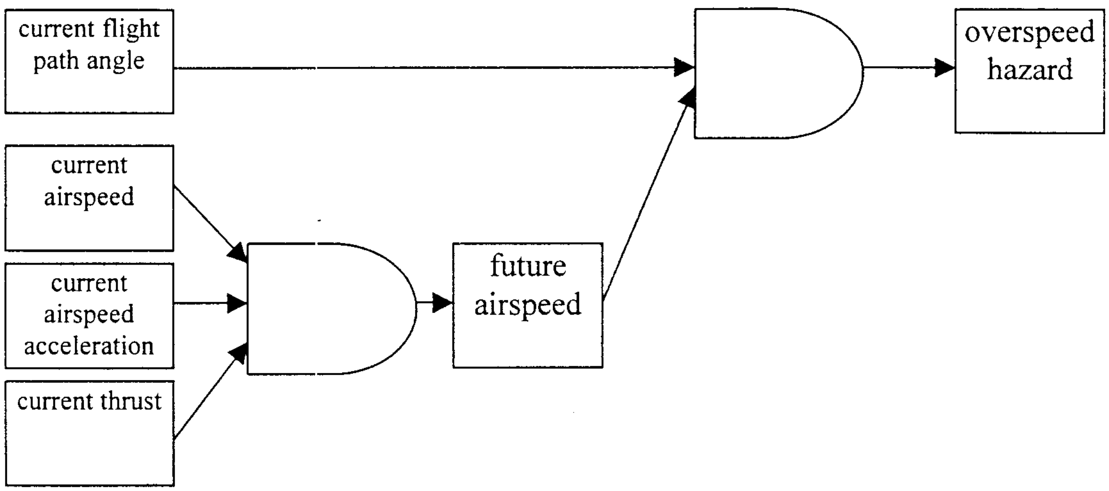


Figure 6.6 FOHS 2 architecture

6.2.4 Fuzzy Hazard and Error Integration System

The fuzzy hazard and error integration system (FHEIS) combined the output of three hazard systems with measures indicating how well the specified goal was being pursued. Two error measures called past error and current error were used in combination to quantify if progress was being made towards achieving the goal. Past error represented the distance between the past state to the goal while current error represented the distance between the current state and the goal. Therefore, if the past error was positive large and the current error was positive small, then it was inferred that airplane was heading in the right direction and thus was in compliance with the goal. Figure 6.7 shows how the hazard system outputs were used to compute the overall hazard. The latter and the error measures were combined to yield function performance. The knowledge base, comprised of 496 rules (see appendix table A.7), approximately reasoned as follows. If there was a severe hazard present then function performance was low, if there was none and the airplane was going in the right direction then function performance was high.

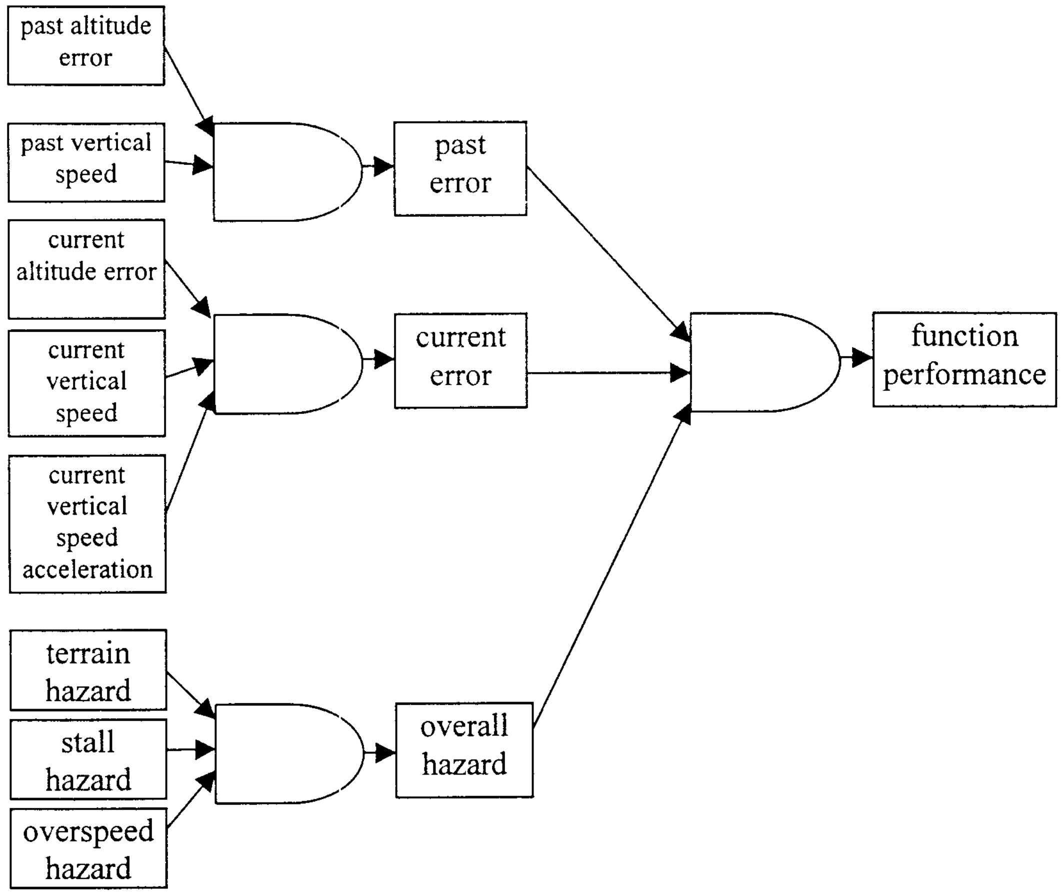


Figure 6.7 FHEIS architecture

6.3 Knowledge Acquisition

This section explains how human expert knowledge was acquired in order to train the initial FFA systems. First, scenarios were developed in which *capture altitude* functions were carried out at various levels of performance. Those scenarios were played to a panel of human experts who then rated how well the functions portrayed in the scenarios were performed. Next, the expert ratings were averaged to yield the average expert assessment given a certain scenario.

In total 104 scenarios were first designed and then executed. Each scenario portrayed a *capture altitude* function for 7.5 seconds. While a scenario was executed, data from the flight simulator, was recorded at 20Hz in form of vectors as defined in equation 6.1. The vectors forming memory were later used to replay the scenarios to the human experts. The recorded memory was transformed into vectors as defined in equation 6.2. Those resulting vectors were saved and associated with the corresponding scenario since they were used later as inputs to the FFA systems during the training process. Appendix table B.1 lists the fuzzy system inputs associated with the scenarios.

After the scenarios had been created human experts were invited to participate in this study. Their job was to rate the scenarios according to the measures defined in table 6.2. The ratings for each measure were averaged across the human experts to yield average measures for each scenario. Thus any scenario, say scenario number k, had a vector pair (X_k, Z_k) associated with it where X_k was an input to the FFA systems and Z_k was the desired output of the FFA systems. Those vector pairs were exactly the sample data necessary for training the FFA system using a GA as described in section 4.5.3.

The following sections explain limitations of the knowledge acquisition process and how the scenarios were designed and executed.

6.3.1 Limitations

Since the initial FFA systems were to be trained to reproduce the average human expert assessment for a number of scenarios, the quality of the resulting fuzzy systems depended on the selection and number of scenarios and human experts. In an ideal situation only the best aviation experts would have been selected and a large number of scenarios would have been created according to the specifications of another set of aviation experts. However, for practical reasons the scenarios were created by the researcher and the selection of human experts did not reflect a panel of best experts.

6.3.2 Scenarios

Scenarios were first designed, then executed and recorded. A scenario was designed by specifying the initial conditions of the altitude capture function such as target altitude, initial altitude, initial airspeed, and initial flight path angle and control settings such as commanded airspeed, and commanded flight path angle. These scenario specifications were executed on a real-time flight simulator modeling a twin jet engine civil transport aircraft. During simulation data was recorded and transformed to yield inputs to the FFA systems. To collect desired output values regarding the FFA systems, the recorded scenarios were played to human experts by using the same flight simulator the scenarios were executed on. Human experts saw how the simulated airplane was operated by following the airplane's state dynamics displayed on an attitude director indicator (ADI) and mode control panel (MCP). The following assumptions were made with regards to the simulated environment and aircraft:

- Stall speed: 120 knots indicated airspeed
- Overspeed: 340 knots indicated airspeed
- No winds
- Ground is level, no mountains
- Ground impact occurs at sea level (0 ft. altitude)

6.3.2.1 Scenario Design

The fuzzy system had to work well in a broad variety of situations. Therefore the scenarios, which the fuzzy system were trained on, were designed to be diverse as well. The fuzzy systems were supposed to assess terrain, stall, and overspeed hazards at various severity levels and rate function performance at various levels correctly. Given the objective of a robust fuzzy system, the scenarios were grouped into terrain hazard scenarios, stall hazard scenarios, overspeed hazard scenarios, and altitude capturing scenarios. Each scenario group was created to yield a wide range of ratings for the metric it was designed for. However, a priori it was hard to predict any rating distribution within the scenario groups. Although great care was taken to randomly generate scenarios within each group, the sample of all scenarios was not a random sample of possible scenarios. Scenario duration was fixed at 7.5 seconds. This relatively short duration was necessary to run 104 scenarios within a 2.5 hour experiment. Each scenario group consisted of 26 randomly generated scenarios.

6.3.2.1.1 Hazard Groups

Hazard scenarios were characterized by a state vector *S* describing the initial state of *capture altitude* function, and a reaction vector *C* describing the control settings determining how well the function was performed. Vector *S* was of the form (*altitude*, *fpa*, *airspeed*, *target*); where *altitude* was the initial altitude, *fpa* was the initial flight path angle, *airspeed* was the initial airspeed, *target* was the target altitude to be captured. Vector *C* was of the form (*cfpa*, *cairspeed*); where *cfpa* was the commanded flight path angle, and *cairspeed* was the commanded airspeed. In the following section the design for each hazard scenario group is discussed in more detail.

The terrain hazard group was expected to obtain a wide range of terrain hazard ratings for its scenarios. Vector *S* as well as vector *C* was randomly generated according to the following scheme:

Variable *altitude* = $r_1 \cdot 100$; where r_1 was a random variable that with probability of 0.5 took on integer values between 5 and 12 and with probability 0.5 took on integer values between 13 and 27. Its distribution was:

$$p(r_1 = x) = \begin{cases} \frac{1}{16} & , x \in \{5, 6, \ldots, 12\} \\ \frac{1}{30} & , x \in \{13, 14, \ldots, 27\} \\ 0 & , \text{otherwise} \end{cases}$$

Variable *fpa* = r_2; where r_2 was a random variable that with probability 0.5 took on real values between -8 and 0 and with probability 0.5 took on real values between 0 and 20. Its density function was:

$$f_{r_2}(x) = \begin{cases} \frac{1}{16} & , -8 \le x \le 0 \\ \frac{1}{40} & , 0 < x \le 20 \\ 0 & , \text{otherwise} \end{cases}$$

Variable *airspeed* = r_3; where r_3 was a random variable taking on integer values between 120 and 320. Its distribution was:

$$p(r_3 = x) = \begin{cases} \frac{1}{201} & , x \in \{120, 121, \ldots, 320\} \\ 0 & , \text{otherwise} \end{cases}$$

Variable *target* = $r_4 \cdot 100$; where r_4 was a random variable that with probability 0.5 took on value 0 and with probability 0.5 took on integer values between 20 and 350. Its distribution was:

$$p(r_4 = x) = \begin{cases} \frac{1}{2} & , x = 0 \\ \frac{1}{662} & , x \in \{20, 21, \ldots, 350\} \\ 0 & , \text{otherwise} \end{cases}$$

Variable *cfpa* = r_5; where r_5 was a random variable that with probability 0.5 took on real values between -8 and -5, with probability 0.25 took on real values between -5 and 5, with probability 0.25 took on real values between 5 and 20. Its density function was:

$$f_{r_5}(x) = \begin{cases} \frac{1}{6} & , -8 \le x \le -5 \\ \frac{1}{40} & , -5 < x \le 5 \\ \frac{1}{60} & , 5 < x \le 20 \\ 0 & , \text{otherwise} \end{cases}$$

Variable *cairspeed* = *airspeed* + r_6; where r_6 was a random variable taking on real values between -30 and 30. Its density function was:

$$f_{r_6}(x) = \begin{cases} \frac{1}{60} & , -30 \le x \le 30 \\ 0 & , \text{otherwise} \end{cases}$$

The scenarios within this group started out at low altitude, below 2,700 ft. The hypothetical pilot reaction was biased towards extreme maneuvers that was either rapid descends or rapid climbs. In order to create low altitude stall conditions that posed a terrain hazard the probability of positive large commanded flight path angle was set to 0.25.

The stall hazard group was expected to obtain a wide range of stall hazard ratings for its scenarios. The randomization scheme for vector S and C are given as follows:

Variable *altitude* = $r_7 \cdot 100$; where r_7 was a random variable that took on integer values between 25 and 350. Its distribution was:

$$p(r_7 = x) = \begin{cases} \frac{1}{326} & , x \in \{25, 26, \ldots, 350\} \\ 0 & , \text{otherwise} \end{cases}$$

Variable *fpa* = r_2. Variable *airspeed* = r_8; where r_8 was a random variable that with probability 0.7 took on integer values between 120 and 150 and with probability 0.3 took on integer values between 145 and 240. Its distribution was:

$$p(r_8 = x) = \begin{cases} \dfrac{7}{310} & , x \in \{120, 121, \ldots, 144\} \\ \dfrac{125}{4712} & , x \in \{145, 146, \ldots, 150\} \\ \dfrac{3}{760} & , x \in \{151, 152, \ldots, 220\} \\ 0 & , \text{otherwise} \end{cases}$$

Variable *target* = $r_9 \cdot 100$; where r_9 was a random variable that took on integer values between 20 and 350. Its distribution was:

$$p(r_9 = x) = \begin{cases} \dfrac{1}{331} & , x \in \{20, 21, \ldots, 350\} \\ 0 & , \text{otherwise} \end{cases}$$

Variable *cfpa* = r_{10}; where r_{10} was a random variable that with probability 0.6 took on real values between 4 and 20 and with probability 0.4 took on real values between -5 and 4. Its density function was:

$$f_{r_{10}}(x) = \begin{cases} \dfrac{2}{45} & , -5 \leq x \leq 4 \\ \dfrac{3}{80} & , 4 < x \leq 20 \\ 0 & , \text{otherwise} \end{cases}$$

Variable *cairspeed* = *airspeed* + r_{11}; where r_{11} was a random variable that with probability 0.7 took on real values between -10 and 0 and with probability 0.3 took on values between 0 and 20. Its density function was:

$$f_{r_{11}}(x) = \begin{cases} \dfrac{7}{100} & , -10 \leq x \leq 0 \\ \dfrac{3}{200} & , 0 < x \leq 20 \\ 0 & , \text{otherwise} \end{cases}$$

These scenarios were biased towards scenarios exhibiting low initial and commanded airspeeds and positive large commanded flight path angles.

The overspeed hazard group was expected to obtain a wide range of overspeed ratings for its scenarios. The randomization scheme for vector S and C are given as follows:

Variable *altitude* $= r_7 \cdot 100$. Variable *fpa* $= r_2$. Variable *airspeed* $= r_{12}$; where r_{12} was a random variable that took on integer values between 280 and 340. Its distribution was:

$$p(r_{12} = x) = \begin{cases} \frac{1}{61} & , x \in \{280, 281, \ldots, 340\} \\ 0 & , \text{otherwise} \end{cases}$$

Variable *target* $= r_9 \cdot 100$. Variable *cfpa* $= r_{13}$; where r_{13} was a random variable that with probability 0.6 took on real values between -8 and 0 and with probability 0.4 took on real values between 0 and 8. Its density function was:

$$f_{r_{13}}(x) = \begin{cases} \frac{3}{40} & , -8 \le x \le 0 \\ \frac{1}{20} & , 0 < x \le 8 \\ 0 & , \text{otherwise} \end{cases}$$

Variable *cairspeed* $=$ *airspeed* $+ r_{14}$; where r_{14} was a random variable that with probability 0.6 took on real values between 0 and 60 and with probability 0.4 took on values between -20 and 0. Its density function was:

$$f_{r_{14}}(x) = \begin{cases} \frac{1}{50} & , -20 \le x \le 0 \\ \frac{1}{100} & , 0 < x \le 60 \\ 0 & , \text{otherwise} \end{cases}$$

Since scenarios starting out a very low airspeed were likely to result in very low overspeed hazard, the scenarios within this group started out at medium to high airspeed. The scenarios were biased towards exhibiting high commanded airspeeds.

6.3.2.1.2 Altitude Capture Group

The altitude capture group contained scenarios in which the difference between initial altitude and target altitude was small enough to actually capture the altitude.

Like hazard scenarios altitude capture scenarios were characterized by state vector *S* describing the initial state of the *capture altitude* function. Vector *C* describing the control settings was defined differently for altitude capture scenarios.

C = (*bfpa*, *cairspeed*, *caltitude*); where *bfpa* was the flight path angle boundary, *cairspeed* was the commanded airspeed, and *caltitude* was the commanded altitude. The flight path angle boundary represented either the maximum flight path angle if the airplane had to climb or the minimum flight path angle if the airplane had to descend to capture the target.

The randomization scheme was as follows:

Variable *altitude* = $r_7 \cdot 100$. Variable *fpa* = r_2. Variable *airspeed* = r_{15}; where r_{15} was a random variable that took on integer values between 120 and 340. Its distribution was:

$$p(r_{15} = x) = \begin{Bmatrix} \frac{1}{221} & , x \in \{120, 121, \ldots, 340\} \\ 0 & \text{, otherwise} \end{Bmatrix}$$

Variable *target* = $r_9 \cdot 100$. Variable *bfpa* = r_{16}; where r_{16} was a random variable that took on real values between -8 and 8. Its density function was:

$$f_{r_{16}}(x) = \begin{Bmatrix} \frac{1}{16} & , -8 \le x \le 8 \\ 0 & \text{, otherwise} \end{Bmatrix}$$

Variable *cairspeed* = *airspeed* + r_{17}; where r_{17} was a random variable that took on integer values between -25 and 25. Its density function was:

$$p(r_{17} = x) = \begin{Bmatrix} \frac{1}{51} & , x \in \{-25, -24, \ldots 25\} \\ 0 & \text{, otherwise} \end{Bmatrix}$$

Variable *caltitude* = *target* + $r_{18} \cdot 10$; where r_{18} was a random variable that with probability 0.5 took on integer values between -5 and 5 and with probability 0.5 took on integer values between -30 and 30. Its distribution was:

$$p(r_{18} = x) = \begin{cases} \frac{1}{122} & , x \in \{-30, -29, \ldots, -6\} \\ \frac{36}{671} & , x \in \{-5, -4, \ldots, 5\} \\ \frac{1}{122} & , x \in \{6, 7, \ldots, 30\} \\ 0 & , \text{otherwise} \end{cases}$$

6.3.2.2 Scenario Execution

The scenarios were executed on a flight simulator, a modified version of NASA-Langley's Advanced Civil Transport Simulator (ACTS) (Cha, 1996).

The following displays of the simulator are relevant to this study:

- thrust, and manual flight path angle settings of the mode control panel (MCP)
- artificial horizon, altitude tape, indicated airspeed tape, and vertical speed indicator of the ADI

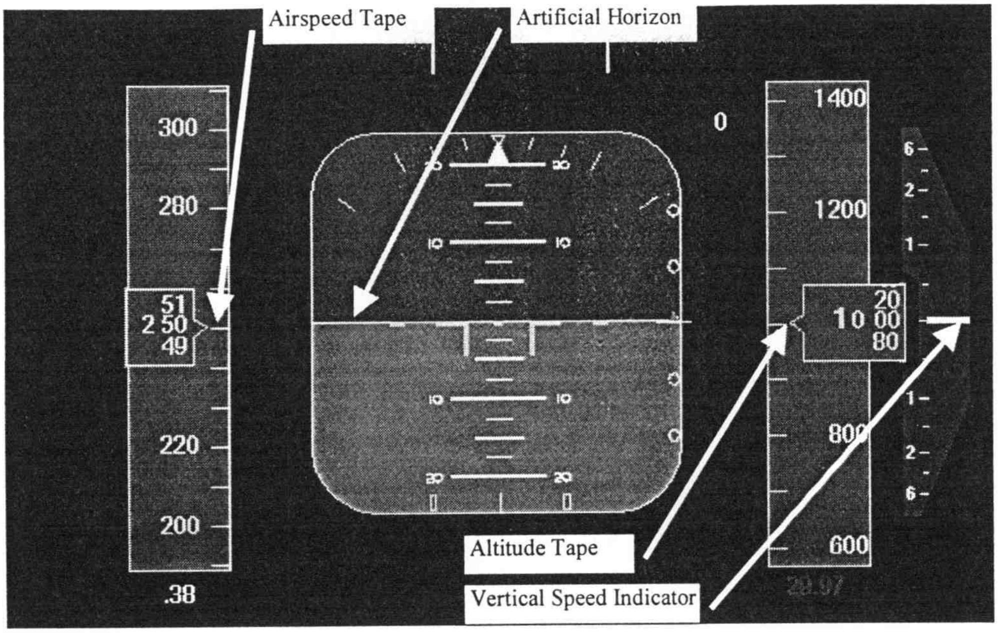


Figure 6.8 ADI display

Figure 6.8 shows the ADI representing the state of the aircraft at 250 knots, 1000 ft. above sea level, level flight. The MCP display is depicted in Figure 6.9. Executing a scenario involved bringing the airplane into the initial state as described by variables *altitude*, *fpa*, and *airspeed* of vector *S*. Once the initial state was reached, the flight simulator was interrupted in order to input the control settings of vector *C*. Thereafter, the simulator resumed while for the following 7.5 seconds flight data was recorded. All hazard scenarios were flown by the autopilot, controlling pitch and airspeed, while the altitude capture scenarios were hand-flown with the autopilot controlling only airspeed. Manual pitch control for the *capture altitude* scenarios was necessary as the autopilot was unable to capture a commanded altitude accurately. A hazard scenario for instance, specified by S = (10000 feet, 5.1 degrees, 230 kts, 15000 feet) and *C* = (6.5 degrees, 130 kts), was executed by flying the airplane to 10000 ft. at a flight path angle of 5.1 degrees and an airspeed of 230 kts. Once the state was reached the simulator was interrupted and the commanded flight path angle and the commanded airspeed were set in the MCP to 6.5 degrees and 130 kts respectively. Then the autopilot was engaged, the simulation resumed, and data recorded for the next 7.5 seconds. *Capture altitude* scenarios were executed similarly. For those scenarios pitch was controlled manually while airspeed was controlled by the autopilot. Manual pitch control was guided by variable *bfpa* and *caltitude*. Specifically, the pitch was controlled to capture altitude *caltitude* while making sure that the flight path angle would not violate the boundary specified by *bfpa*. Meanwhile, the autopilot tried to maintain the commanded airspeed *cairspeed*.

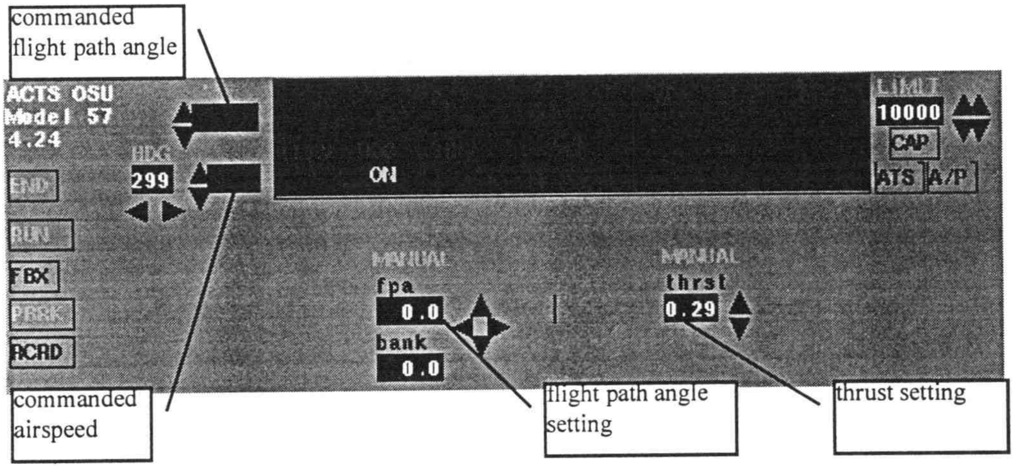


Figure 6.9 MCP display

The next section talks about how the simulator in combination with a computer-based questionnaire was used to collect assessments of human experts.

6.3.3 Collecting Human Expert Assessment

Any pilot with flight experience was accepted to serve as a member of the expert panel. No claim was made to represent a general population of pilots. The knowledge acquisition involved the following steps:

1. let subject read task description (see appendix C)
2. explain simulator interface
3. explain assessment interface description

4. let subject fly sample tasks to become familiar with airplane
5. have subject watch and rate 12 warm-up scenarios
6. have subject watch and rate 104 scenarios in random order

The subject analyzed the state dynamics of the simulated airplane with regards to the specified target altitude and rated how well the *capture altitude* function was performed with regards to the measures defined in table 6.2. The necessary information to carry out these assessments was located on the ADI, MCP, and a computer-based questionnaire. While the ADI and MCP indicated how the airplane was moving in space and time, the computer-based questionnaire displayed the target altitude and accepted ratings of the human experts. The assessment interface is depicted in figure 6.10. The target altitude for a given scenario was located on the upper left. By clicking on the "Run" button the scenario was replayed. A "ping" sound indicated whenever a scenario was over. The subjects were allowed to replay a scenario as many times as they wanted. Assessments with regards to terrain, stall, overspeed, and function performance were made in the right part of the interface. So-called "sliders" were used to specify the magnitude of a measure. The numeric assessment values were located at the far right. For example, subject 1 identified a relatively high overspeed hazard (7.1 out of 10). Subject 1 rated function performance, labeled as response accuracy on the interface, as moderately low (1.9 out of 10).

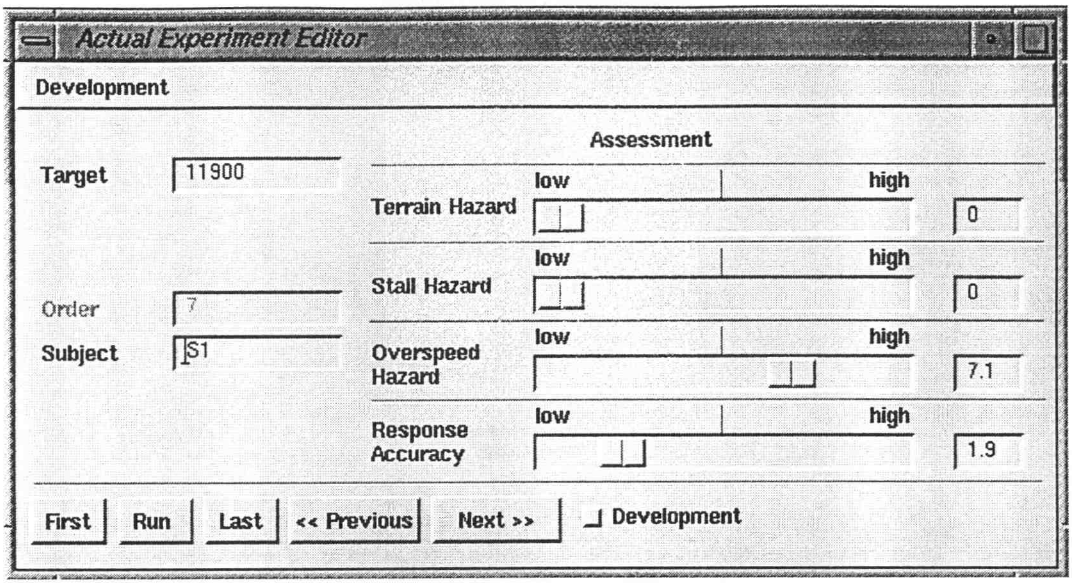


Figure 6.10 Computer-based questionnaire

Once the assessments were made, the subject started the next scenario by clicking the "Next" button. Without much interference from the researcher, the subjects were able to navigate through all 104 scenarios that were presented to them in a random order. After 6 human experts had made their assessments, averages for each measure across the human experts were calculated. The fuzzy system inputs, calculated from the recorded flight memories, in combination with the average measures, collected from the human experts, formed the sample data used for training the FFA systems. The next section addresses what had to be done to train the initial FFA systems and how the parameters of the GA were set.

6.4 Training of Initial FFA Systems

This section gives a brief overview of how the sample data was divided into training, overtraining test, and validation test data and explains how the initial fuzzy systems were trained. Before any training took place the sample data (104 runs) was divided randomly into 40 training, 32 overtraining test, and 32 validation test data samples as described in section 5.5. From each scenario group 10, 8, and 8 scenarios were randomly assigned to training, overtraining test, and validation test data set respectively. Appendix table B.1 shows the division of the scenarios into these three classes. Again, training was performed by the GA as described in chapter 5 and was facilitated by the genetic algorithm tool described in section 5.6. Training was performed on the fuzzy systems in order to improve their fitness or in other words to make them better conform to the average human expert assessment. There were two alternatives under consideration for each hazard system type. However, only one of each type was needed to supply hazard information to the FHEIS system. Therefore, a selection was made based on how well each alternative fit the union of overtraining and training data on average after the GA had performed its operations. The selected systems were then used to provide data to the FHEIS which was then trained also.

Since the GA incorporated randomness in solution finding the results varied even with constant GA parameter settings. Therefore, in order to determine how good the results were on average, the training runs were replicated 5 and 4 times for the hazard systems and FHEIS respectively. So, each initial hazard system was trained five times and in each run the best resulting fuzzy system as defined in section 5.5 was saved. Thus, 5 training runs would yield 5 fuzzy systems. Therefore, for each hazard, there were 10 trained fuzzy hazard systems, 5 from each system alternative. A choice had to be made among 30 hazard systems to supply overspeed, stall, and terrain hazard values to the FHEIS. For each hazard, the fuzzy system out of the 10 final candidates was selected that minimized the MSE with regards to union of testing and overtraining test data. This selection of hazard systems is referred to as the selected hazard systems (SHS). The initial FHEIS, supplied with hazard measure values by the SHS, was then trained in 4 runs and the system resulting in the smallest MSE regarding the union of training and overtraining test data was referred to as the selected FHEIS.

Systems FTHS 1, FTHS 2, FSHS 1, FSHS 2, FOHS 1, and FOHS 2 were trained using a GA as described in chapter 5 and the parameters were set as follows:

Table 6.3[1] GA parameter settings for hazard systems

Parameter	Value
population size	50
initial mutation rate	0.2
number of crossovers	5
regular mutation rate	0.01
maximum number of generations	100

FHEIS was trained according to the GA with slightly different parameters as depicted in table 6.4.

Table 6.4[2] GA parameter settings for FHEIS

Parameter	Value
population size	50
initial mutation rate	0.2
number of crossovers	5
regular mutation rate	0.005
maximum number of generations	150

The number of generations was higher for the FHEIS than for the hazard systems because of its greater size.

[1] The if parts of the rules in systems FTHS 1, FTHS 2, FSHS 1, FSHS 2 were excluded from GA manipulation, while the if parts of the rules of FOHS 1, and partially of 2 were included. Excluding the if parts from manipulation decreases genome size and consequently computation time.

[2] The if parts of the rules of the FHEIS were excluded from GA manipulation.

The next chapter is dedicated to the results and their analysis. As noted in section 5.5, the quality of the fuzzy systems was measured by exposing the fuzzy systems to the validation data they were not trained on. Beyond the MSE, other measures were taken into consideration to evaluate the applicability of these systems in a real environment.

7. Results and Analysis

This chapter summarizes the results and provides an analysis of the scenarios and initial and trained FFA systems. In a questionnaire human experts were asked to rate how realistic they found the scenarios. This measure along with verbal explanations of what was unrealistic about the scenarios indicates a level of face fidelity associated with this study. Another important factor was how the human expert ratings were distributed across the scenarios. The question of whether the FFA systems matched human assessment was addressed by calculating performance measures indicating the degree of agreement between the FFA systems and the human expert panel.

7.1 Scenarios

The human experts rated the realism of the scenarios on a scale between 0(unrealistic) and 10(realistic). The average response was 7.56. The individual ratings of the six human experts that participated in this study are graphed in Figure 7.1.

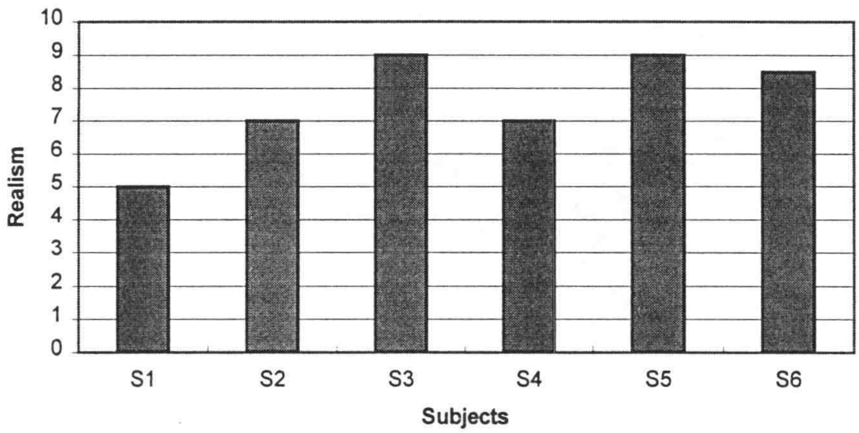


Figure 7.1 Scenario realism

Four out six subjects noted that the climb and descent rates were higher than would be observed in reality, especially in case of a civil commercial transport aircraft. Two out of six subjects complained about the thrust rate display that was considered to be too small. The distribution of human expert ratings across scenarios is described next. As can be seen from the histograms below (Figure 7.2), the human expert ratings were not evenly distributed. In fact over 50% of the scenarios obtained a function performance rating of 2.5 or lower. Similarly, over 80% of the scenarios were rated 2.5 or lower with regards to terrain, stall, and overspeed hazard. As there was an over proportionally large number of hazard-less scenarios, the fuzzy systems were trained in a way that overall would give more emphasis to hazard-free situations.

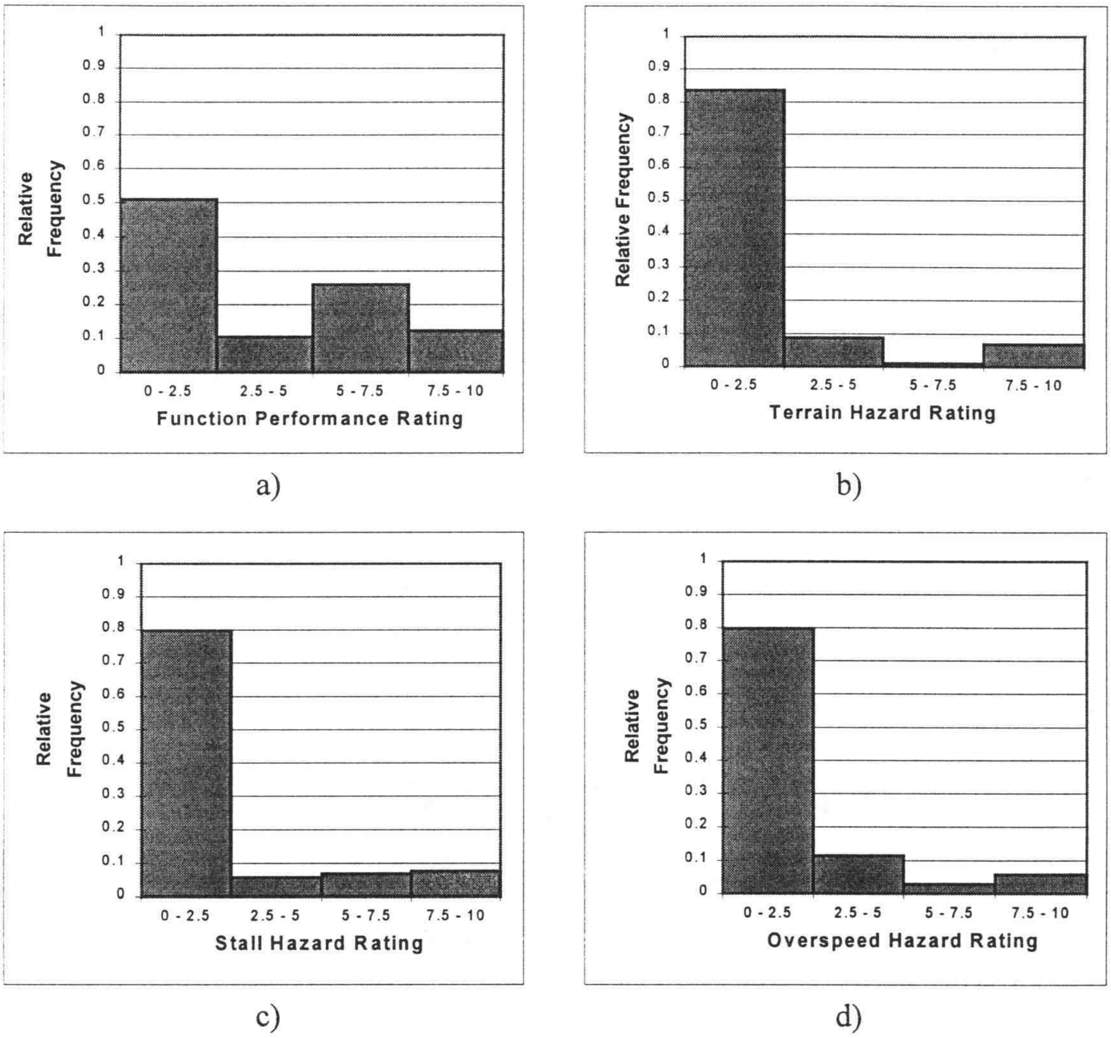


Figure 7.2 Distribution of measures within sample data. (a) function performance, (b) terrain hazard, (c) stall hazard, (d) overspeed hazard

7.2 Pilot Assessments

It is not surprising to see that human experts rated scenarios differently. The variation of ratings for each measure is depicted in the following histograms (Figure 7.3).

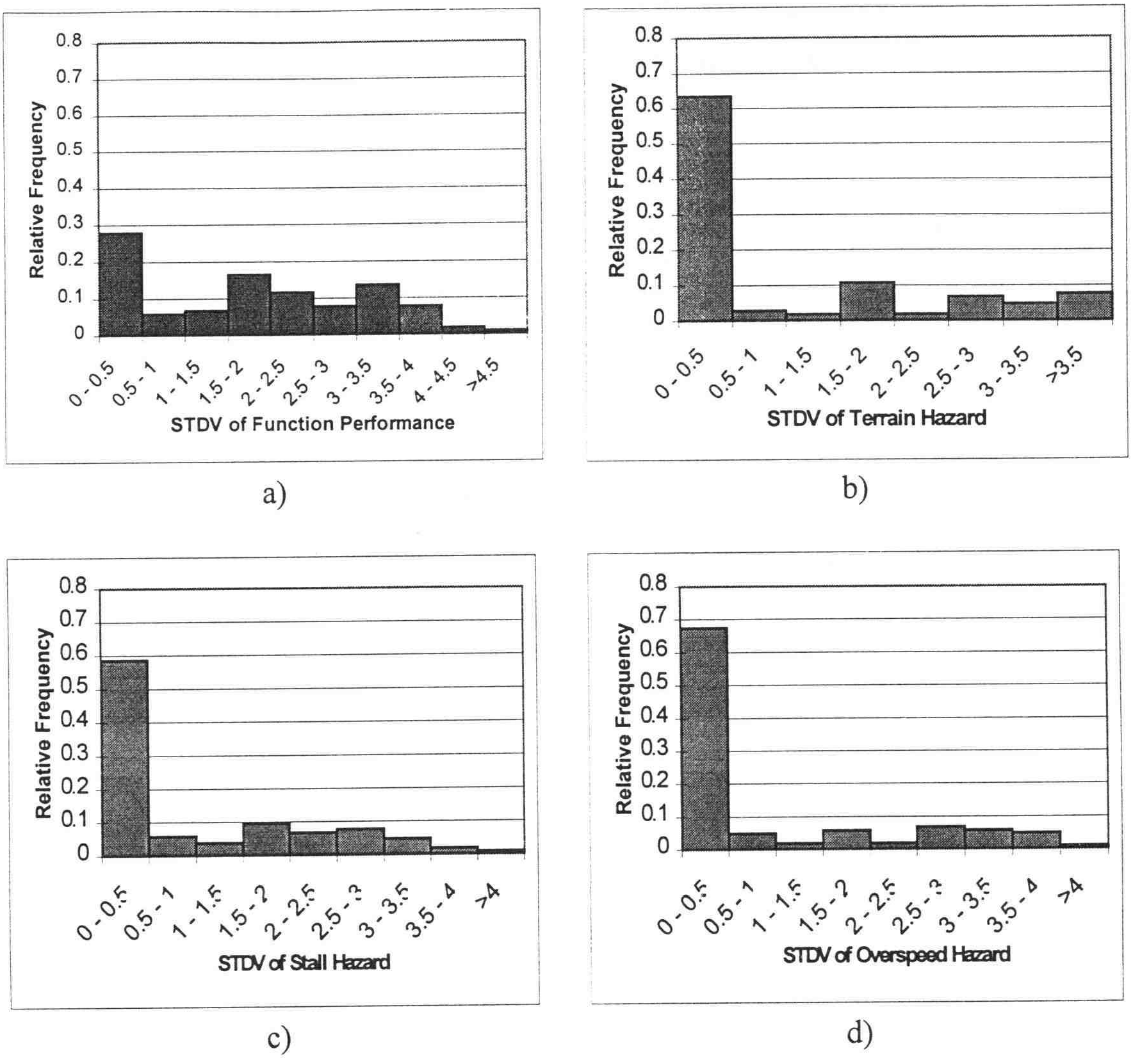


Figure 7.3 Distribution of measure standard deviation. (a) function performance, (b) terrain hazard, (c) stall hazard, (d) overspeed hazard

As can be seen from the histograms above (Figures 7.3), in which the distribution of standard deviation of measures within the expert panel is exhibited, human experts disagreed more in their function performance ratings than in their hazard ratings. The highest agreement was with regards to the overspeed hazard measure. This finding is supported by the subjective ratings the human experts gave when asked how clearly they thought the measures had been defined. On the average, the function performance measure obtained the lowest clearness rating while the overspeed measure ranked highest (Figure 7.4). When asked what the difficulties were with understanding the measures, three out of six subjects indicated problems with understanding the function performance measure. One subject stated that he/she "had a hard time determining [a] compromise between safety and target." Similarly, another subject noted that it was not easy to consider both goal compliance and safety when assessing function performance. Lastly, one subject said that the function performance measure was very subjective.

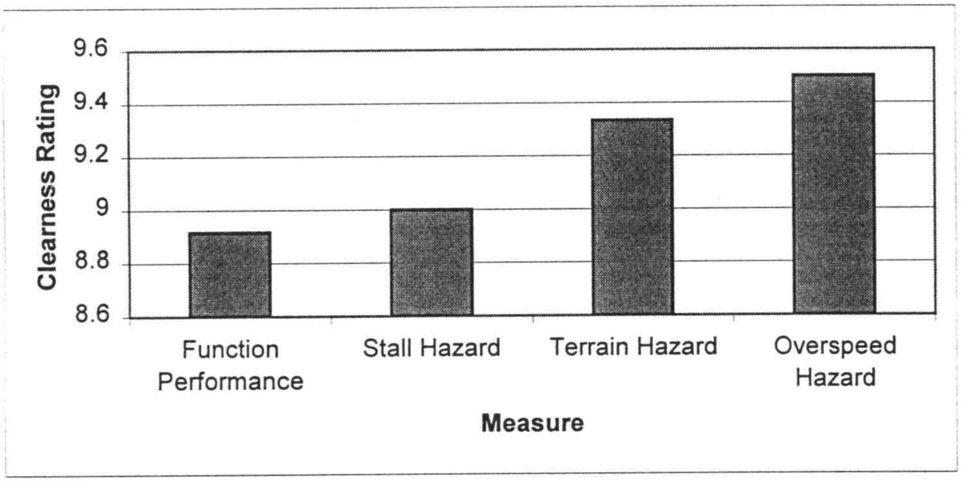


Figure 7.4 Measure clearness ratings

7.3 Data Split

As described earlier the sample data was divided into training, overtraining test, and validation test data. This section analyzes the distribution of the hazard measures within those data sets. Major differences in measure distributions between data sets are likely to reduce training effectiveness as the FFA systems are then trained and validated against dissimilar data. The following histograms depict the measure distributions within the data sets (Figure 7.5).

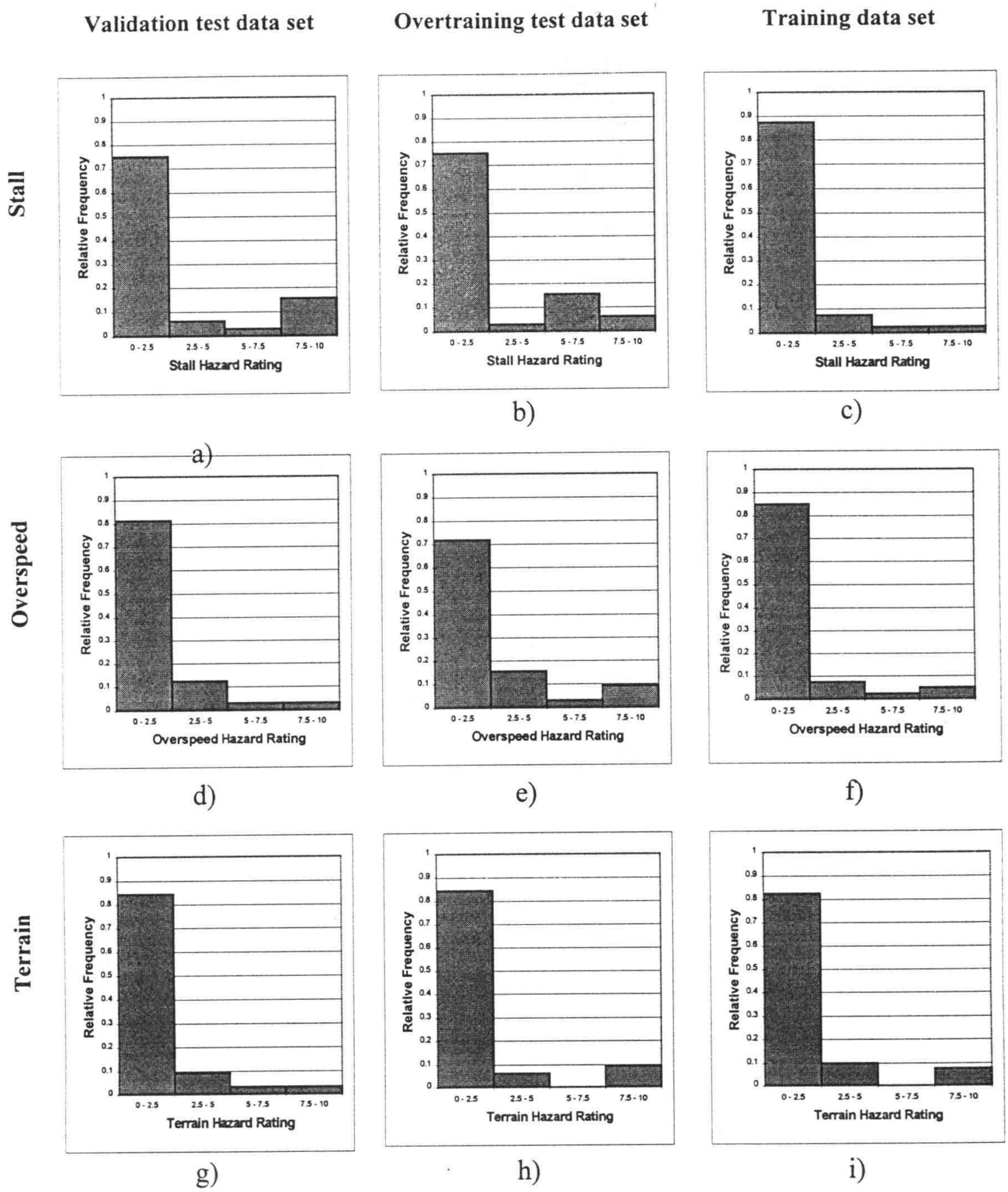


Figure 7.5 Distribution of measures within data sets (a-i)

As can be seen from figure 7.5c, there were only very few scenarios associated with high stall hazards in the training data set; however there were comparatively many of them in the validation test data set (figure 7.5a). As is discussed in section 7.4.4, the relatively mediocre training result for the stall hazard system was partly caused by the unfavorable data split.

7.4 Fuzzy Systems

FFA systems were evaluated as to how well they matched the average human expert assessment. In addition, the validity of fuzzy systems was examined for a few selective continuous value ranges. In total six performance measures were established to quantify how well the systems matched the average human expert assessment and are explained in the next section. The effectiveness of the training algorithm was assessed by comparing the fuzzy systems before and after training with regards to the performance measures. Finally, the best performing hazard system alternatives were identified.

7.4.1 Performance Measures

The six performance measures presented here are all based on the output data obtained from the FFA systems using the validation test data.

7.4.1.1 Mean of Sum of Squared Error

The mean of sum of squared error (MSE) expressed how well a fuzzy system fit data – in this case the validation test data derived from 32 scenarios. MSE is the sum of squared errors (SQE) divided by the size of the data set. In order to demonstrate how well the GA was able to train a fuzzy system, an overspeed hazard system was trained on the whole sample data. Figure 7.6 below exhibits that after a total of 100 generations the MSE of the fittest individual was a mere 0.25. The untrained fuzzy system resulted in a MSE of 9.6! After 50 generations the MSE virtually remained constant – the population converged. The lower the MSE, the better the fuzzy system fit the data and thus; the closer it matched the average human expert assessment.

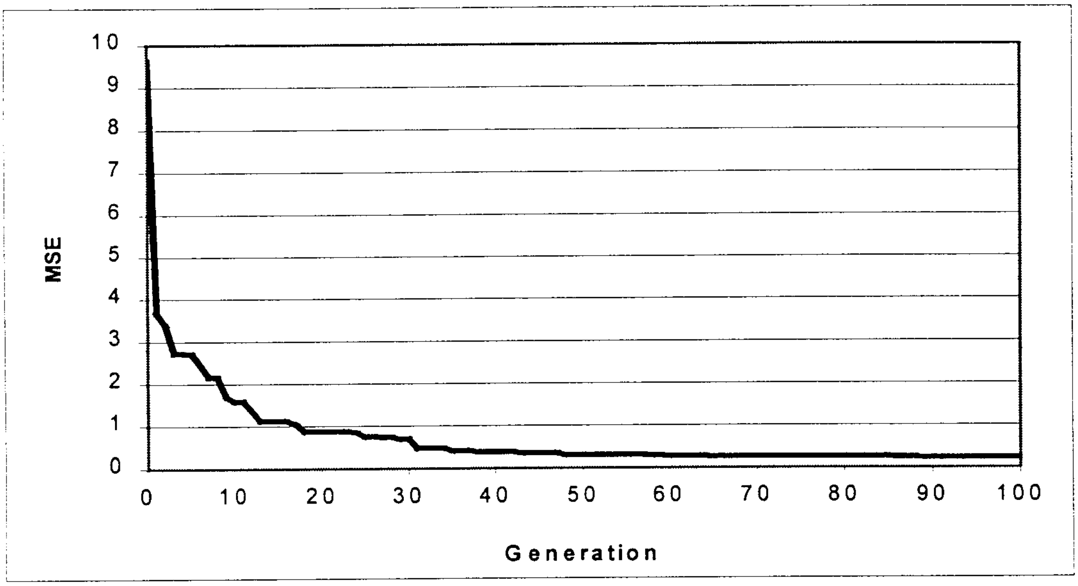


Figure 7.6 Training of FOHS 1 on sample data (104 scenarios)

7.4.1.2 Correlation Coefficient

The correlation coefficient is defined as:

$$r = \frac{\frac{1}{n}\cdot\sum_{i=1}^{n}(x_i - \bar{x})\cdot(y_i - \bar{y})}{\sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}(x_i - \bar{x})^2}\cdot\sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

In the context of this study, x_i was the actual fuzzy system output for scenario i, while y_i was the average human expert assessment for the corresponding measure. Variable $n = 32$, since the FFA systems were evaluated using the validation test data set. Note that the correlation coefficient r takes on values between 1 and –1. It was used as another indicator expressing how well the fuzzy systems matched the human experts' assessments. The closer r is to one the better the match between system and human expert panel.

7.4.1.3 False Alarm Rate

In context of this study the false alarm rate was defined for hazards and function performance as follows. Regarding hazards, the false alarm rate was defined as the relative frequency of cases in which the fuzzy system output a value greater than 5, while the corresponding average human expert assessment was equal or less than 5. Regarding function performance, it was defined as the relative frequency of cases in which the FHEIS output a value less than 5, while the corresponding average human expert assessment was equal or greater than 5. While the choice of cutoff value 5 was arbitrary, it was implied that hazard values greater than 5 represented hazards that were developed enough to issue a full alert. Similarly, function performance values less than 5 represented cases with highly deteriorated function performance.

7.4.1.4 Missed Alarm Rate

Regarding the hazard systems, the missed alarm rate was defined as the relative frequency of cases in which the value provided by the fuzzy system was equal or less than 5, while according to the human expert panel the value should have been greater than 5. Regarding FHEIS, the missed alarm rate was defined as the relative frequency of cases in which the value provided by FHEIS was equal or greater than 5, while according to the human expert panel the value was rated less than 5.

7.4.1.5 Overestimation Percentage

The performance metric overestimation percentage was defined as the relative frequency of cases in which the fuzzy systems output a value that was at least 2 units greater than the corresponding human expert assessment.

7.4.1.6 Underestimation Percentage

The underestimation percentage was defined as the percentage of cases in which the fuzzy system output a value that was at least 2 units smaller than the corresponding human expert assessment.

7.4.2 Training Effectiveness

This section focuses on the effectiveness of the training procedure. The quality of the FFA systems was measured before and after training in terms of the performance metrics described in section 7.4.1. Table 7.1 summarizes the performance metrics for trained and untrained systems.

Table 7.1 Performance measures

System	Correlation	MSE	False Alarm Rate	Missed Alarm Rate	Over-Estimation	Under-Estimation
Initial FTHS 1	0.754	1.900	0.000	0.500	0.000	0.031
Trained FTHS 1	0.767	1.549	0.007	0.300	0.006	0.138
p-Value	0.726	0.230	0.326	**0.142**	0.326	**0.003**
Initial FTHS 2	0.539	4.150	0.000	1.000	0.313	0.063
Trained FTHS 2	0.757	1.539	0.007	0.500	0.019	0.056
p-Value	**0.014**	**0.001**	0.326	**0.024**	**<0.001**	0.720
Initial FSHS 1	0.863	4.523	0.154	0.000	0.344	0.031
Trained FSHS 1	0.799	4.719	0.046	0.567	0.050	0.150
p-Value	**0.141**	0.765	**0.001**	**0.018**	**<0.001**	**0.004**

Table 7.1 (Continued)

Initial FSHS 2 Trained FSHS 2	0.804 0.720	8.558 5.567	0.192 0.054	0.000 0.567	0.656 0.044	0.125 0.131
p-Value	**0.071**	**<u>0.006</u>**	**<u>0.001</u>**	**<u><0.001</u>**	**<u><0.001</u>**	0.776
Initial FOHS 1 Trained FOHS 1	0.697 0.910	9.571 0.932	0.300 0.000	0.000 0.800	0.594 0.006	0.000 0.069
p-Value	**<u><0.001</u>**	**<u><0.001</u>**	**<u><0.001</u>**	**<u>0.002</u>**	**<u><0.001</u>**	**<u>0.003</u>**
Initial FOHS 2 Trained FOHS 2	0.669 0.860	20.519 1.397	0.533 0.027	0.000 0.500	0.906 0.044	0.000 0.056
p-Value	**<u>0.007</u>**	**<u><0.001</u>**	**<u><0.001</u>**	**0.067**	**<u><0.001</u>**	**0.066**
Initial FHEIS Trained FHEIS	0.561 0.704	2.370 2.528	0.733 0.467	0.059 0.118	0.375 0.250	0.313 0.242
p-Value	**0.084**	0.632	**<u>0.011</u>**	**<u>0.018</u>**	**<u>0.002</u>**	**<u>0.018</u>**

Entries regarding trained fuzzy systems represent averages calculated from 5 training runs. As mentioned earlier the training results varied and therefore, it had to be determined if the difference between the trained systems and the initial system were significant. In table 7.1 significant differences are typed in bold face and are underlined, suggestive differences are typed in bold face. The following hypothesis test was conducted for each measure to determine if a difference was significant or suggestive or neither (Montgomery & Hines, 1980):

Consider x is a normally distributed random variable denoting the performance measure regarding a trained fuzzy system. Its mean μ and standard deviation σ are unknown. It is wished to test the hypothesis that μ equals the performance measure value of the corresponding untrained system. Given a random sample of n=5, the sample mean $\bar{x}$ and standard deviation S can be calculated. The two-sided alternative is tested as follows:

H_0: $x = x_0$

H_1: $x \neq x_0$

The test statistic is:

$$t_0 = \left| \frac{x - x_0}{\frac{S}{\sqrt{n}}} \right|$$

A p-value representing the significance level is calculated as $p = P(|t| > t_0)$; where t follows the t distribution with n-1 degrees of freedom if the null hypothesis is true. For the purposes of this study a p-value of 0.1 or below was considered significant while a value of 0.2 or below was considered suggestive.

Regarding FHTS 1, training increased correlation from 0.754 to 0.767 at p=0.73. Training reduced MSE from 1.9 to 1.549 at p=0.23. It increased false alarm rate from 0 to 0.007 at p=0.33, reduced missed alarm rate from 0.5 to 0.3 at p=0.14, increased overestimation rate from 0 to 0.006 at p=0.33, and increased underestimation rate from 0.031 to 0.138 at p=0.003. Overall, it is felt that the training procedure did little to improve system FHTS 1 performance.

In case of FTHS 2, the training procedure increased correlation from 0.539 to 0.757 at p=0.014 and reduced MSE from 4.15 to 1.539 at p=0.001. It increased false alarm rate from 0 to 0.007 at p=0.33, reduced the missed alarm rate from 1.0 to 0.5 at p=0.024, reduced overestimation rate from 0.313 to 0.019 at p≅0, reduced underestimation rate from 0.063 to 0.056 at p= 0.72.

Concerning FSHS 1, the training procedure decreased correlation from 0.863 to 0.799 at p=0.14 and increased MSE from 4.523 to 4.719 at p=0.765. It reduced false alarm rate from 0.154 to 0.046 at p=0.001, increased missed alarm rate from 0 to 0.567 at p=0.018, reduced overestimation rate from 0.344 to 0.05 at p≅0, and increased underestimation rate from 0.031 to 0.15 at p=0.004. In this case it is argued that training actually deteriorated system performance.

In case of FSHS 2, the training procedure decreased correlation from 0.804 to 0.72 at p=0.077 and decreased MSE from 8.558 to 5.567 at p=0.006. It reduced false alarm rate from 0.192 to 0.054 at p=0.001, increased missed alarm rate from 0 to 0.567 at p≅0, reduced overestimation rate from 0.656 to 0.044 at p≅0, and increased underestimation rate from 0.125 to 0.131 at p=0.776.

With regards to FOVS 1, the training procedure increased correlation from 0.697 to 0.[illegible] at p≅0, decreased MSE from 9.571 to 0.932 at p≅0. It reduced false alarm rate fr[illegible]3 to 0 at p≅0, increased missed alarm rate from 0 to 0.8 at p=0.002, decreased [illegible]stimation rate from 0.594 to 0.006 at p≅0, and increased underestimation rate from [illegible] to 0.069 at p=0.003. In this case, it is felt [illegible]ning procedure improved system performance. However, of great concern is that training significantly increased the missed alarm rate.

With regards to FOVS 2, the training procedure increased correlation from 0.669 to 0.86 at p=0.007 and reduced MSE from 20.519 to 1.397 at p≅0. It reduced false alarm rate from 0.533 to 0.027 at p≅0, increased missed alarm rate from 0 to 0.5 at p=0.067, reduced overestimation rate from 0.906 to 0.044 at p≅0, and increased underestimation rate from 0 to 0.056 at p=0.066.

With regard to FHEIS, the training procedure increased correlation from 0.561 to 0.704 at p=0.084 and increased MSE from 2.37 to 2.528 at p=0.632. It reduced false alarm rate from 0.733 to 0.467 at p=0.011, increased missed alarm rate from 0.059 to 0.118 at p=0.018, reduced overestimation rate from 0.375 to 0.25 at p=0.002, and reduced underestimation rate from 0.313 to 0.242 at p=0.018. The training procedure is felt to have improved system performance, however, I would call the performance level unacceptable as the false alarm rate was 11.8% and the missed alarm rate was 46.7%.

The next section analyzes the effect of training on the performance measures in general. A nonparametric test, the Wilcoxon Signed-Rank Test (Hines and Montgomery, 1980) was used to find out if testing significantly affected the performance measures across all training runs of all systems. As can be seen from Table 7.2, training on average increased correlation by 0.088, increased MSE by 4.911, reduced false alarm rate by 0.184, increased missed alarm rate by 0.262, reduced overestimation rate by 0.403, and increased underestimation rate by 0.043. All observed differences were significant.

Table 7.2 Training effect on performance measures

System	ΔCorrelation	ΔMSE	ΔFalse Alarm Rate	ΔMissed Alarm Rate	ΔOver-Estimation	ΔUnder-Estimation
FTHS 1	0.013	-0.351	0.007	-0.200	0.006	0.106
FTHS 2	0.219	-2.612	0.007	-0.500	-0.294	-0.006
FSHS 1	-0.065	0.197	-0.108	0.567	-0.294	0.119
FSHS 2	-0.084	-2.990	-0.138	0.567	-0.613	0.006
FOHS 1	0.213	-8.639	-0.300	0.800	-0.588	0.069
FOHS 2	0.191	-19.122	-0.507	0.500	-0.863	0.056
FHEIS	0.142	0.158	–0.267	0.059	-0.125	-0.070
Average over System Averages	0.090	-4.555	-0.173	0.256	-0.385	0.044
Average over all systems and runs	0.088	-4.911	-0.184	0.262	-0.403	0.043
Wilcoxon Signed Test Results (confidence level = 0.05)	significant	significant	significant	significant	significant	signifcant

In addition to that, the training impact on selected fuzzy systems (SFS) and FHEIS is demonstrated by examining how they behaved over a continuous range of values. Some figures depict examples in which the initial systems were modified in a reasonable way, whereas, in others the systems output controversial values. Figure 7.7 depicts how training improved the initial FHEIS in the special case (past altitude error = 800, current altitude = 10000, current airspeed = 230, current airspeed acceleration = 0, current flight path angle = 3, current vertical speed = 1218, current vertical speed acceleration = 0, current thrust = 1, past vertical speed = 0).

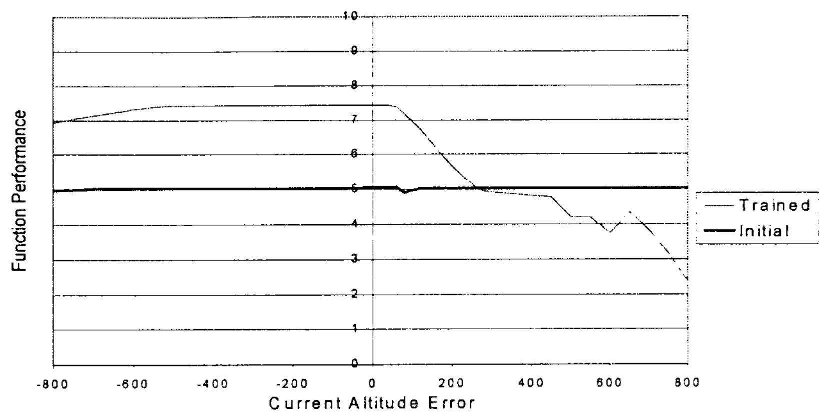


Figure 7.7 FHEIS behavior intially and after training

With decreasing positive current altitude error, the trained system increased the function performance rating while the initial system assigned a constant value of around 5. In this case the training procedure improved the behavior of the fuzzy system. Figure 7.8 demonstrates that the trained FTHS 2 behaved unreasonably compared with the initial version given the situation (current altitude = 1000, current flight path angle = -9, current thrust = 1, current altitude error = -20000).

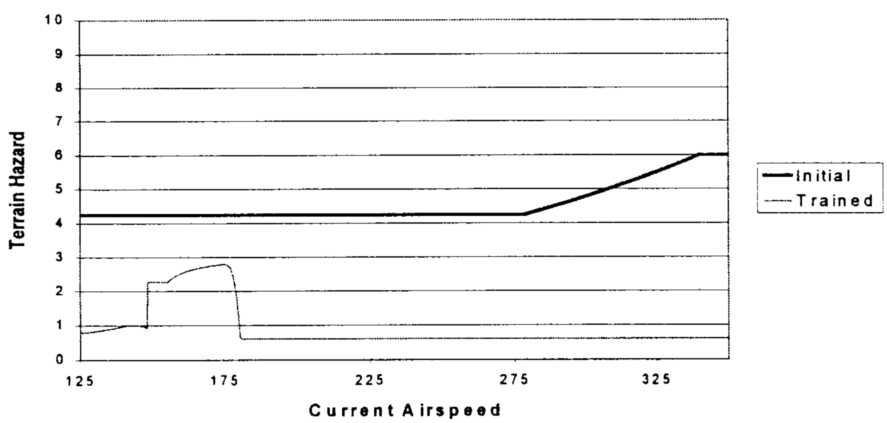


Figure 7.8 FTHS 2 behavior intially and after training

With increasing airspeed the trained system did not increase terrain hazard. This was unreasonable since in this example the airplane headed towards the ground while the goal was to actually climb to 21000 ft. Figure 7.9 exhibits a situation at current airspeed = 121 kts.

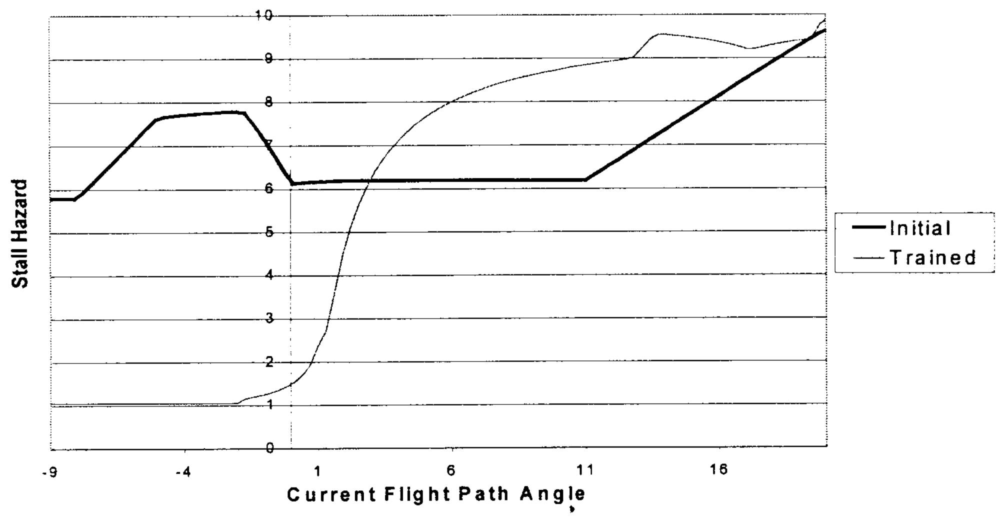


Figure 7.9 FSHS 1 behavior initially and after training

The trained FSHS 1 responded with significantly lower stall hazard values for negative large flight path angles compared with the initial system. This made sense, as a stall during rapid descent was unlikely. Figure 7.10 depicts the effect of training FOHS 1 for the special case at which current airspeed acceleration = 2.2 kts/second.

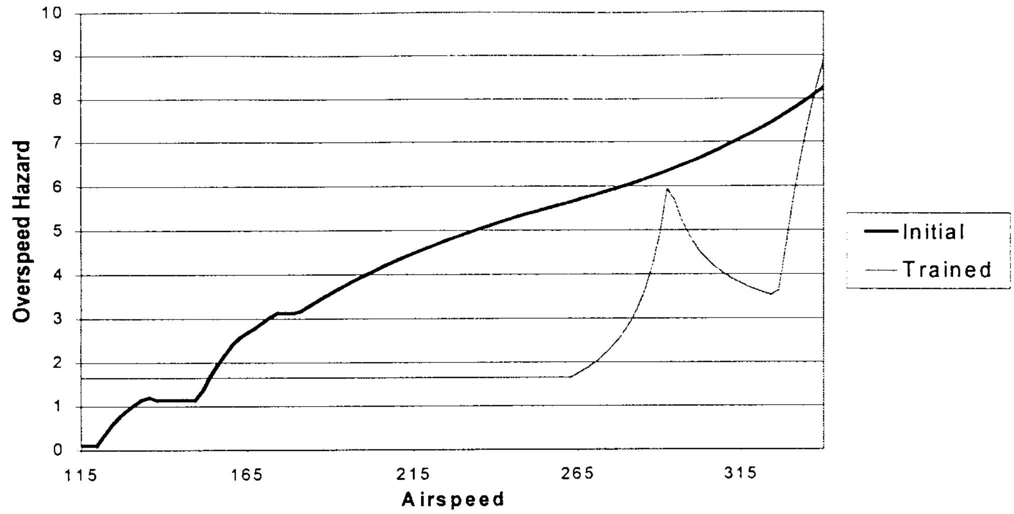


Figure 7.10 FOHS 1 behavior initially and after training

The trained system behaved strangely around 300 kts airspeed. As expected, the system increased overspeed hazard as airspeed increased up to around 300 kts. As airspeed increased beyond 300 kts, the system reduced terrain hazard unexpectedly until airspeed reached about 325 kts. From then on overspeed hazard increased sharply with increasing airspeed.

The question whether the fuzzy systems worked better for certain situations than in others was addressed by examining the trained SHS as well as the trained selected FHEIS. The output of each system was compared with the corresponding average pilot score for all validation scenarios. Figure 7.11 depicts the plot of pilot terrain hazard score vs system terrain hazard score contained in the validation test data set.

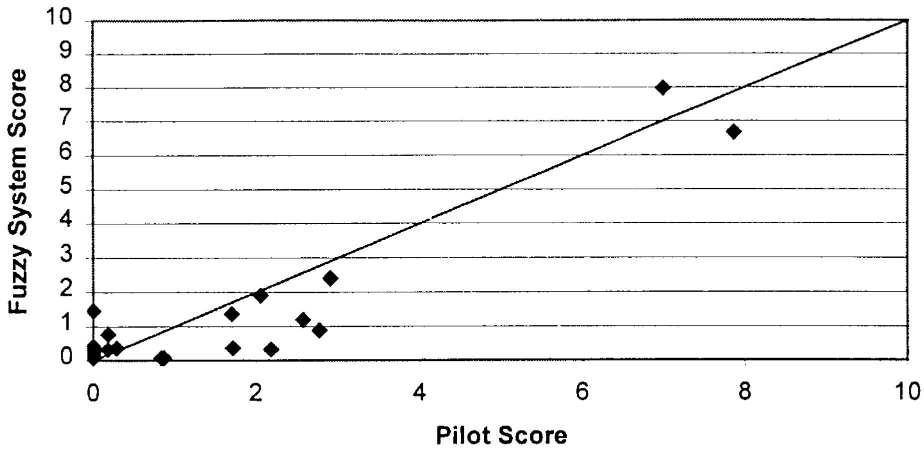


Figure 7.11 Pilot score vs FTHS 2 score

As can be seen from the plot, most pilot scores were made in the lower range between 0 and 3. Overall, the fuzzy system scored similarly. A perfect fuzzy system would match pilot scores exactly, in such a case all data points would be situated on the 45° line. As can be seen from figure 7.12 and 7.13, the stall and overspeed hazard system fared well for scenarios involving low pilot scores, but missed to respond correctly in cases when a high hazard was present. For high pilot scores, the stall and overspeed system consistently output too low values, in other words the systems underestimated the hazard.

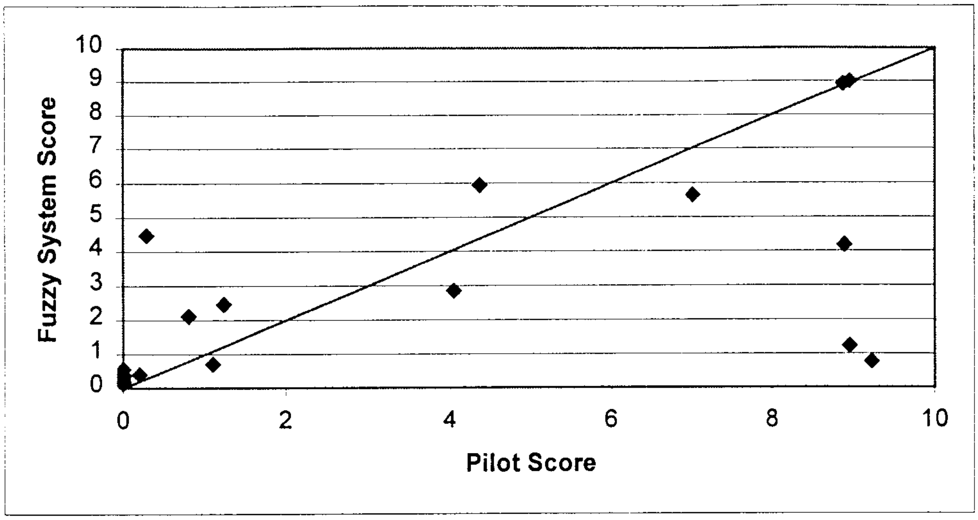


Figure 7.12 Pilot score vs FSHS score

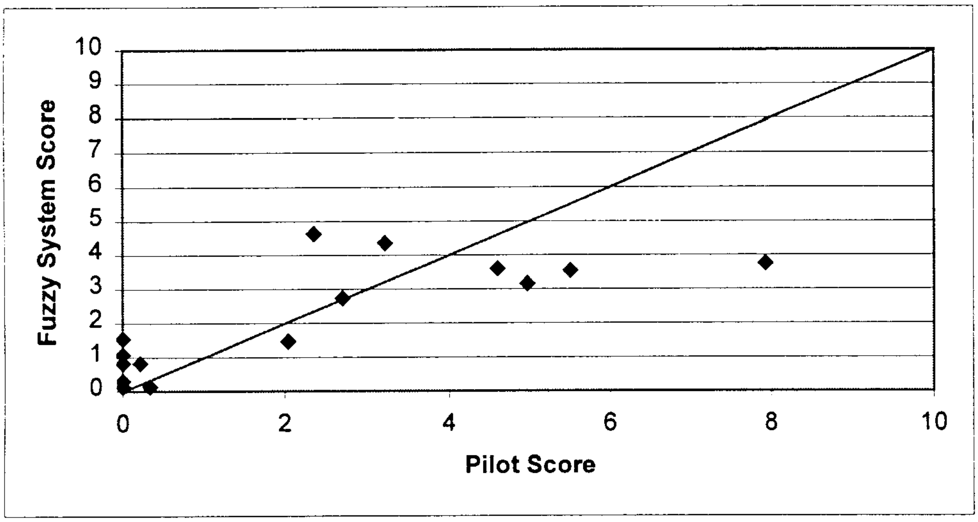


Figure 7.13 Pilot score vs FOHS score

With regards to the function performance measure, one can tell from figure 7.14, that the FHEIS overestimated for low pilot scores and underestimated for high ones.

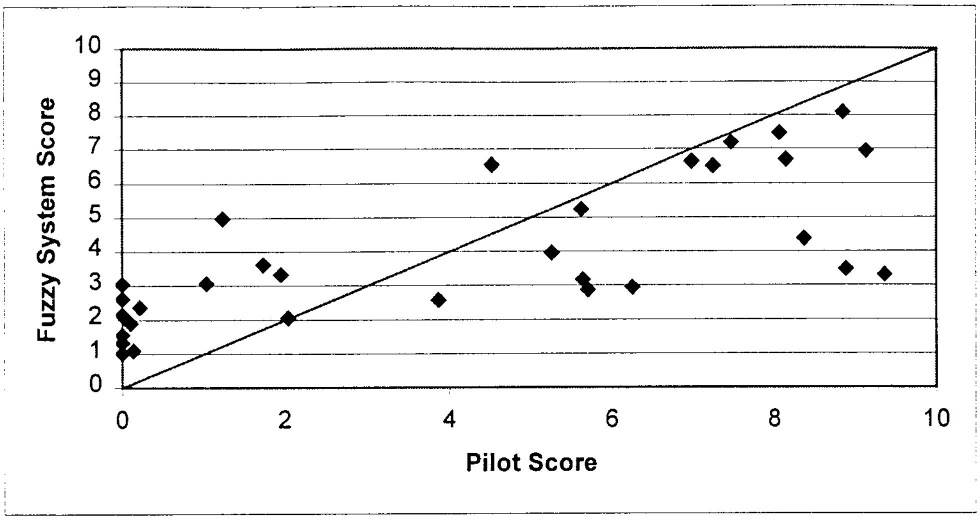


Figure 7.14 Pilot score vs FHEIS score

The next section deals with a comparison between trained systems.

7.4.3 Comparison between Trained Hazard System Alternatives

The question of which of the system alternatives adapted better to the validation test data set is addressed. Again, a t-test as described in section 7.4.2 was conducted to identify significant and suggestive differences. Table 7.3 summarizes the results.

Table 7.3 Comparison between system altnernatives

System	Correlation	MSE	False Alarm Rate	Missed Alarm Rate	Over-Estimation Rate	Under-Estimation Rate
Trained FTHS 1	0.767	1.549	0.007	0.300	0.006	0.138
Trained FTHS 2	0.757	1.539	0.007	0.500	0.019	0.056
p-Value	0.900	0.982	0.999	0.347	0.406	**0.015**

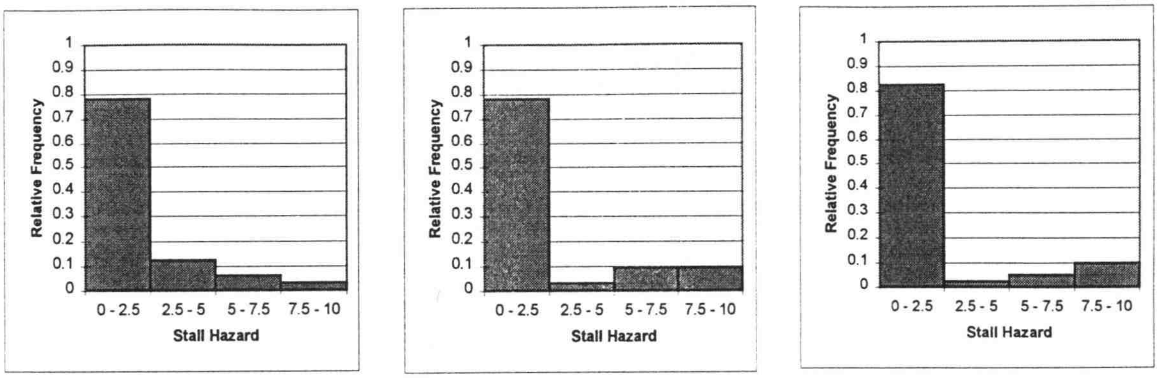
Table 7.3 (Continued)

Trained FSHS 1	0.799	4.719	0.046	0.567	0.050	0.150
Trained FSHS 2	0.720	5.567	0.054	0.567	0.044	0.131
p-Value	**0.200**	0.388	0.724	0.999	0.683	0.580
Trained FOHS 1	0.910	0.932	0.000	0.800	0.006	0.069
Trained FOHS 2	0.860	1.397	0.027	0.500	0.044	0.056
p-Value	0.330	0.260	**0.099**	0.284	**0.08**	0.667

A comparison between trained FTHS 1 and 2 revealed that the only significant difference was observable with regards to the underestimation rate, which was higher for system 1. A comparison between trained FSHS 1 and 2 indicated that higher correlation for system 1 was suggestive. A comparison between FOSH 1 and 2 indicated that there were significant differences in false alarm rate, and overestimation rate. In both cases, FOSH 1 fared better.

7.4.4 Effect of Scenario Split

As noted before, the distribution of the average human expert assessment with regards to stall hazard was notably dissimilar across training, overtraining test, and validation test data set. Therefore, the sample data was resplit, and the initial FSHS 1 was trained again. After resplit, the distribution of the stall hazard assessment was distributed similarly within each data set as can be seen from figure 7.15.



a) b) c)

Figure 7.15 Distribution of stall hazard after resplit. (a) validation test, (b) overtraining test, (c) training data set

As expected, training effectiveness improved under resplit conditions as can be seen from table 7.4.

Table 7.4 Training effect on FSHS 1 under resplit and normal conditions

System	Correlation	MSE	False Alarm Rate	Missed Alarm Rate	Over-Estimation	Under-Estimation
Initial FSHS 1	0.863	4.523	0.154	0.000	0.344	0.031
Trained FSHS 1	0.799	4.719	0.046	0.567	0.050	0.150
p-Value	**0.141**	0.765	**0.001**	**0.018**	**<0.001**	**0.004**
Initial FSHS 1 Resplit	0.811	4.516	0.172	0.000	0.344	0.000
Trained FSHS 1 Resplit	0.841	1.433	0.007	0.333	0.031	0.069
p-Value	**0.100**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**0.007**

Correlation increased from 0.811 to 0.841, MSE decreased from 4.516 to 1.433, false alarm rate decreased from 0.172 to 0.007, missed alarm rate increased from 0 to 0.333, overestimation rate decreased from 0.34 to 0.031, and underestimation rate increased from 0 to 0.069. The observed differences were all significant. In addition, note that before data resplit, the trained FSHS 1 actually reduced correlation suggestively and did not reduce MSE, whereas under resplit, the system increased correlation and reduced MSE significantly. Table 7.5 shows that FSHS 1 under resplit condition outperformed its counterpart under normal condition in the categoriesas MSE, false alarm rate, and underestimation rate significantly.

Table 7.5 Comparison between trained FSHS 1 under resplit and normal conditions

System	Correlation	MSE	False Alarm Rate	Missed Alarm Rate	Over-Estimation	Under-Estimation
Trained Stall 1	0.799	4.719	0.046	0.567	0.050	0.150
Trained Stall 1 Resplit	0.841	1.433	0.007	0.333	0.031	0.069
p-Value	0.367	**0.009**	**0.049**	0.226	0.273	**0.022**

7.5 Summary

Overall, the quality of the resulting fuzzy systems was unacceptable for use in any real environment. For instance, in 50% of the validation scenarios the FHEIS deviated more than two units from the average human response. Also, a great concern is the high missed alarm rate for any of the trained systems. Five out of the seven system types listed in table 7.1 yielded missed alarm rates in excess of 50%. While on one hand the training procedure improved correlation, MSE, false alarm rate, and overestimation rate, it deteriorated missed alarm rate and underestimation rate significantly. Worrisome is the observation that the training procedure rendered the system controversial for certain input values. I believe that many of the above shortcomings are caused by insufficient sample data. As explained, FSHS 1 did not improve during training as the result of a detrimental division of the sample data into training, overtraining test, and validation test data. However, after resplitting the sample data, training significantly improved FSHS 1 performance mainly because the resplit was more homogeneous than the split before. Thus, it was shown that the training algorithm was very sensitive to how the 104 data sets were split. This high degree of sensitivity was caused by a lack of medium to high hazard scenarios. As there were only a few high hazard scenarios, the probability of distributing those unevenly across training, overtraining test, and validation test data set was not negligible. In addition, the lack of medium to high hazard scenarios biased the training procedure towards producing good results for low hazard scenarios and less so for high hazard ones. Ways to improve the results, practical relevance of this study, and major challenges faced in this study are described in the next chapter.

8. Conclusions and Recommendations

This chapter presents some final thoughts about this study by mentioning its inherent limitations and highlighting general conclusions and research recommendations.

8.1 Study Limitations

Mainly three factors limit this study to be generalizable in a broad sense. First of all, scenarios were created using a flight simulator whose aerodynamic model did not match any real existing airplane and limiting assumptions were made regarding the scenarios (see section 6.3.2). Secondly, the scenarios were created by the researcher and were conceived unrealistic to a certain extent by the human expert panel. Thirdly, the panel by no means represented a set of best experts or a representative set of pilots.

8.2 General Conclusions

In this section conclusions are drawn with regards to the effectiveness of the employed methods and potentially useful applications are mentioned. Major challenges and problem areas of this approach are identified as well.

As part of this thesis research, fuzzy systems were created to output values representing degrees of overspeed hazard, stall hazard, terrain hazards and function performance. The initial quality of these systems, as defined by the measures of section 7.4.1, could be improved through the application of a genetic algorithm with respect to correlation, MSE, false alarm rate, and overestimation rate. Regarding these measures, the training method was successful in making the fuzzy systems better conform to human expert assessments. While the performance of trained systems was far from being good enough to be employed in a real environment, this study is a first step towards implementing advisory systems indicating a degree of emergency and function performance. It is felt that fuzzy systems are very suitable for defining an "electronic cocoon" that would sense to what degree the pilot violated operational limits and safety constraints. While the underlying logic of current warning and alerting systems determines if a trigger condition is present or not, fuzzy systems are well suited to indicate to what degree the hazards are present and therefore allow for the identification of trends indicating ensuing problems as well as much more flexible alarm formats. There are instances in which an early indication of a hazard might have saved lives. For instance, the current form of the GPWS triggered an alarm too late for the aircrew of American Airlines, flight 965 to recover the airplane that was approaching a mountain rapidly (Aeronautica Civil of the Republic of Colombia, 1995). As the subjects of this study assessed hazards similarly and the fuzzy hazard systems could be trained to conform better to the average subject rating, it is believed that fuzzy logic may very well be a technology to be used for future GPWS, TCAS, and WSAS implementations. Modified and more accurate FFA systems may find application in pilot training operations where the task of evaluating pilot performance would be performed by fuzzy systems automatically assessing how well the pilot is performing the assigned tasks.

However, there are a number of challenges to the approach taken in this study. First, the knowledge acquisition process took a lot of time and effort as scenarios had to be artificially created and rated by human experts. Designing scenarios to yield good data for training was an objective hard to meet. Second, as the number of variables that a fuzzy system had to handle increased, the size of its rule base increased almost exponentially. For example the FHEIS consisted of close to 500 rules; too many to debug the system with ease. Third, during training the genetic algorithm injected "errors" randomly into the rule base. Some of these errors were not revealed by the validation data. In other words, good results with regards to the performance measures of section 7.4.1 did not guarantee that the trained systems would behave reasonably over a wide range of inputs. Fourth, designing the fuzzy systems to perform FFA was an art more than anything else. Making the choice of which variables to use as input variables, selecting the number of linguistic terms, and designing the initial knowledge base were guided mostly by intuition. Fifth, there are unresolved certification issues with respect to AI-based avionics systems (Harrison, Saunders, 1993). As the method to demonstrate that avionics software is compliant with FAA requirements was designed for conventional software, a clear guideline of how to obtain certification for AI-based systems is lacking. The question of how to verify and validate the knowledge base of AI-based systems has yet to be answered. Sixth, the results show that assessing function performance was a subjective process. For the subjects it was a lot easier to assess hazards that they understood. The meaning of function performance was not clearly defined. Seventh, the question remains how many and what kinds of experts should be used to train the fuzzy systems and gauge their goodness.

8.3 Recommendations for Future Research

It is believed that the amount and kind of training data available was not sufficient to train the fuzzy systems for medium and high hazard levels. Part of the problem was that the training data contained only a few cases in which medium and high hazard levels were present. Therefore, to create a more reliable FFA system more high-hazard level data should be collected. Also, the data should be clustered prior to being used for training in order to remove redundant and conflicting data. To make the fuzzy system easy to debug, it is essential to limit the size of the rule base to reasonable levels by keeping the number of variables to a bare minimum. Further research could concentrate on if gains in accuracy and user acceptance can be achieved by using fuzzy logic to drive GPWS, TCAS, and WSAS. If FFA is implemented in a real environment, designing its front end becomes an issue. In particular, ways to display and represent the outputs of FFA have to be found. With regards to auditory warnings, modulating the loudness of the warning message could facilitate the level of emergency. Warning messages displayed on a CRT could be changed in size, color, or form with varying degrees of emergency. Further studies could follow, testing whether alarms expressing a degree of emergency are preferable over ones that do not.

BIBLIOGRAPHY

Aeronautica Civil of the Republic of Columbia (1995). Aircraft accident report, controlled flight into terrain, American Airlines, flight 965, Boeing 757-223, N651AA, near Cali, Columbia, December 20, 1995. Santafe de Bogota, D.C., Columbia.

Altrock, C. von. (1993). Fuzzy logic - Band 1: Technologie. München/Wien: Oldenbourg Verlag.

Altrock, C. von. (1995). Fuzzy logic applications in Europe. In J. Yen, R. Langari, & L. A. Zadeh, (Eds.), Industrial applications of fuzzy logic and intelligent systems (pp. 277-310). New York: IEEE, Inc.

Altrock, C. von. (1997). Fuzzy logic and neurofuzzy applications in business and finance. Upper Saddle River, NJ: Prentice Hall PTR.

Banks, S. B., Lizza, C. S. (1991). Pilot's Associate. IEEE, 6, 18-29.

Boeing (1988). Model 757 operations manual, volume 1. Seattle, WA: Boeing Commercial Airplanes.

Boeing (1995). Statistical summary of commercial jet aircraft accidents, worldwide operations, 1959-1995. Seattle, WA: Boeing Commercial Airplanes.

Boy, G. (1991). Intelligent Assitant Systems. Academic Press, London, 1991.

Bureau of Transportation Statistics (1995). National transportation statistics: 1995.

Cha, W. C. (1996). Communicating pilot goals to an intelligent cockpit aiding system. Unpublished doctoral dissertation, Oregon State University.

Chiu, S. (1995). Software tools for fuzzy control. In J. Yen, R. Langari, & L. A. Zadeh, (Eds.), Industrial applications of fuzzy logic and intelligent systems (pp. 277-310). New York: IEEE, Inc.

Chou, C. (1991). Cockpit task management errors: a design issue for intelligent pilot – vehicle interfaces. Unpublished doctoral dissertation, Oregon State University.

Chou, C., Madhavan, D., Funk, K. (1996). Studies of cockpit task management errors. The International Journal of Aviation Psychology, 6(4), 307-320.

Dote, Y. (1995). Introduction to fuzzy logic. Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation, 21, 50-56.

Funk, K., Lyall, B., Riley, V. (1995). Perceived human factors problems of flightdeck automation. Phase 1 final report, a comparative analysis of flightdecks with varying levels of automation, Federal Aviation Administration, Grant 93-G-039.

Funk, K., Braune, R. (in press). Expanding the functionality of existing airframe systems monitors: the agendamanager. Proceedings of the Ninth International Symposium on Aviation Psychology.

Funk, K., Cha ,W. C., Wilson, R., Zaspel, J., Braune, R. (in press). Development and evaluation of an aid to facilitate agenda management. Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics.

Gerlach, M. Onken, R., Prevot T., Ruckdeschel, W. (1995). The cockpit assistant system cassy - design and in-flight evaluation. Advances in Human Factors Ergonomics, 20, 759-764.

Hammer, J. M. (1984). An intelligent flight management aid for procedure execution. IEEE Transactions on Systems, Man, and Cybernetics, 14(6), 885-888

Harrison, L. H., & Saunders, P. J. (1993). Artificial intelligence and expert systems for avionics. AIAA/IEEE Digital Avionics Systems Conference, Fort Worth, 12, 157-172.

Hirouta, K. (1995). History of industrial applications of fuzzy logic in Japan. In J. Yen, R. Langari, & L. A. Zadeh, (Eds.), Industrial applications of fuzzy logic and intelligent systems (pp. 43-54). New York: IEEE, Inc.

Kantowitz, B. H., Casper, P. A. (1988). Human workload in aviation. In E. L. Wiener & D. C. Nagel (Eds.), Human factors in aviation (pp. 111-155). Academic Press.

Langari, R., & Yen, J. (1995) Introduction to fuzzy logic control. In J. Yen, R. Langari, & L. A. Zadeh, (Eds.), Industrial applications of fuzzy logic and intelligent systems (pp. 3 -39). New York: IEEE, Inc.

Linkens, D. A., Nyongesa, H. O. (1995). Genetic algorithms for fuzzy control, part 1: Offline system development and application. IEE Proceedings of Control Theory Applications, 142(3), 161-176.

Linkens, D. A., & Nyongesa, H. O. (1996). Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications. IEEE Proceedings - Control Theory and Applications, 143, 367-386.

Mamdani, E. H., & Assilian S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 7, 1-13.

Mendel, H. M. (1995). Fuzzy logic systems for engineering: A tutorial. Proceedings of the IEEE, 83(3), 345-377.

Munakata, T., & Jani, Y. (1994). Fuzzy systems: an overview. Communications of the ACM, 37(3), 69-76.

National Transportation Safety Board (1972). Aircraft accident report, Eastern Air Lines, Inc., L-1011, N310EA, Miami, Florida, December 29, 1972 (NTSB-AAR-73-14).

National Transportation Safety Board (1979). Aircraft accident report, Aeromexico DC-10-30, XA-DUH, Over Luxembourg, Europe, November 11, 1979 (NTSB-AAR-80-10).

Nagel, D. C. (1988). Human error in aviation operations. In E. L. Wiener & D. C. Nagel (Eds.), Human factors in aviation (pp. 111-155). Academic Press.

Onken, R.(1992). New development in aerospace guidance and control: knowledge-based pilot assistance. IFAC Symposia Series, 12, 23-32.

Onken, R. (1995). Functional development and field test of cassy - a knowledge-based cockpit assistant system. AGARD Lecture Series 2000: Knowledge based guidance and control functions in aerospace systems, 4, 1-22.

Parasuraman, R. (1996). Monitoring of automated systems. In R. Parasuraman & M. Mouloua (Eds.), Automation and human performance – theory and applications (pp. 91-116). Mahwah, NJ: Lawrence Erlbaum Associates.

Pélegrin, M. J. (1992). Towards a fully automatic flight for passenger aircraft and space and earth reference systems. IFAC Automatic Control in Aerospace, 531-543,

Rouse, W. B., Geddes, N. D., Hammer, J. M. (1990). Computer-aided fighter pilots. IEEE Spectrum, 3, 38-41.

Rudolph, F. M., Homoki, D. A., Sexton, G. A. (1990). Diverter - decision aiding for in-flight diversions (NASA Contractor Report 182070).

Scerpo, M. W. (1996). Theoretical perspectives on adaptive automation. In R. Parasuraman & M. Mouloua (Eds.), Automation and human performance – theory and applications (pp. 37-64). Mahwah, NJ: Lawrence Erlbaum Associates.

Search Technology (1995). A prototype flight management systems error monitor. (STI-TR-9308-001).

Sexton, G. A. (1988). Cockpit Crew Systems Design and Integration. In E. L. Wiener & D. C. Nagel (Eds.), Human factors in aviation (pp. 111-155). Academic Press.

Sugeno, M. (1985). An introductory survey of fuzzy control. Information Sciences, 36, 59-83.

Sweet, W. (1995). The glass cockpit. IEEE Spectrum, 10, 31-38.

Takagi, H. (1993). Fusion techniques of fuzzy systems and neural networks, and fuzzy systems and genetic algorithms. Proceedings of SPIE – The International Society for Optical Engineering, 2061, 402-413.

Wickens, C. D., Flach, J. M. (1988). Information Processing. In E. L. Wiener & D. C. Nagel (Eds.), Human factors in aviation (pp. 111-155). Academic Press.

Wiener E. L. (1980). Flight-deck automation: promises and problems. Ergonomics, 23(10), 995-1011.

Wiener E. L. (1983). Computers in the cockpit: but what about the pilots? Second Aerospace Behavioral Engineering Technology Conference Proceedings, 453-458.

Wiener E. L. (1985a). Cockpit automation: in need of a philosophy. Proceedings of 1985 Behavioral Engineering Conference, Warrendale, PA: Society of Automotive Engineers.

Wiener E. L. (1985b). Beyond the sterile cockpit. Human Factors, 27(1), 75-90.

Wiener, E. L. (1988). Cockpit automation. In E. L. Wiener & D. C. Nagel (Eds.), Human factors in aviation (pp. 111-155). Academic Press.

Wiener E. L. (1989). Errors and error management in high technology aircraft. The 7th Meeting on Aeronautic and Space Medicine, 1-9.

Woods, D. D., Potter, S. S., Johannesen, L., Holloway, M. (1991). Human interaction with intelligent systems: volume I -- trends, problems, new directions. NASA CSEL Report 1991-001, 133-206.

Yasunobu, S., Miyamoto, S., & Ihara, H. (1983). Fuzzy control for automatic train operation system. Proceedings of the 4th IFAC/IFIP/IFORS International Congress on Control in Transportation Systems, Baden-Baden.

Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8, 338-353.

Zadeh, L. A. (1975a). The concept of a linguistic variable and its application to approximate reasoning – 1. Information Sciences, 8,199-249.

Zadeh, L. A. (1975b). The concept of a linguistic variable and its application to approximate reasoning – 2. Information Sciences, 8, 301-357.

Zadeh, L. A. (1975c). The concept of a linguistic variable and its application to approximate reasoning – 3. Information Sciences, 9, 43-80.

Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. Communications of the ACM, 37(3), 77-84.

Zimmermann, H.-J. (1991). Fuzzy set theory and its applications. Boston: Kluver Academic Publisher.

APPENDICES

APPENDIX A – Defintion of Fuzzy Systems

APPENDIX A – Defintion of Fuzzy Systems

Table A.1 Initial FOHS 1

name:	**current airspeed (ca)**			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 120.0	right: 150.0	
slow:	type: #lamda	typical: 150.0	left: 120.0	right: 180.0
medium:	type: #lamda	typical: 180	left: 150.0	right: 340.0
fast:	type: #s	typical: 340.0	left: 180.0	

name:	**current airspeed acceleration (caa)**			
type:	input			
terms:	negative, medium, positive			
negative:	type: #z	typical: -5.0	right: 0.0	
medium:	type: #lamda	typical: 0.0	left: -5.0	right: 5.0
positive:	type: #s	typical: 5.0	left: 0.0	

name:	**overspeed hazard (oh)**			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.1	right: 2.5	
low:	type: #lamda	typical: 2.5	left: 0.1	right: 6.0
high:	type: #lamda	typical: 6	left: 2.5	right: 9.9
veryHigh:	type: #s	typical: 9.9	left: 6.0	

Block: overspeed hazard					
Rule#	**ca**	**caa**	**oh**	**gamma**	**dos**
1	verySlow	negative	veryLow	nil	0.5
2	verySlow	negative	veryLow	nil	0.5
3	verySlow	medium	veryLow	nil	0.5
4	verySlow	medium	veryLow	nil	0.5
5	verySlow	positive	veryLow	nil	0.5
6	verySlow	positive	veryLow	nil	0.5
7	slow	negative	veryLow	nil	0.5
8	slow	negative	veryLow	nil	0.5
9	slow	medium	veryLow	nil	0.5
10	slow	medium	veryLow	nil	0.5
11	slow	positive	veryLow	nil	0.5
12	slow	positive	low	nil	0.5
13	medium	negative	veryLow	nil	0.9
14	medium	negative	low	nil	0.1
15	medium	medium	veryLow	nil	0.2
16	medium	medium	low	nil	0.8
17	medium	positive	low	nil	0.5
18	medium	positive	high	nil	0.5
19	fast	negative	low	nil	0.9
20	fast	negative	high	nil	0.1
21	fast	medium	high	nil	0.5
22	fast	medium	veryHigh	nil	0.5
23	fast	positive	veryHigh	nil	0.9
24	fast	positive	high	nil	0.1

Table A.2 Initial FOHS 2

name:	**current airspeed (ca)**			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 120.0	right: 150.0	
slow:	type: #lamda	typical: 150.0	left: 120.0	right: 180.0
medium:	type: #lamda	typical: 180	left: 150.0	right: 340.0
fast:	type: #s	typical: 340.0	left: 180.0	

name:	**current airspeed acceleration (caa)**			
type:	input			
terms:	negative, medium, positive			
negative:	type: #z	typical: -5.0	right: 0.0	

Table A.2 (Continued)

medium:	type: #lamda	typical: 0.0	left: -5.0	right: 5.0
positive:	type: #s	typical: 5.0	left: 0.0	

name:	**current thrust (ct)**			
type:	input			
terms:	low, medium, high			
low:	type: #z	typical: 0.01	right: 0.5	
medium:	type: #lamda	typical: 0.5	left: 0.01	right: 0.99
high:	type: #s	typical: 0.99	left: 0.5	

name:	**future airspeed (fa)**
type:	intermediate
terms:	verySlow, slow, medium, fast

name:	**current flight path angle (cfpa)**			
type:	input			
terms:	negative, negativeSmall, positiveSmall, positive			
negative:	type: #z	typical: -8	right: -2	
negativeSmall:	type: #lamda	typical: -2	left: -8.0	right: 2
positiveSmall:	type: #lamda	typical: 2	left: -2.0	right: 20
positive:	type: #s	typical: 20.0	left: 2.0	

name:	**overspeed hazard (oh)**			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.1	right: 2.5	
low:	type: #lamda	typical: 2.5	left: 0.1	right: 6.0
high:	type: #lamda	typical: 6	left: 2.5	right: 9.9
veryHigh:	type: #s	typical: 9.9	left: 6.0	

Block: future airspeed						
Rule#	**ca**	**caa**	**ct**	**fa**	**gamma**	**dos**
1	verySlow	negative	low	verySlow	nil	1
2	verySlow	negative	low	verySlow	nil	1
3	verySlow	negative	medium	verySlow	nil	1
4	verySlow	negative	medium	verySlow	nil	1
5	verySlow	negative	high	verySlow	nil	1
6	verySlow	negative	high	slow	nil	1
7	verySlow	medium	low	verySlow	nil	1
8	verySlow	medium	low	slow	nil	1
9	verySlow	medium	medium	slow	nil	1
10	verySlow	medium	medium	medium	nil	1
11	verySlow	medium	high	slow	nil	1
12	verySlow	medium	high	medium	nil	1
13	verySlow	positive	low	slow	nil	1
14	verySlow	positive	low	verySlow	nil	1
15	verySlow	positive	medium	slow	nil	1
16	verySlow	positive	medium	slow	nil	1
17	verySlow	positive	high	slow	nil	1
18	verySlow	positive	high	medium	nil	1
19	slow	negative	low	verySlow	nil	1
20	slow	negative	low	verySlow	nil	1
21	slow	negative	medium	slow	nil	1
22	slow	negative	medium	verySlow	nil	1
23	slow	negative	high	slow	nil	1
24	slow	negative	high	verySlow	nil	1
25	slow	medium	low	slow	nil	1
26	slow	medium	low	verySlow	nil	1
27	slow	medium	medium	slow	nil	1
28	slow	medium	medium	slow	nil	1
29	slow	medium	high	slow	nil	1
30	slow	medium	high	medium	nil	1
31	slow	positive	low	medium	nil	1
32	slow	positive	low	medium	nil	1
33	slow	positive	medium	medium	nil	1
34	slow	positive	medium	medium	nil	1
35	slow	positive	high	medium	nil	1
36	slow	positive	high	medium	nil	1
37	medium	negative	low	slow	nil	1
38	medium	negative	low	slow	nil	1
39	medium	negative	medium	slow	nil	1
40	medium	negative	medium	medium	nil	1

Table A.2 (Continued)

41	medium	negative	high	slow	nil	1
42	medium	negative	high	medium	nil	1
43	medium	medium	low	medium	nil	1
44	medium	medium	low	slow	nil	1
45	medium	medium	medium	medium	nil	1
46	medium	medium	medium	medium	nil	1
47	medium	medium	high	medium	nil	1
48	medium	medium	high	fast	nil	1
49	medium	positive	low	medium	nil	1
50	medium	positive	low	medium	nil	1
51	medium	positive	medium	fast	nil	1
52	medium	positive	medium	medium	nil	1
53	medium	positive	high	fast	nil	1
54	medium	positive	high	medium	nil	1
55	fast	negative	low	medium	nil	1
56	fast	negative	low	medium	nil	1
57	fast	negative	medium	medium	nil	1
58	fast	negative	medium	fast	nil	1
59	fast	negative	high	medium	nil	1
60	fast	negative	high	fast	nil	1
61	fast	medium	low	fast	nil	1
62	fast	medium	low	medium	nil	1
63	fast	medium	medium	fast	nil	1
64	fast	medium	medium	fast	nil	1
65	fast	medium	high	fast	nil	1
66	fast	medium	high	fast	nil	1
67	fast	positive	low	fast	nil	1
68	fast	positive	low	fast	nil	1
69	fast	positive	medium	fast	nil	1
70	fast	positive	medium	fast	nil	1
71	fast	positive	high	fast	nil	1
72	fast	positive	high	fast	nil	1

Block: overspeed hazard					
Rule#	fa	cfpa	oh	gamma	dos
1	verySlow	negative	veryLow	nil	0.9
2	verySlow	negative	low	nil	0.1
3	verySlow	negativeSmall	veryLow	nil	0.5
4	verySlow	negativeSmall	veryLow	nil	0.5
5	verySlow	positiveSmall	veryLow	nil	0.5
6	verySlow	positiveSmall	veryLow	nil	0.5
7	verySlow	positive	veryLow	nil	0.5
8	verySlow	positive	veryLow	nil	0.5
9	slow	negative	low	nil	0.7
10	slow	negative	veryLow	nil	0.3
11	slow	negativeSmall	low	nil	0.6
12	slow	negativeSmall	veryLow	nil	0.4
13	slow	positiveSmall	low	nil	0.5
14	slow	positiveSmall	veryLow	nil	0.5
15	slow	positive	low	nil	0.2
16	slow	positive	veryLow	nil	0.8
17	medium	negative	low	nil	0.2
18	medium	negative	high	nil	0.8
19	medium	negativeSmall	low	nil	0.3
20	medium	negativeSmall	high	nil	0.7
21	medium	positiveSmall	low	nil	0.5
22	medium	positiveSmall	high	nil	0.5
23	medium	positive	low	nil	0.6
24	medium	positive	high	nil	0.4
25	fast	negative	veryHigh	nil	0.9
26	fast	negative	high	nil	0.1
27	fast	negativeSmall	veryHigh	nil	0.8
28	fast	negativeSmall	high	nil	0.2
29	fast	positiveSmall	veryHigh	nil	0.7
30	fast	positiveSmall	high	nil	0.3
31	fast	positive	veryHigh	nil	0.5
32	fast	positive	high	nil	0.5

Table A.3 Initial FSHS 1

name:	current flight path angle (cfpa)			
type:	input			
terms:	negative, negativeSmall, positiveSmall, positive			
negative:	type: #z	typical: -8	right: -2	
negativeSmall:	type: #lamda	typical: -2	left: -8.0	right: 2
positiveSmall:	type: #lamda	typical: 2	left: -2.0	right: 20
positive:	type: #s	typical: 20.0	left: 2.0	

name:	current airspeed (ca)			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 120.0	right: 150.0	
slow:	type: #lamda	typical: 150.0	left: 120.0	right: 180.0
medium:	type: #lamda	typical: 180	left: 150.0	right: 340.0
fast:	type: #s	typical: 340.0	left: 180.0	

name:	stall hazard (sh)			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.1	right: 2.5	
low:	type: #lamda	typical: 2.5	left: 0.1	right: 6.0
high:	type: #lamda	typical: 6	left: 2.5	right: 9.9
veryHigh:	type: #s	typical: 9.9	left: 6.0	

Block: stall hazard					
Rule#	cfpa	ca	sh	gamma	dos
1	negative	verySlow	high	nil	1
2	negative	verySlow	high	nil	1
3	negative	slow	high	nil	1
4	negative	slow	low	nil	1
5	negative	medium	low	nil	1
6	negative	medium	veryLow	nil	1
7	negative	fast	veryLow	nil	1
8	negative	fast	veryLow	nil	1
9	negativeSmall	verySlow	high	nil	1
10	negativeSmall	verySlow	veryHigh	nil	1
11	negativeSmall	slow	low	nil	1
12	negativeSmall	slow	high	nil	1
13	negativeSmall	medium	low	nil	1
14	negativeSmall	medium	low	nil	1
15	negativeSmall	fast	veryLow	nil	1
16	negativeSmall	fast	veryLow	nil	1
17	positiveSmall	verySlow	veryHigh	nil	1
18	positiveSmall	verySlow	low	nil	1
19	positiveSmall	slow	high	nil	1
20	positiveSmall	slow	low	nil	1
21	positiveSmall	medium	low	nil	1
22	positiveSmall	medium	low	nil	1
23	positiveSmall	fast	veryLow	nil	1
24	positiveSmall	fast	low	nil	1
25	positive	verySlow	veryHigh	nil	1
26	positive	verySlow	veryHigh	nil	1
27	positive	slow	veryHigh	nil	1
28	positive	slow	high	nil	1
29	positive	medium	high	nil	1
30	positive	medium	low	nil	1
31	positive	fast	low	nil	1
32	positive	fast	low	nil	1

Table A.4 Initial FSHS 2

name:	current flight path angle (cfpa)			
type:	input			
terms:	negative, negativeSmall, positiveSmall, positive			
negative:	type: #z	typical: -8	right: -2	
negativeSmall:	type: #lamda	typical: -2	left: -8.0	right: 2
positiveSmall:	type: #lamda	typical: 2	left: -2.0	right: 20
positive:	type: #s	typical: 20.0	left: 2.0	

Table A.4 (Continued)

name:	**current airspeed (ca)**			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 120.0	right: 150.0	
slow:	type: #lamda	typical: 150.0	left: 120.0	right: 180.0
medium:	type: #lamda	typical: 180	left: 150.0	right: 340.0
fast:	type: #s	typical: 340.0	left: 180.0	

name:	**current airspeed acceleration (caa)**			
type:	input			
terms:	negative, medium, positive			
negative:	type: #z	typical: -5.0	right: 0.0	
medium:	type: #lamda	typical: 0.0	left: -5.0	right: 5.0
positive:	type: #s	typical: 5.0	left: 0.0	

name:	**current thrust (ct)**			
type:	input			
terms:	low, medium, high			
low:	type: #z	typical: 0.01	right: 0.5	
medium:	type: #lamda	typical: 0.5	left: 0.01	right: 0.99
high:	type: #s	typical: 0.99	left: 0.5	

name:	**future airspeed (fa)**
type:	intermediate
terms:	verySlow, slow, medium, fast

name:	**stall hazard (sh)**			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.1	right: 2.5	
low:	type: #lamda	typical: 2.5	left: 0.1	right: 6.0
high:	type: #lamda	typical: 6	left: 2.5	right: 9.9
veryHigh:	type: #s	typical: 9.9	left: 6.0	

Block: future airspeed						
Rule#	**ca**	**caa**	**ct**	**fa**	**gamma**	**dos**
1	verySlow	negative	low	verySlow	nil	1
2	verySlow	negative	low	verySlow	nil	1
3	verySlow	negative	medium	verySlow	nil	1
4	verySlow	negative	medium	verySlow	nil	1
5	verySlow	negative	high	verySlow	nil	1
6	verySlow	negative	high	slow	nil	1
7	verySlow	medium	low	verySlow	nil	1
8	verySlow	medium	low	slow	nil	1
9	verySlow	medium	medium	slow	nil	1
10	verySlow	medium	medium	medium	nil	1
11	verySlow	medium	high	slow	nil	1
12	verySlow	medium	high	medium	nil	1
13	verySlow	positive	low	slow	nil	1
14	verySlow	positive	low	verySlow	nil	1
15	verySlow	positive	medium	slow	nil	1
16	verySlow	positive	medium	slow	nil	1
17	verySlow	positive	high	slow	nil	1
18	verySlow	positive	high	medium	nil	1
19	slow	negative	low	verySlow	nil	1
20	slow	negative	low	verySlow	nil	1
21	slow	negative	medium	slow	nil	1
22	slow	negative	medium	verySlow	nil	1
23	slow	negative	high	slow	nil	1
24	slow	negative	high	verySlow	nil	1
25	slow	medium	low	slow	nil	1
26	slow	medium	low	verySlow	nil	1
27	slow	medium	medium	slow	nil	1
28	slow	medium	medium	slow	nil	1
29	slow	medium	high	slow	nil	1
30	slow	medium	high	medium	nil	1
31	slow	positive	low	medium	nil	1
32	slow	positive	low	medium	nil	1
33	slow	positive	medium	medium	nil	1
34	slow	positive	medium	medium	nil	1
35	slow	positive	high	medium	nil	1

Table A.4 (Continued)

36	slow	positive	high	medium	nil	1
37	medium	negative	low	slow	nil	1
38	medium	negative	low	slow	nil	1
39	medium	negative	medium	slow	nil	1
40	medium	negative	medium	medium	nil	1
41	medium	negative	high	slow	nil	1
42	medium	negative	high	medium	nil	1
43	medium	medium	low	medium	nil	1
44	medium	medium	low	slow	nil	1
45	medium	medium	medium	medium	nil	1
46	medium	medium	medium	medium	nil	1
47	medium	medium	high	medium	nil	1
48	medium	medium	high	fast	nil	1
49	medium	positive	low	medium	nil	1
50	medium	positive	low	medium	nil	1
51	medium	positive	medium	fast	nil	1
52	medium	positive	medium	medium	nil	1
53	medium	positive	high	fast	nil	1
54	medium	positive	high	medium	nil	1
55	fast	negative	low	medium	nil	1
56	fast	negative	low	medium	nil	1
57	fast	negative	medium	medium	nil	1
58	fast	negative	medium	fast	nil	1
59	fast	negative	high	medium	nil	1
60	fast	negative	high	fast	nil	1
61	fast	medium	low	fast	nil	1
62	fast	medium	low	medium	nil	1
63	fast	medium	medium	fast	nil	1
64	fast	medium	medium	fast	nil	1
65	fast	medium	high	fast	nil	1
66	fast	medium	high	fast	nil	1
67	fast	positive	low	fast	nil	1
68	fast	positive	low	fast	nil	1
69	fast	positive	medium	fast	nil	1
70	fast	positive	medium	fast	nil	1
71	fast	positive	high	fast	nil	1
72	fast	positive	high	fast	nil	1

Block: current stall hazard					
Rule#	**cfpa**	**fa**	**sh**	**gamma**	**dos**
1	negative	verySlow	high	nil	1
2	negative	verySlow	veryHigh	nil	1
3	negative	slow	high	nil	1
4	negative	slow	low	nil	1
5	negative	medium	low	nil	1
6	negative	medium	veryLow	nil	1
7	negative	fast	low	nil	1
8	negative	fast	veryLow	nil	1
9	negativeSmall	verySlow	veryHigh	nil	1
10	negativeSmall	verySlow	high	nil	1
11	negativeSmall	slow	high	nil	1
12	negativeSmall	slow	high	nil	1
13	negativeSmall	medium	low	nil	1
14	negativeSmall	medium	high	nil	1
15	negativeSmall	fast	low	nil	1
16	negativeSmall	fast	veryLow	nil	1
17	positiveSmall	verySlow	veryHigh	nil	1
18	positiveSmall	verySlow	veryHigh	nil	1
19	positiveSmall	slow	veryHigh	nil	1
20	positiveSmall	slow	high	nil	1
21	positiveSmall	medium	low	nil	1
22	positiveSmall	medium	high	nil	1
23	positiveSmall	fast	veryLow	nil	1
24	positiveSmall	fast	low	nil	1
25	positive	verySlow	veryHigh	nil	1
26	positive	verySlow	veryHigh	nil	1
27	positive	slow	veryHigh	nil	1
28	positive	slow	high	nil	1
29	positive	medium	high	nil	1
30	positive	medium	low	nil	1

Table A.4 (Continued)

31	positive	fast	low	nil	1
32	positive	fast	veryLow	nil	1

Table A.5 Initial FTHS 1

name:	**current vertical speed (cvs)**			
type:	input			
terms:	veryNegative, negative, positive, veryPositive			
veryNegative:	type: #z	typical: -6000	right: -3000	
negative:	type: #lamda	typical: -3000	left: -6000	right: 3000
positive:	type: #lamda	typical: 3000	left: -3000	right: 6000
veryPositive:	type: #s	typical: 6000	left: 3000	

name:	**current altitude (calt)**			
type:	input			
terms:	extremelyLow	veryLow	low	medium
extremelyLow:	type: #z	typical: 0.0	right: 500.0	
veryLow:	type: #lamda	typical: 500	left: 0	right: 1000
low:	type: #lamda	typical: 1000	left: 500	right: 2500
medium:	type: #s	typical: 2500	left: 1000	

name:	**terrain hazard (th)**			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.1	right: 2.5	
low:	type: #lamda	typical: 2.5	left: 0.1	right: 6.0
high:	type: #lamda	typical: 6	left: 2.5	right: 9.9
veryHigh:	type: #s	typical: 9.9	left: 6.0	

Block: terrain hazard					
Rule#	**cvs**	**calt**	**th**	**gamma**	**dos**
1	veryNegative	extremelyLow	veryHigh	nil	1
2	veryNegative	extremelyLow	veryHigh	nil	1
3	veryNegative	veryLow	high	nil	1
4	veryNegative	veryLow	veryHigh	nil	1
5	veryNegative	low	high	nil	1
6	veryNegative	low	high	nil	1
7	veryNegative	medium	high	nil	1
8	veryNegative	medium	low	nil	1
9	negative	extremelyLow	high	nil	1
10	negative	extremelyLow	veryHigh	nil	1
11	negative	veryLow	high	nil	1
12	negative	veryLow	veryHigh	nil	1
13	negative	low	low	nil	1
14	negative	low	low	nil	1
15	negative	medium	low	nil	1
16	negative	medium	veryLow	nil	1
17	positive	extremelyLow	low	nil	1
18	positive	extremelyLow	high	nil	1
19	positive	veryLow	low	nil	1
20	positive	veryLow	veryLow	nil	1
21	positive	low	low	nil	1
22	positive	low	veryLow	nil	1
23	positive	medium	veryLow	nil	1
24	positive	medium	veryLow	nil	1
25	veryPositive	extremelyLow	high	nil	1
26	veryPositive	extremelyLow	high	nil	1
27	veryPositive	veryLow	high	nil	1
28	veryPositive	veryLow	veryHigh	nil	1
29	veryPositive	low	low	nil	1
30	veryPositive	low	high	nil	1
31	veryPositive	medium	low	nil	1
32	veryPositive	medium	low	nil	1

Table A.6 Initial FTHS 2

name:	**current flight path angle (cfpa)**			
type:	input			
terms:	negative, negativeSmall, positiveSmall			
negative:	type: #z	typical: -8	right: -2	
negativeSmall:	type: #lamda	typical: -2	left: -8.0	right: 2
positiveSmall:	type: #lamda	typical: 2	left: -2.0	right: 20
positive:	type: #s	typical: 20.0	left: 2.0	

name:	**current airspeed (ca)**			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 120.0	right: 150.0	
slow:	type: #lamda	typical: 150.0	left: 120.0	right: 180.0
medium:	type: #lamda	typical: 180	left: 150.0	right: 340.0
fast:	type: #s	typical: 340.0	left: 180.0	

name:	**current thrust (ct)**			
type:	input			
terms:	low, medium, high			
low:	type: #z	typical: 0.01	right: 0.5	
medium:	type: #lamda	typical: 0.5	left: 0.01	right: 0.99
high:	type: #s	typical: 0.99	left: 0.5	

name:	**attitude (at)**
type:	intermediate
terms:	diving, descending, climbing, climbingExtreme

name:	**current altitude (calt)**			
type:	input			
terms:	extremelyLow, veryLow, low, medium			
extremelyLow:	type: #z	typical: 0.0	right: 500.0	
veryLow:	type: #lamda	typical: 500	left: 0	right: 1000
low:	type: #lamda	typical: 1000	left: 500	right: 2500
medium:	type: #s	typical: 2500	left: 1000	

name:	**terrain hazard (th)**			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.1	right: 2.5	
low:	type: #lamda	typical: 2.5	left: 0.1	right: 6.0
high:	type: #lamda	typical: 6	left: 2.5	right: 9.9
veryHigh:	type: #s	typical: 9.9	left: 6.0	

name:	**current altitude error (cae)**				
type:	input				
terms:	negative, negativeSmall, positiveSmall, positive				
negative:	type: #z	typical: -15000	right: -5000		
negativeSmall:	type: #pi	typical: -5000	typicalEnd: -100	left: -15000.0	right: 0
positiveSmall:	type: #lamda	typical: 500	left: -100.0	right: 1000	
positive:	type: #s	typical: 1000.0	left: 500.0		

Block: attitude						
Rule#	**cfpa**	**ca**	**ct**	**at**	**gamma**	**dos**
1	negative	verySlow	low	descending	nil	0.5
2	negative	verySlow	low	descending	nil	0.5
3	negative	verySlow	medium	descending	nil	0.5
4	negative	verySlow	medium	descending	nil	0.5
5	negative	verySlow	high	descending	nil	0.5
6	negative	verySlow	high	descending	nil	0.5
7	negative	slow	low	descending	nil	0.5
8	negative	slow	low	descending	nil	0.5
9	negative	slow	medium	descending	nil	0.5
10	negative	slow	medium	descending	nil	0.5
11	negative	slow	high	descending	nil	0.5
12	negative	slow	high	descending	nil	0.5
13	negative	medium	low	descending	nil	0.5
14	negative	medium	low	descending	nil	0.5
15	negative	medium	medium	descending	nil	0.5
16	negative	medium	medium	descending	nil	0.5
17	negative	medium	high	descending	nil	0.8
18	negative	medium	high	diving	nil	0.2
19	negative	fast	low	descending	nil	0.5
20	negative	fast	low	diving	nil	0.5

Table A.6 (Continued)

21	negative	fast	medium	descending	nil	0.1
22	negative	fast	medium	diving	nil	0.9
23	negative	fast	high	diving	nil	0.5
24	negative	fast	high	diving	nil	0.5
25	negativeSmall	verySlow	low	descending	nil	0.5
26	negativeSmall	verySlow	low	descending	nil	0.5
27	negativeSmall	verySlow	medium	descending	nil	0.5
28	negativeSmall	verySlow	medium	descending	nil	0.5
29	negativeSmall	verySlow	high	descending	nil	0.5
30	negativeSmall	verySlow	high	descending	nil	0.5
31	negativeSmall	slow	low	descending	nil	0.5
32	negativeSmall	slow	low	descending	nil	0.5
33	negativeSmall	slow	medium	descending	nil	0.5
34	negativeSmall	slow	medium	descending	nil	0.5
35	negativeSmall	slow	high	descending	nil	0.5
36	negativeSmall	slow	high	descending	nil	0.5
37	negativeSmall	medium	low	descending	nil	0.5
38	negativeSmall	medium	low	descending	nil	0.5
39	negativeSmall	medium	medium	descending	nil	0.5
40	negativeSmall	medium	medium	descending	nil	0.5
41	negativeSmall	medium	high	descending	nil	0.5
42	negativeSmall	medium	high	descending	nil	0.5
43	negativeSmall	fast	low	descending	nil	0.5
44	negativeSmall	fast	low	descending	nil	0.5
45	negativeSmall	fast	medium	diving	nil	0.5
46	negativeSmall	fast	medium	descending	nil	0.5
47	negativeSmall	fast	high	diving	nil	0.5
48	negativeSmall	fast	high	descending	nil	0.5
49	positiveSmall	verySlow	low	climbing	nil	0.5
50	positiveSmall	verySlow	low	climbingExtreme	nil	0.5
51	positiveSmall	verySlow	medium	climbing	nil	0.8
52	positiveSmall	verySlow	medium	climbingExtreme	nil	0.2
53	positiveSmall	verySlow	high	climbing	nil	0.5
54	positiveSmall	verySlow	high	climbing	nil	0.5
55	positiveSmall	slow	low	climbing	nil	0.5
56	positiveSmall	slow	low	climbing	nil	0.5
57	positiveSmall	slow	medium	climbing	nil	0.5
58	positiveSmall	slow	medium	climbing	nil	0.5
59	positiveSmall	slow	high	climbing	nil	0.5
60	positiveSmall	slow	high	climbing	nil	0.5
61	positiveSmall	medium	low	climbing	nil	0.5
62	positiveSmall	medium	low	climbing	nil	0.5
63	positiveSmall	medium	medium	climbing	nil	0.5
64	positiveSmall	medium	medium	climbing	nil	0.5
65	positiveSmall	medium	high	climbing	nil	0.5
66	positiveSmall	medium	high	climbing	nil	0.5
67	positiveSmall	fast	low	climbing	nil	0.5
68	positiveSmall	fast	low	climbing	nil	0.5
69	positiveSmall	fast	medium	climbing	nil	0.5
70	positiveSmall	fast	medium	climbing	nil	0.5
71	positiveSmall	fast	high	climbing	nil	0.5
72	positiveSmall	fast	high	climbing	nil	0.5
73	positive	verySlow	low	climbingExtreme	nil	0.5
74	positive	verySlow	low	climbingExtreme	nil	0.5
75	positive	verySlow	medium	climbingExtreme	nil	0.5
76	positive	verySlow	medium	climbing	nil	0.5
77	positive	verySlow	high	climbingExtreme	nil	0.2
78	positive	verySlow	high	climbing	nil	0.8
79	positive	slow	low	climbing	nil	0.9
80	positive	slow	low	climbingExtreme	nil	0.1
81	positive	slow	medium	climbing	nil	0.5
82	positive	slow	medium	climbing	nil	0.5
83	positive	slow	high	climbing	nil	0.5
84	positive	slow	high	climbing	nil	0.5
85	positive	medium	low	climbing	nil	0.5
86	positive	medium	low	climbing	nil	0.5
87	positive	medium	medium	climbing	nil	0.5
88	positive	medium	medium	climbing	nil	0.5
89	positive	medium	high	climbing	nil	0.5
90	positive	medium	high	climbing	nil	0.5
91	positive	fast	low	climbing	nil	0.5

Table A.6 (Continued)

92	positive	fast	low	climbing	nil	0.5
93	positive	fast	medium	climbing	nil	0.5
94	positive	fast	medium	climbing	nil	0.5
95	positive	fast	high	climbing	nil	0.5
96	positive	fast	high	climbing	nil	0.5

Block: terrain hazard						
Rule#	at	ca	cae	th	gamma	dos
1	diving	extremelyLow	negative	veryHigh	nil	0.5
2	diving	extremelyLow	negative	veryHigh	nil	0.5
3	diving	extremelyLow	negativeSmall	veryHigh	nil	0.5
4	diving	extremelyLow	negativeSmall	veryHigh	nil	0.5
5	diving	extremelyLow	positiveSmall	veryHigh	nil	0.5
6	diving	extremelyLow	positiveSmall	veryHigh	nil	0.5
7	diving	extremelyLow	positive	veryHigh	nil	0.5
8	diving	extremelyLow	positive	veryHigh	nil	0.5
9	diving	veryLow	negative	veryHigh	nil	0.5
10	diving	veryLow	negative	veryHigh	nil	0.5
11	diving	veryLow	negativeSmall	veryHigh	nil	0.5
12	diving	veryLow	negativeSmall	high	nil	0.5
13	diving	veryLow	positiveSmall	high	nil	0.5
14	diving	veryLow	positiveSmall	high	nil	0.5
15	diving	veryLow	positive	high	nil	0.5
16	diving	veryLow	positive	high	nil	0.5
17	diving	low	negative	high	nil	0.5
18	diving	low	negative	high	nil	0.5
19	diving	low	negativeSmall	high	nil	0.5
20	diving	low	negativeSmall	high	nil	0.5
21	diving	low	positiveSmall	high	nil	0.5
22	diving	low	positiveSmall	low	nil	0.5
23	diving	low	positive	high	nil	0.5
24	diving	low	positive	low	nil	0.5
25	diving	medium	negative	high	nil	0.5
26	diving	medium	negative	high	nil	0.5
27	diving	medium	negativeSmall	high	nil	0.5
28	diving	medium	negativeSmall	high	nil	0.5
29	diving	medium	positiveSmall	high	nil	0.5
30	diving	medium	positiveSmall	low	nil	0.5
31	diving	medium	positive	high	nil	0.5
32	diving	medium	positive	low	nil	0.5
33	descending	extremelyLow	negative	high	nil	0.5
34	descending	extremelyLow	negative	high	nil	0.5
35	descending	extremelyLow	negativeSmall	high	nil	0.8
36	descending	extremelyLow	negativeSmall	low	nil	0.2
37	descending	extremelyLow	positiveSmall	low	nil	0.7
38	descending	extremelyLow	positiveSmall	high	nil	0.3
39	descending	extremelyLow	positive	low	nil	0.5
40	descending	extremelyLow	positive	low	nil	0.5
41	descending	veryLow	negative	low	nil	0.8
42	descending	veryLow	negative	high	nil	0.2
43	descending	veryLow	negativeSmall	low	nil	0.9
44	descending	veryLow	negativeSmall	high	nil	0.1
45	descending	veryLow	positiveSmall	low	nil	0.5
46	descending	veryLow	positiveSmall	veryLow	nil	0.5
47	descending	veryLow	positive	low	nil	0.5
48	descending	veryLow	positive	veryLow	nil	0.5
49	descending	low	negative	low	nil	0.5
50	descending	low	negative	high	nil	0.5
51	descending	low	negativeSmall	low	nil	0.5
52	descending	low	negativeSmall	high	nil	0.5
53	descending	low	positiveSmall	low	nil	0.5
54	descending	low	positiveSmall	veryLow	nil	0.5
55	descending	low	positive	low	nil	0.5
56	descending	low	positive	veryLow	nil	0.5
57	descending	medium	negative	low	nil	0.5
58	descending	medium	negative	high	nil	0.5
59	descending	medium	negativeSmall	low	nil	0.5
60	descending	medium	negativeSmall	high	nil	0.5
61	descending	medium	positiveSmall	low	nil	0.5
62	descending	medium	positiveSmall	veryLow	nil	0.5
63	descending	medium	positive	low	nil	0.5

Table A.6 (Continued)

64	descending	medium	positive	veryLow	nil	0.5
65	climbing	extremelyLow	negative	veryLow	nil	0.5
66	climbing	extremelyLow	negative	veryLow	nil	0.5
67	climbing	extremelyLow	negativeSmall	veryLow	nil	0.5
68	climbing	extremelyLow	negativeSmall	veryLow	nil	0.5
69	climbing	extremelyLow	positiveSmall	veryLow	nil	0.5
70	climbing	extremelyLow	positiveSmall	veryLow	nil	0.5
71	climbing	extremelyLow	positive	veryLow	nil	0.5
72	climbing	extremelyLow	positive	veryLow	nil	0.5
73	climbing	veryLow	negative	veryLow	nil	0.5
74	climbing	veryLow	negative	veryLow	nil	0.5
75	climbing	veryLow	negativeSmall	veryLow	nil	0.5
76	climbing	veryLow	negativeSmall	veryLow	nil	0.5
77	climbing	veryLow	positiveSmall	veryLow	nil	0.5
78	climbing	veryLow	positiveSmall	veryLow	nil	0.5
79	climbing	veryLow	positive	veryLow	nil	0.5
80	climbing	veryLow	positive	veryLow	nil	0.5
81	climbing	low	negative	veryLow	nil	0.5
82	climbing	low	negative	veryLow	nil	0.5
83	climbing	low	negativeSmall	veryLow	nil	0.5
84	climbing	low	negativeSmall	veryLow	nil	0.5
85	climbing	low	positiveSmall	veryLow	nil	0.5
86	climbing	low	positiveSmall	veryLow	nil	0.5
87	climbing	low	positive	veryLow	nil	0.5
88	climbing	low	positive	veryLow	nil	0.5
89	climbing	medium	negative	veryLow	nil	0.5
90	climbing	medium	negative	veryLow	nil	0.5
91	climbing	medium	negativeSmall	veryLow	nil	0.5
92	climbing	medium	negativeSmall	veryLow	nil	0.5
93	climbing	medium	positiveSmall	veryLow	nil	0.5
94	climbing	medium	positiveSmall	veryLow	nil	0.5
95	climbing	medium	positive	veryLow	nil	0.5
96	climbing	medium	positive	veryLow	nil	0.5
97	climbingExtreme	extremelyLow	negative	high	nil	0.5
98	climbingExtreme	extremelyLow	negative	veryHigh	nil	0.5
99	climbingExtreme	extremelyLow	negativeSmall	high	nil	0.5
100	climbingExtreme	extremelyLow	negativeSmall	veryHigh	nil	0.5
101	climbingExtreme	extremelyLow	positiveSmall	high	nil	0.5
102	climbingExtreme	extremelyLow	positiveSmall	veryHigh	nil	0.5
103	climbingExtreme	extremelyLow	positive	high	nil	0.5
104	climbingExtreme	extremelyLow	positive	veryHigh	nil	0.5
105	climbingExtreme	veryLow	negative	high	nil	0.5
106	climbingExtreme	veryLow	negative	high	nil	0.5
107	climbingExtreme	veryLow	negativeSmall	high	nil	0.5
108	climbingExtreme	veryLow	negativeSmall	high	nil	0.5
109	climbingExtreme	veryLow	positiveSmall	high	nil	0.5
110	climbingExtreme	veryLow	positiveSmall	high	nil	0.5
111	climbingExtreme	veryLow	positive	high	nil	0.5
112	climbingExtreme	veryLow	positive	high	nil	0.5
113	climbingExtreme	low	negative	high	nil	0.5
114	climbingExtreme	low	negative	high	nil	0.5
115	climbingExtreme	low	negativeSmall	high	nil	0.5
116	climbingExtreme	low	negativeSmall	high	nil	0.5
117	climbingExtreme	low	positiveSmall	high	nil	0.5
118	climbingExtreme	low	positiveSmall	high	nil	0.5
119	climbingExtreme	low	positive	high	nil	0.5
120	climbingExtreme	low	positive	high	nil	0.5
121	climbingExtreme	medium	negative	high	nil	0.5
122	climbingExtreme	medium	negative	high	nil	0.5
123	climbingExtreme	medium	negativeSmall	high	nil	0.5
124	climbingExtreme	medium	negativeSmall	high	nil	0.5
125	climbingExtreme	medium	positiveSmall	high	nil	0.5
126	climbingExtreme	medium	positiveSmall	high	nil	0.5
127	climbingExtreme	medium	positive	high	nil	0.5
128	climbingExtreme	medium	positive	high	nil	0.5

Table A.7 Initial FHEIS

name:	**stall hazard (sh)**
type:	intermediate
terms:	veryLow, low, high, veryHigh

name:	**current error (ce)**
type:	intermediate
terms:	negativeLarge, negative, zero, positive, positiveLarge

name:	**past error (pe)**
type:	intermediate
terms:	negativeLarge, negative, zero, positive, positiveLarge

name:	**past error (pe)**
type:	intermediate
terms:	negativeLarge, negative, zero, positive, positiveLarge

name:	**past vertical speed (pvs)**			
type:	input			
terms:	veryNegative, negative, positive, veryPositive			
veryNegative:	type: #z	typical: -6000	right: -3000	
negative:	type: #lamda	typical: -3000	left: -6000	right: 3000
positive:	type: #lamda	typical: 3000	left: -3000	right: 6000
veryPositive:	type: #s	typical: 6000	left: 3000	

name:	**current vertical speed (cvs)**			
type:	input			
terms:	veryNegative, negative, positive, veryPositive			
veryNegative:	type: #z	typical: -6000	right: -3000	
negative:	type: #lamda	typical: -3000	left: -6000	right: 3000
positive:	type: #lamda	typical: 3000	left: -3000	right: 6000
veryPositive:	type: #s	typical: 6000	left: 3000	

name:	**past altitude error (pae)**				
type:	input				
terms:	negative, negativeSmall, zero, positiveSmall, positive				
negative:	type: #z	typical: -30000	right: -1000		
negativeSmall:	type: #pi	typical: -10000	typicalEnd: -1000	left: -30000.0	right: 0
zero:	type: #lamda	typical: 0	left: -1000	right: 1000.0	
positiveSmall:	type: #pi	typical: 1000	typicalEnd: 10000	left: 0	right: 30000
positive:	type: #s	typical: 30000.0	left: 1000.0		

name:	**overspeed hazard (oh)**
type:	intermediate
terms:	veryLow, low, high, veryHigh

name:	**current altitude error (cae)**				
type:	input				
terms:	negative, negativeSmall, zero, positiveSmall, positive				
negative:	type: #z	typical: -30000	right: -1000		
negativeSmall:	type: #pi	typical: -10000	typicalEnd: -1000	left: -30000.0	right: 0
zero:	type: #lamda	typical: 0	left: -1000	right: 1000.0	
positiveSmall:	type: #pi	typical: 1000	typicalEnd: 10000	left: 0	right: 30000
positive:	type: #s	typical: 30000.0	left: 1000.0		

name:	**current vertical speed acceleration (cvsa)**			
type:	input			
terms:	veryNegative, negative, positive, veryPositive			
veryNegative:	type: #z	typical: -2000	right: -500	
negative:	type: #lamda	typical: -500	left: -2000	right: 100.0
positive:	type: #lamda	typical: 100.0	left: -500.0	right: 450.0
veryPositive:	type: #s	typical: 450	left: 100.0	

name:	**overall hazard (h)**
type:	intermediate
terms:	veryLow, low, high, veryHigh

name:	**terrainHazard (th)**
type:	intermediate
terms:	veryLow, low, high, veryHigh

Table A.7 (Continued)

name:	**function performance (fp)**			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.1	right: 2.5	
low:	type: #lamda	typical: 2.5	left: 0.1	right: 6.0
high:	type: #lamda	typical: 6	left: 2.5	right: 9.9
veryHigh:	type: #s	typical: 9.9	left: 6.0	

Block: overall hazard						
Rule#	**sh**	**oh**	**th**	**h**	**gamma**	**dos**
1	veryLow	veryLow	veryLow	veryLow	nil	1
2	veryLow	veryLow	veryLow	veryLow	nil	1
3	veryLow	veryLow	low	veryLow	nil	0.8
4	veryLow	veryLow	low	low	nil	0.2
5	veryLow	veryLow	high	high	nil	0.8
6	veryLow	veryLow	high	low	nil	0.2
7	veryLow	veryLow	veryHigh	veryHigh	nil	0.8
8	veryLow	veryLow	veryHigh	high	nil	0.2
9	veryLow	low	veryLow	veryLow	nil	0.9
10	veryLow	low	veryLow	low	nil	0.1
11	veryLow	low	low	veryLow	nil	0.8
12	veryLow	low	low	low	nil	0.2
13	veryLow	low	high	low	nil	0.1
14	veryLow	low	high	high	nil	0.9
15	veryLow	low	veryHigh	veryHigh	nil	0.9
16	veryLow	low	veryHigh	high	nil	0.1
17	veryLow	high	veryLow	low	nil	0.5
18	veryLow	high	veryLow	high	nil	0.5
19	veryLow	high	low	low	nil	0.4
20	veryLow	high	low	high	nil	0.6
21	veryLow	high	high	veryHigh	nil	0.2
22	veryLow	high	high	high	nil	0.8
23	veryLow	high	veryHigh	veryHigh	nil	0.7
24	veryLow	high	veryHigh	high	nil	0.3
25	veryLow	veryHigh	veryLow	high	nil	0.8
26	veryLow	veryHigh	veryLow	veryHigh	nil	0.2
27	veryLow	veryHigh	low	high	nil	0.7
28	veryLow	veryHigh	low	veryHigh	nil	0.6
29	veryLow	veryHigh	high	high	nil	0.5
30	veryLow	veryHigh	high	veryHigh	nil	0.5
31	veryLow	veryHigh	veryHigh	veryHigh	nil	0.5
32	veryLow	veryHigh	veryHigh	veryHigh	nil	0.5
33	low	veryLow	veryLow	low	nil	0.1
34	low	veryLow	veryLow	veryLow	nil	0.9
35	low	veryLow	low	low	nil	0.4
36	low	veryLow	low	veryLow	nil	0.6
37	low	veryLow	high	low	nil	0.1
38	low	veryLow	high	high	nil	0.9
39	low	veryLow	veryHigh	veryHigh	nil	0.9
40	low	veryLow	veryHigh	high	nil	0.1
41	low	low	veryLow	veryLow	nil	0.9
42	low	low	veryLow	low	nil	0.1
43	low	low	low	low	nil	0.5
44	low	low	low	low	nil	0.5
45	low	low	high	high	nil	0.9
46	low	low	high	low	nil	0.1
47	low	low	veryHigh	veryHigh	nil	0.9
48	low	low	veryHigh	high	nil	0.1
49	low	high	veryLow	high	nil	0.5
50	low	high	veryLow	low	nil	0.5
51	low	high	low	high	nil	0.6
52	low	high	low	low	nil	0.4
53	low	high	high	high	nil	0.5
54	low	high	high	high	nil	0.5
55	low	high	veryHigh	veryHigh	nil	0.9
56	low	high	veryHigh	high	nil	0.1
57	low	veryHigh	veryLow	high	nil	0.4
58	low	veryHigh	veryLow	veryHigh	nil	0.6
59	low	veryHigh	low	veryHigh	nil	0.7
60	low	veryHigh	low	high	nil	0.3
61	low	veryHigh	high	veryHigh	nil	0.8

Table A.7 (Continued)

62	low	veryHigh	high	high	nil	0.2
63	low	veryHigh	veryHigh	veryHigh	nil	0.5
64	low	veryHigh	veryHigh	veryHigh	nil	0.5
65	high	veryLow	veryLow	low	nil	0.2
66	high	veryLow	veryLow	high	nil	0.8
67	high	veryLow	low	low	nil	0.1
68	high	veryLow	low	high	nil	0.9
69	high	veryLow	high	high	nil	0.5
70	high	veryLow	high	high	nil	0.5
71	high	veryLow	veryHigh	veryHigh	nil	0.5
72	high	veryLow	veryHigh	veryHigh	nil	0.5
73	high	low	veryLow	high	nil	0.7
74	high	low	veryLow	low	nil	0.3
75	high	low	low	high	nil	0.8
76	high	low	low	low	nil	0.3
77	high	low	high	high	nil	0.5
78	high	low	high	high	nil	0.5
79	high	low	veryHigh	veryHigh	nil	0.5
80	high	low	veryHigh	veryHigh	nil	0.5
81	veryHigh	veryLow	veryLow	veryHigh	nil	0.9
82	veryHigh	veryLow	veryLow	high	nil	0.1
83	veryHigh	veryLow	low	veryHigh	nil	0.9
84	veryHigh	veryLow	low	high	nil	0.1
85	veryHigh	veryLow	high	veryHigh	nil	0.5
86	veryHigh	veryLow	high	veryHigh	nil	0.5
87	veryHigh	veryLow	veryHigh	veryHigh	nil	0.5
88	veryHigh	veryLow	veryHigh	veryHigh	nil	0.5
89	veryHigh	low	veryLow	veryHigh	nil	0.6
90	veryHigh	low	veryLow	high	nil	0.4
91	veryHigh	low	low	veryHigh	nil	0.7
92	veryHigh	low	low	high	nil	0.3
93	veryHigh	low	high	veryHigh	nil	0.8
94	veryHigh	low	high	high	nil	0.2
95	veryHigh	low	veryHigh	veryHigh	nil	0.5
96	veryHigh	low	veryHigh	veryHigh	nil	0.5

Block: past error					
Rule#	pae	pvs	pe	gamma	dos
1	negative	veryNegative	negativeLarge	nil	0.5
2	negative	veryNegative	negativeLarge	nil	0.5
3	negative	negative	negativeLarge	nil	0.5
4	negative	negative	negativeLarge	nil	0.5
5	negative	positive	negativeLarge	nil	0.7
6	negative	positive	negative	nil	0.3
7	negative	veryPositive	negativeLarge	nil	0.5
8	negative	veryPositive	negative	nil	0.5
9	negativeSmall	veryNegative	negativeLarge	nil	0.4
10	negativeSmall	veryNegative	negative	nil	0.6
11	negativeSmall	negative	negativeLarge	nil	0.3
12	negativeSmall	negative	negative	nil	0.7
13	negativeSmall	positive	negative	nil	0.2
14	negativeSmall	positive	zero	nil	0.8
15	negativeSmall	veryPositive	zero	nil	0.7
16	negativeSmall	veryPositive	positive	nil	0.3
17	zero	veryNegative	negative	nil	0.8
18	zero	veryNegative	negativeLarge	nil	0.2
19	zero	negative	negative	nil	0.7
20	zero	negative	zero	nil	0.3
21	zero	positive	positive	nil	0.7
22	zero	positive	zero	nil	0.3
23	zero	veryPositive	positive	nil	0.8
24	zero	veryPositive	positiveLarge	nil	0.2
25	positiveSmall	veryNegative	negative	nil	0.3
26	positiveSmall	veryNegative	zero	nil	0.7
27	positiveSmall	negative	zero	nil	0.8
28	positiveSmall	negative	positive	nil	0.2
29	positiveSmall	positive	positive	nil	0.8
30	positiveSmall	positive	positiveLarge	nil	0.2
31	positiveSmall	veryPositive	positive	nil	0.5
32	positiveSmall	veryPositive	positiveLarge	nil	0.5
33	positive	veryNegative	positiveLarge	nil	0.5

Table (A.7 Continued)

34	positive	veryNegative	positive	nil	0.5
35	positive	negative	positiveLarge	nil	0.7
36	positive	negative	positive	nil	0.3
37	positive	positive	negativeLarge	nil	0.5
38	positive	positive	negativeLarge	nil	0.5
39	positive	veryPositive	negativeLarge	nil	0.5
40	positive	veryPositive	negativeLarge	nil	0.5

Block: current error						
Rule#	**cae**	**cvs**	**cvsa**	**ce**	**gamma**	**dos**
1	negative	veryNegative	veryNegative	negativeLarge	nil	0.5
2	negative	veryNegative	veryNegative	negativeLarge	nil	0.5
3	negative	veryNegative	negative	negativeLarge	nil	0.5
4	negative	veryNegative	negative	negativeLarge	nil	0.5
5	negative	veryNegative	positive	negativeLarge	nil	0.9
6	negative	veryNegative	positive	negative	nil	0.1
7	negative	veryNegative	veryPositive	negativeLarge	nil	0.8
8	negative	veryNegative	veryPositive	negative	nil	0.2
9	negative	negative	veryNegative	negativeLarge	nil	0.5
10	negative	negative	veryNegative	negativeLarge	nil	0.5
11	negative	negative	negative	negativeLarge	nil	0.5
12	negative	negative	negative	negativeLarge	nil	0.5
13	negative	negative	positive	negativeLarge	nil	0.8
14	negative	negative	positive	negative	nil	0.2
15	negative	negative	veryPositive	negativeLarge	nil	0.7
16	negative	negative	veryPositive	negative	nil	0.3
17	negative	positive	veryNegative	negative	nil	0.4
18	negative	positive	veryNegative	negativeLarge	nil	0.6
19	negative	positive	negative	negative	nil	0.4
20	negative	positive	negative	negativeLarge	nil	0.6
21	negative	positive	positive	negative	nil	0.5
22	negative	positive	positive	negativeLarge	nil	0.5
23	negative	positive	veryPositive	negative	nil	0.5
24	negative	positive	veryPositive	negativeLarge	nil	0.5
25	negative	veryPositive	veryNegative	negativeLarge	nil	0.5
26	negative	veryPositive	veryNegative	negative	nil	0.5
27	negative	veryPositive	negative	negativeLarge	nil	0.4
28	negative	veryPositive	negative	negative	nil	0.6
29	negative	veryPositive	positive	negativeLarge	nil	0.4
30	negative	veryPositive	positive	negative	nil	0.6
31	negative	veryPositive	veryPositive	negativeLarge	nil	0.4
32	negative	veryPositive	veryPositive	negative	nil	0.6
33	negativeSmall	veryNegative	veryNegative	negative	nil	0.5
34	negativeSmall	veryNegative	veryNegative	negativeLarge	nil	0.5
35	negativeSmall	veryNegative	negative	negative	nil	0.6
36	negativeSmall	veryNegative	negative	negativeLarge	nil	0.4
37	negativeSmall	veryNegative	positive	negative	nil	0.6
38	negativeSmall	veryNegative	positive	negativeLarge	nil	0.4
39	negativeSmall	veryNegative	veryPositive	negative	nil	0.6
40	negativeSmall	veryNegative	veryPositive	negativeLarge	nil	0.4
41	negativeSmall	negative	veryNegative	negative	nil	0.6
42	negativeSmall	negative	veryNegative	negativeLarge	nil	0.4
43	negativeSmall	negative	negative	negative	nil	0.6
44	negativeSmall	negative	negative	negativeLarge	nil	0.4
45	negativeSmall	negative	positive	negative	nil	0.7
46	negativeSmall	negative	positive	negativeLarge	nil	0.3
47	negativeSmall	negative	veryPositive	negative	nil	0.7
48	negativeSmall	negative	veryPositive	negativeLarge	nil	0.3
49	negativeSmall	positive	veryNegative	negative	nil	0.4
50	negativeSmall	positive	veryNegative	zero	nil	0.6
51	negativeSmall	positive	negative	negative	nil	0.2
52	negativeSmall	positive	negative	zero	nil	0.8
53	negativeSmall	positive	positive	positive	nil	0.1
54	negativeSmall	positive	positive	zero	nil	0.9
55	negativeSmall	positive	veryPositive	positive	nil	0.2
56	negativeSmall	positive	veryPositive	zero	nil	0.8
57	negativeSmall	veryPositive	veryNegative	zero	nil	0.5
58	negativeSmall	veryPositive	veryNegative	zero	nil	0.5
59	negativeSmall	veryPositive	negative	zero	nil	0.9
60	negativeSmall	veryPositive	negative	positive	nil	0.1
61	negativeSmall	veryPositive	positive	zero	nil	0.7

Table A.7 (Continued)

62	negativeSmall	veryPositive	positive	positive	nil	0.3
63	negativeSmall	veryPositive	veryPositive	zero	nil	0.6
64	negativeSmall	veryPositive	veryPositive	positive	nil	0.4
65	zero	veryNegative	veryNegative	zero	nil	0.1
66	zero	veryNegative	veryNegative	negative	nil	0.9
67	zero	veryNegative	negative	zero	nil	0.2
68	zero	veryNegative	negative	negative	nil	0.8
69	zero	veryNegative	positive	zero	nil	0.3
70	zero	veryNegative	positive	negative	nil	0.7
71	zero	veryNegative	veryPositive	zero	nil	0.5
72	zero	veryNegative	veryPositive	negative	nil	0.5
73	zero	negative	veryNegative	zero	nil	0.3
74	zero	negative	veryNegative	negative	nil	0.7
75	zero	negative	negative	zero	nil	0.4
76	zero	negative	negative	negative	nil	0.6
77	zero	negative	positive	zero	nil	0.6
78	zero	negative	positive	negative	nil	0.4
79	zero	negative	veryPositive	zero	nil	0.7
80	zero	negative	veryPositive	negative	nil	0.3
81	zero	positive	veryNegative	zero	nil	0.8
82	zero	positive	veryNegative	positive	nil	0.2
83	zero	positive	negative	zero	nil	0.7
84	zero	positive	negative	positive	nil	0.3
85	zero	positive	positive	zero	nil	0.5
86	zero	positive	positive	positive	nil	0.5
87	zero	positive	veryPositive	zero	nil	0.3
88	zero	positive	veryPositive	positive	nil	0.7
89	zero	veryPositive	veryNegative	zero	nil	0.5
90	zero	veryPositive	veryNegative	positive	nil	0.5
91	zero	veryPositive	negative	zero	nil	0.4
92	zero	veryPositive	negative	positive	nil	0.6
93	zero	veryPositive	positive	zero	nil	0.3
94	zero	veryPositive	positive	positive	nil	0.7
95	zero	veryPositive	veryPositive	zero	nil	0.1
96	zero	veryPositive	veryPositive	positive	nil	0.9
97	positiveSmall	veryNegative	veryNegative	zero	nil	0.6
98	positiveSmall	veryNegative	veryNegative	negative	nil	0.4
99	positiveSmall	veryNegative	negative	zero	nil	0.7
100	positiveSmall	veryNegative	negative	negative	nil	0.3
101	positiveSmall	veryNegative	positive	zero	nil	0.9
102	positiveSmall	veryNegative	positive	negative	nil	0.1
103	positiveSmall	veryNegative	veryPositive	zero	nil	0.5
104	positiveSmall	veryNegative	veryPositive	zero	nil	0.5
105	positiveSmall	negative	veryNegative	zero	nil	0.8
106	positiveSmall	negative	veryNegative	negative	nil	0.2
107	positiveSmall	negative	negative	zero	nil	0.9
108	positiveSmall	negative	negative	negative	nil	0.1
109	positiveSmall	negative	positive	zero	nil	0.8
110	positiveSmall	negative	positive	positive	nil	0.2
111	positiveSmall	negative	veryPositive	zero	nil	0.7
112	positiveSmall	negative	veryPositive	positive	nil	0.3
113	positiveSmall	positive	veryNegative	positive	nil	0.7
114	positiveSmall	positive	veryNegative	positiveLarge	nil	0.3
115	positiveSmall	positive	negative	positive	nil	0.7
116	positiveSmall	positive	negative	positiveLarge	nil	0.3
117	positiveSmall	positive	positive	positive	nil	0.6
118	positiveSmall	positive	positive	positiveLarge	nil	0.4
119	positiveSmall	positive	veryPositive	positive	nil	0.6
120	positiveSmall	positive	veryPositive	positiveLarge	nil	0.4
121	positiveSmall	veryPositive	veryNegative	positive	nil	0.6
122	positiveSmall	veryPositive	veryNegative	positiveLarge	nil	0.4
123	positiveSmall	veryPositive	negative	positive	nil	0.6
124	positiveSmall	veryPositive	negative	positiveLarge	nil	0.4
125	positiveSmall	veryPositive	positive	positive	nil	0.5
126	positiveSmall	veryPositive	positive	positiveLarge	nil	0.5
127	positiveSmall	veryPositive	veryPositive	positive	nil	0.4
128	positiveSmall	veryPositive	veryPositive	positiveLarge	nil	0.6
129	positive	veryNegative	veryNegative	positive	nil	0.6
130	positive	veryNegative	veryNegative	positiveLarge	nil	0.4
131	positive	veryNegative	negative	positive	nil	0.6
132	positive	veryNegative	negative	positiveLarge	nil	0.4

Table A.7 (Continued)

133	positive	veryNegative	positive	positive	nil	0.6
134	positive	veryNegative	positive	positiveLarge	nil	0.4
135	positive	veryNegative	veryPositive	positive	nil	0.5
136	positive	veryNegative	veryPositive	positiveLarge	nil	0.5
137	positive	negative	veryNegative	positive	nil	0.5
138	positive	negative	veryNegative	positiveLarge	nil	0.5
139	positive	negative	negative	positive	nil	0.5
140	positive	negative	negative	positiveLarge	nil	0.5
141	positive	negative	positive	positive	nil	0.4
142	positive	negative	positive	positiveLarge	nil	0.6
143	positive	negative	veryPositive	positive	nil	0.4
144	positive	negative	veryPositive	positiveLarge	nil	0.6
145	positive	positive	veryNegative	positive	nil	0.3
146	positive	positive	veryNegative	positiveLarge	nil	0.7
147	positive	positive	negative	positive	nil	0.2
148	positive	positive	negative	positiveLarge	nil	0.8
149	positive	positive	positive	positiveLarge	nil	0.5
150	positive	positive	positive	positiveLarge	nil	0.5
151	positive	positive	veryPositive	positiveLarge	nil	0.5
152	positive	positive	veryPositive	positiveLarge	nil	0.5
153	positive	veryPositive	veryNegative	positive	nil	0.2
154	positive	veryPositive	veryNegative	positiveLarge	nil	0.8
155	positive	veryPositive	negative	positive	nil	0.1
156	positive	veryPositive	negative	positiveLarge	nil	0.9
157	positive	veryPositive	positive	positiveLarge	nil	0.5
158	positive	veryPositive	positive	positiveLarge	nil	0.5
159	positive	veryPositive	veryPositive	positiveLarge	nil	0.5
160	positive	veryPositive	veryPositive	positiveLarge	nil	0.5

Block: function performance						
Rule#	pe	ce	high	fp	gamma	dos
1	negativeLarge	negativeLarge	veryLow	low	nil	0.6
2	negativeLarge	negativeLarge	veryLow	high	nil	0.4
3	negativeLarge	negativeLarge	low	low	nil	0.7
4	negativeLarge	negativeLarge	low	high	nil	0.3
5	negativeLarge	negativeLarge	high	low	nil	0.8
6	negativeLarge	negativeLarge	high	veryLow	nil	0.2
7	negativeLarge	negativeLarge	veryHigh	low	nil	0.1
8	negativeLarge	negativeLarge	veryHigh	veryLow	nil	0.9
9	negativeLarge	negative	veryLow	veryHigh	nil	0.8
10	negativeLarge	negative	veryLow	high	nil	0.2
11	negativeLarge	negative	low	high	nil	0.8
12	negativeLarge	negative	low	veryHigh	nil	0.2
13	negativeLarge	negative	high	low	nil	0.8
14	negativeLarge	negative	high	veryLow	nil	0.2
15	negativeLarge	negative	veryHigh	low	nil	0.2
16	negativeLarge	negative	veryHigh	veryLow	nil	0.8
17	negativeLarge	zero	veryLow	veryHigh	nil	0.9
18	negativeLarge	zero	veryLow	high	nil	0.1
19	negativeLarge	zero	low	veryHigh	nil	0.3
20	negativeLarge	zero	low	high	nil	0.7
21	negativeLarge	zero	high	low	nil	0.5
22	negativeLarge	zero	high	veryLow	nil	0.5
23	negativeLarge	zero	veryHigh	veryLow	nil	0.8
24	negativeLarge	zero	veryHigh	low	nil	0.2
25	negativeLarge	positive	veryLow	high	nil	0.5
26	negativeLarge	positive	veryLow	veryHigh	nil	0.5
27	negativeLarge	positive	low	high	nil	0.9
28	negativeLarge	positive	low	veryHigh	nil	0.1
29	negativeLarge	positive	high	low	nil	0.5
30	negativeLarge	positive	high	veryLow	nil	0.5
31	negativeLarge	positive	veryHigh	veryLow	nil	0.9
32	negativeLarge	positive	veryHigh	low	nil	0.1
33	negativeLarge	positiveLarge	veryLow	veryLow	nil	0.5
34	negativeLarge	positiveLarge	veryLow	low	nil	0.5
35	negativeLarge	positiveLarge	low	veryLow	nil	0.6
36	negativeLarge	positiveLarge	low	low	nil	0.4
37	negativeLarge	positiveLarge	high	veryLow	nil	0.7
38	negativeLarge	positiveLarge	high	low	nil	0.3
39	negativeLarge	positiveLarge	veryHigh	veryLow	nil	0.5
40	negativeLarge	positiveLarge	veryHigh	veryLow	nil	0.5

Table A.7 (Continued)

41	negative	negativeLarge	veryLow	veryLow	nil	0.5
42	negative	negativeLarge	veryLow	low	nil	0.5
43	negative	negativeLarge	low	veryLow	nil	0.6
44	negative	negativeLarge	low	low	nil	0.4
45	negative	negativeLarge	high	veryLow	nil	0.7
46	negative	negativeLarge	high	low	nil	0.3
47	negative	negativeLarge	veryHigh	veryLow	nil	0.5
48	negative	negativeLarge	veryHigh	veryLow	nil	0.5
49	negative	negative	veryLow	low	nil	0.5
50	negative	negative	veryLow	high	nil	0.5
51	negative	negative	low	low	nil	0.6
52	negative	negative	low	high	nil	0.4
53	negative	negative	high	low	nil	0.5
54	negative	negative	high	veryLow	nil	0.5
55	negative	negative	veryHigh	veryLow	nil	0.8
56	negative	negative	veryHigh	low	nil	0.2
57	negative	zero	veryLow	veryHigh	nil	0.9
58	negative	zero	veryLow	high	nil	0.1
59	negative	zero	low	veryHigh	nil	0.6
60	negative	zero	low	high	nil	0.4
61	negative	zero	high	low	nil	0.5
62	negative	zero	high	veryLow	nil	0.5
63	negative	zero	veryHigh	veryLow	nil	0.7
64	negative	zero	veryHigh	low	nil	0.3
65	negative	positive	veryLow	high	nil	0.5
66	negative	positive	veryLow	low	nil	0.5
67	negative	positive	low	high	nil	0.2
68	negative	positive	low	low	nil	0.8
69	negative	positive	high	low	nil	0.4
70	negative	positive	high	veryLow	nil	0.6
71	negative	positive	veryHigh	veryLow	nil	0.9
72	negative	positive	veryHigh	low	nil	0.1
73	negative	positiveLarge	veryLow	low	nil	0.6
74	negative	positiveLarge	veryLow	veryLow	nil	0.4
75	negative	positiveLarge	low	low	nil	0.4
76	negative	positiveLarge	low	veryLow	nil	0.6
77	negative	positiveLarge	high	veryLow	nil	0.7
78	negative	positiveLarge	high	low	nil	0.3
79	negative	positiveLarge	veryHigh	veryLow	nil	0.9
80	negative	positiveLarge	veryHigh	low	nil	0.1
81	zero	negativeLarge	veryLow	veryLow	nil	0.6
82	zero	negativeLarge	veryLow	low	nil	0.4
83	zero	negativeLarge	low	veryLow	nil	0.7
84	zero	negativeLarge	low	low	nil	0.3
85	zero	negativeLarge	high	veryLow	nil	0.9
86	zero	negativeLarge	high	low	nil	0.1
87	zero	negativeLarge	veryHigh	veryLow	nil	0.5
88	zero	negativeLarge	veryHigh	veryLow	nil	0.5
89	zero	negative	veryLow	veryLow	nil	0.5
90	zero	negative	veryLow	low	nil	0.5
91	zero	negative	low	veryLow	nil	0.6
92	zero	negative	low	low	nil	0.4
93	zero	negative	high	veryLow	nil	0.7
94	zero	negative	high	low	nil	0.3
95	zero	negative	veryHigh	veryLow	nil	0.9
96	zero	negative	veryHigh	low	nil	0.1
97	zero	zero	veryLow	veryHigh	nil	0.7
98	zero	zero	veryLow	high	nil	0.3
99	zero	zero	low	veryHigh	nil	0.3
100	zero	zero	low	high	nil	0.7
101	zero	zero	high	low	nil	0.6
102	zero	zero	high	veryLow	nil	0.4
103	zero	zero	veryHigh	veryLow	nil	0.7
104	zero	zero	veryHigh	low	nil	0.3
105	zero	positive	veryLow	low	nil	0.6
106	zero	positive	veryLow	high	nil	0.4
107	zero	positive	low	veryLow	nil	0.6
108	zero	positive	low	low	nil	0.4
109	zero	positive	high	veryLow	nil	0.7
110	zero	positive	high	low	nil	0.3
111	zero	positive	veryHigh	veryLow	nil	0.5

Table A.7 (Continued)

112	zero	positive	veryHigh	veryLow	nil	0.5
113	zero	positiveLarge	veryLow	low	nil	0.7
114	zero	positiveLarge	veryLow	high	nil	0.3
115	zero	positiveLarge	low	low	nil	0.9
116	zero	positiveLarge	low	veryLow	nil	0.1
117	zero	positiveLarge	high	low	nil	0.5
118	zero	positiveLarge	high	veryLow	nil	0.5
119	zero	positiveLarge	veryHigh	veryLow	nil	0.9
120	zero	positiveLarge	veryHigh	low	nil	0.1
121	positive	negativeLarge	veryLow	veryLow	nil	0.2
122	positive	negativeLarge	veryLow	low	nil	0.8
123	positive	negativeLarge	low	veryLow	nil	0.8
124	positive	negativeLarge	low	low	nil	0.2
125	positive	negativeLarge	high	veryLow	nil	0.9
126	positive	negativeLarge	high	low	nil	0.1
127	positive	negativeLarge	veryHigh	veryLow	nil	0.5
128	positive	negativeLarge	veryHigh	veryLow	nil	0.5
129	positive	negative	veryLow	low	nil	0.6
130	positive	negative	veryLow	veryLow	nil	0.4
131	positive	negative	low	low	nil	0.4
132	positive	negative	low	veryLow	nil	0.6
133	positive	negative	high	veryLow	nil	0.8
134	positive	negative	high	low	nil	0.2
135	positive	negative	veryHigh	veryLow	nil	0.5
136	positive	negative	veryHigh	veryLow	nil	0.5
137	positive	zero	veryLow	veryHigh	nil	0.9
138	positive	zero	veryLow	high	nil	0.1
139	positive	zero	low	veryHigh	nil	0.5
140	positive	zero	low	high	nil	0.5
141	positive	zero	high	low	nil	0.9
142	positive	zero	high	veryLow	nil	0.1
143	positive	zero	veryHigh	veryLow	nil	0.7
144	positive	zero	veryHigh	low	nil	0.3
145	positive	positive	veryLow	low	nil	0.3
146	positive	positive	veryLow	high	nil	0.7
147	positive	positive	low	low	nil	0.9
148	positive	positive	low	veryLow	nil	0.1
149	positive	positive	high	veryLow	nil	0.8
150	positive	positive	high	low	nil	0.2
151	positive	positive	veryHigh	veryLow	nil	0.5
152	positive	positive	veryHigh	veryLow	nil	0.5
153	positive	positiveLarge	veryLow	veryLow	nil	0.5
154	positive	positiveLarge	veryLow	low	nil	0.5
155	positive	positiveLarge	low	veryLow	nil	0.6
156	positive	positiveLarge	low	low	nil	0.4
157	positive	positiveLarge	high	veryLow	nil	0.7
158	positive	positiveLarge	high	low	nil	0.3
159	positive	positiveLarge	veryHigh	veryLow	nil	0.5
160	positive	positiveLarge	veryHigh	veryLow	nil	0.5
161	positiveLarge	negativeLarge	veryLow	low	nil	0.5
162	positiveLarge	negativeLarge	veryLow	low	nil	0.5
163	positiveLarge	negativeLarge	low	low	nil	0.4
164	positiveLarge	negativeLarge	low	veryLow	nil	0.6
165	positiveLarge	negativeLarge	high	low	nil	0.3
166	positiveLarge	negativeLarge	high	veryLow	nil	0.7
167	positiveLarge	negativeLarge	veryHigh	veryLow	nil	0.9
168	positiveLarge	negativeLarge	veryHigh	low	nil	0.1
169	positiveLarge	negative	veryLow	low	nil	0.4
170	positiveLarge	negative	veryLow	high	nil	0.6
171	positiveLarge	negative	low	low	nil	0.8
172	positiveLarge	negative	low	high	nil	0.2
173	positiveLarge	negative	high	low	nil	0.4
174	positiveLarge	negative	high	veryLow	nil	0.6
175	positiveLarge	negative	veryHigh	veryLow	nil	0.8
176	positiveLarge	negative	veryHigh	low	nil	0.2
177	positiveLarge	zero	veryLow	veryHigh	nil	0.9
178	positiveLarge	zero	veryLow	high	nil	0.1
179	positiveLarge	zero	low	veryHigh	nil	0.5
180	positiveLarge	zero	low	high	nil	0.5
181	positiveLarge	zero	high	low	nil	0.5
182	positiveLarge	zero	high	low	nil	0.5

Table A.7 (Continued)

183	positiveLarge	zero	veryHigh	veryLow	nil	0.8
184	positiveLarge	zero	veryHigh	low	nil	0.2
185	positiveLarge	positive	veryLow	veryHigh	nil	0.9
186	positiveLarge	positive	veryLow	high	nil	0.1
187	positiveLarge	positive	low	veryHigh	nil	0.4
188	positiveLarge	positive	low	high	nil	0.6
189	positiveLarge	positive	high	low	nil	0.9
190	positiveLarge	positive	high	high	nil	0.1
191	positiveLarge	positive	veryHigh	veryLow	nil	0.9
192	positiveLarge	positive	veryHigh	low	nil	0.1
193	positiveLarge	positiveLarge	veryLow	high	nil	0.5
194	positiveLarge	positiveLarge	veryLow	low	nil	0.5
195	positiveLarge	positiveLarge	low	low	nil	0.8
196	positiveLarge	positiveLarge	low	high	nil	0.2
197	positiveLarge	positiveLarge	high	veryLow	nil	0.6
198	positiveLarge	positiveLarge	high	low	nil	0.4
199	positiveLarge	positiveLarge	veryHigh	veryLow	nil	0.5
200	positiveLarge	positiveLarge	veryHigh	veryLow	nil	0.5

Table A.8 Trained Selected FOSH 1

name:	**current airspeed (ca)**			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 262.581	right: 292.713	
slow:	type: #lamda	typical: 292.713	left: 262.581	right: 324.164
medium:	type: #lamda	typical: 324.164	left: 292.713	right: 340
fast:	type: #s	typical: 340	left: 324.164	

name:	**current airspeed acceleration (caa)**			
type:	input			
terms:	negative, medium, positive			
negative:	type: #z	typical: -1.16813	right: 0.68915	
medium:	type: #lamda	typical: 0.68915	left: -1.16813	right: 4.91202
positive:	type: #s	typical: 4.91202	left: 0.68915	

name:	**overspeed hazard (oh)**			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.136852	right: 0.782014	
low:	type: #lamda	typical: 0.782014	left: 0.136852	right: 6.19746
high:	type: #lamda	typical: 6.19746	left: 0.782014	right: 9.58944
veryHigh:	type: #s	typical: 9.58944	left: 6.19746	

Block: overspeed hazard					
Rule#	**ca**	**caa**	**oh**	**gamma**	**dos**
1	verySlow	negative	nil	nil	0.535433
2	verySlow	nil	veryLow	nil	0.566929
3	verySlow	medium	veryLow	0	0.795276
4	nil	positive	high	nil	0.535433
5	verySlow	medium	veryLow	0.0787402	0.574803
6	verySlow	positive	nil	0.125984	0.574803
7	medium	negative	nil	nil	0.503937
8	verySlow	negative	nil	nil	0.692913
9	nil	medium	nil	nil	0.637795
10	slow	positive	nil	nil	0.787402
11	nil	positive	veryLow	0.251969	0
12	medium	medium	veryHigh	0	0.212598
13	medium	nil	veryLow	nil	0.393701
14	verySlow	nil	nil	0.283465	0.354331
15	nil	negative	veryLow	nil	0.188976
16	medium	positive	nil	nil	0.677165
17	medium	negative	low	nil	0.582677
18	medium	negative	nil	nil	0.629921
19	fast	nil	nil	nil	0.401575
20	fast	positive	high	nil	0.212598
21	slow	negative	veryLow	nil	0.503937
22	fast	nil	veryHigh	0.535433	0.732283

Table A.8 (Continued)

23	fast	positive	nil	nil	0.267717
24	fast	nil	nil	0	0.102362

Table A.9 Trained Selected FSHS 1

name:	current flight path angle (cfpa)			
type:	input			
terms:	negative, negativeSmall, positiveSmall, positive			
negative:	type: #z	typical: -1.9912	right: 1.38123	
negativeSmall:	type: #lamda	typical: 1.38123	left: -1.9912	right: 13.5806
positiveSmall:	type: #lamda	typical: 13.5806	left: 1.38123	right: 19.8856
positive:	type: #s	typical: 19.8856	left: 13.5806	

name:	current airspeed (ca)			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 120.059	right: 146.232	
slow:	type: #lamda	typical: 146.232	left: 120.059	right: 180.103
medium:	type: #lamda	typical: 180.103	left: 146.232	right: 336.481
fast:	type: #s	typical: 336.481	left: 180.103	

name:	stall hazard (sh)			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #lamda	typical: 0.107527	left: 0.0700028	right: 0.762463
low:	type: #lamda	typical: 0.762463	left: 0.107527	right: 4.69208
high:	type: #lamda	typical: 4.69208	left: 0.762463	right: 9.90225
veryHigh:	type: #s	typical: 9.90225	left: 4.69208	

Block: stall hazard					
Rule#	cfpa	ca	sh	gamma	dos
1	negative	verySlow	low	0.133858	0.68504
2	negative	verySlow	nil	0.283465	0.48819
3	negative	slow	high	nil	0.74803
4	negative	slow	low	nil	0.9685
5	negative	medium	low	nil	0.82677
6	negative	medium	veryLow	nil	1
7	negative	fast	nil	nil	1
8	negative	fast	veryLow	nil	1
9	negativeSmall	verySlow	nil	nil	0.98425
10	negativeSmall	verySlow	nil	nil	1
11	negativeSmall	slow	low	nil	1
12	negativeSmall	slow	high	nil	1
13	negativeSmall	medium	low	nil	0.49606
14	negativeSmall	medium	low	nil	0.93701
15	negativeSmall	fast	nil	nil	0.93701
16	negativeSmall	fast	veryLow	nil	1
17	positiveSmall	verySlow	veryHigh	nil	0.98425
18	positiveSmall	verySlow	nil	nil	1
19	positiveSmall	slow	high	nil	0.99213
20	positiveSmall	slow	high	nil	1
21	positiveSmall	medium	low	nil	0.88976
22	positiveSmall	medium	low	nil	0.88976
23	positiveSmall	fast	nil	0.606299	0.99213
24	positiveSmall	fast	nil	nil	0.28347
25	positive	verySlow	veryHigh	nil	0.74016
26	positive	verySlow	nil	nil	0.68504
27	positive	slow	nil	nil	0.99213
28	positive	slow	nil	0.031496	0.11811
29	positive	medium	low	0.031496	0.2126
30	positive	medium	high	nil	0.47244
31	positive	fast	low	0.007874	0.92913
32	positive	fast	veryLow	0.503937	0.68504

Table A.10 Trained Selected FTHS 2

name:	current flight path angle (cfpa)			
type:	input			
terms:	negative, negativeSmall, positiveSamll, positive			
negative:	type: #z	typical: -8.00293	right: 2.17302	
negativeSmall:	type: #lamda	typical: 2.17302	left: -8.00293	right: 12.9648
positiveSmall:	type: #lamda	typical: 12.9648	left: 2.17302	right: 12.9648
positive:	type: #s	typical: 12.9648	left: 12.9648	

name:	current airspeed (ca)			
type:	input			
terms:	verySlow, slow, medium, fast			
verySlow:	type: #z	typical: 127.097	right: 148.211	
slow:	type: #lamda	typical: 148.211	left: 127.097	right: 175.044
medium:	type: #lamda	typical: 175.044	left: 148.211	right: 180.323
fast:	type: #s	typical: 180.323	left: 175.044	

name:	current thrust (ct)			
type:	input			
terms:	low, medium, high			
low:	type: #z	typical: 0.504399	right: 0.565005	
medium:	type: #lamda	typical: 0.565005	left: 0.504399	right: 0.73998
high:	type: #s	typical: 0.73998	left: 0.565005	

name:	attitude (at)
type:	intermediate
terms:	diving, climbing, descending, climbingExtreme

name:	current altitude (calt)			
type:	input			
terms:	extremelyLow, veryLow, low, medium			
extremelyLow:	type: #z	typical: 732.16	right: 1067.45	
veryLow:	type: #lamda	typical: 1067.45	left: 732.16	right: 5474.1
low:	type: #lamda	typical: 5474.1	left: 1067.45	right: 5905.18
medium:	type: #s	typical: 5905.18	left: 5474.1	

name:	terrain hazard (th)			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #lamda	typical: 0.0782014	left: 0.0418909	right: 0.332356
low:	type: #lamda	typical: 0.332356	left: 0.0782014	right: 2.36559
high:	type: #lamda	typical: 2.36559	left: 0.332356	right: 9.26686
veryHigh:	type: #s	typical: 9.26686	left: 2.36559	

name:	current altitude error (cae)				
type:	input				
terms:	negative, negativeSmall, positiveSmall, positive				
negative:	type: #z	typical: -8739.0	right: -4985.34		
negativeSmall:	type: #pi	typical: -4985.34	typicalEnd: 467.354	left: -8739.0	right: 488.758
positiveSmall:	type: #lamda	typical: 488.758	left: -4985.34	right: 723.361	
positive:	type: #s	typical: 723.361	left: 488.758		

Block: attitude						
Rule#	cfpa	ca	ct	at	gamma	dos
1	negative	verySlow	low	descending	nil	0.377953
2	negative	verySlow	low	climbing	nil	0.535433
3	negative	verySlow	medium	descending	0.409449	0.535433
4	negative	verySlow	medium	climbing	nil	0.574803
5	negative	verySlow	high	climbing	nil	0.787402
6	negative	verySlow	high	descending	0.811024	0.527559
7	negative	slow	low	descending	0.125984	0.188976
8	negative	slow	low	diving	nil	0.503937
9	negative	slow	medium	climbing	nil	0.220472
10	negative	slow	medium	climbingExtreme	0.299213	0.566929
11	negative	slow	high	nil	nil	0.527559
12	negative	slow	high	nil	nil	0.503937
13	negative	medium	low	nil	nil	0.267717
14	negative	medium	low	descending	nil	0.0629921
15	negative	medium	medium	nil	nil	0.511811
16	negative	medium	medium	climbingExtreme	nil	0.755905
17	negative	medium	high	descending	nil	0.795276
18	negative	medium	high	diving	nil	0.606299
19	negative	fast	low	diving	0.0393701	0.149606

Table A.10 (Continued)

20	negative	fast	low	diving	0	0.503937
21	negative	fast	medium	nil	nil	0.0393701
22	negative	fast	medium	nil	nil	0.88189
23	negative	fast	high	climbing	0.0708661	0.622047
24	negative	fast	high	descending	nil	0.188976
25	negativeSmall	verySlow	low	descending	nil	0.944882
26	negativeSmall	verySlow	low	descending	nil	0.503937
27	negativeSmall	verySlow	medium	descending	nil	0.535433
28	negativeSmall	verySlow	medium	descending	nil	0.629921
29	negativeSmall	verySlow	high	descending	nil	0.755905
30	negativeSmall	verySlow	high	nil	nil	0
31	negativeSmall	slow	low	nil	nil	0.535433
32	negativeSmall	slow	low	climbing	0.125984	0.511811
33	negativeSmall	slow	medium	nil	nil	0.519685
34	negativeSmall	slow	medium	descending	nil	0.503937
35	negativeSmall	slow	high	descending	nil	0.566929
36	negativeSmall	slow	high	nil	nil	0.755905
37	negativeSmall	medium	low	nil	nil	0.566929
38	negativeSmall	medium	low	descending	nil	0.503937
39	negativeSmall	medium	medium	descending	nil	0.503937
40	negativeSmall	medium	medium	descending	nil	0.503937
41	negativeSmall	medium	high	descending	0	0
42	negativeSmall	medium	high	nil	0.535433	0.535433
43	negativeSmall	fast	low	descending	nil	0.503937
44	negativeSmall	fast	low	descending	nil	0.535433
45	negativeSmall	fast	medium	climbing	nil	0.511811
46	negativeSmall	fast	medium	nil	nil	0.00787402
47	negativeSmall	fast	high	climbing	nil	0.755905
48	negativeSmall	fast	high	descending	nil	0
49	positiveSmall	verySlow	low	descending	0.629921	0.740157
50	positiveSmall	verySlow	low	nil	nil	0.637795
51	positiveSmall	verySlow	medium	climbing	nil	0.929134
52	positiveSmall	verySlow	medium	nil	nil	0.188976
53	positiveSmall	verySlow	high	diving	nil	0
54	positiveSmall	verySlow	high	descending	nil	0
55	positiveSmall	slow	low	nil	nil	0.629921
56	positiveSmall	slow	low	nil	0.0393701	0.535433
57	positiveSmall	slow	medium	nil	0.125984	0.535433
58	positiveSmall	slow	medium	climbing	nil	0.527559
59	positiveSmall	slow	high	climbing	nil	0.0314961
60	positiveSmall	slow	high	climbing	nil	0.00787402
61	positiveSmall	medium	low	descending	nil	0.503937
62	positiveSmall	medium	low	climbing	nil	0.519685
63	positiveSmall	medium	medium	descending	nil	0.251969
64	positiveSmall	medium	medium	climbing	0.551181	0.503937
65	positiveSmall	medium	high	climbing	nil	0.0787402
66	positiveSmall	medium	high	climbing	nil	0.755905
67	positiveSmall	fast	low	descending	nil	0.519685
68	positiveSmall	fast	low	climbing	nil	0.88189
69	positiveSmall	fast	medium	descending	nil	0.511811
70	positiveSmall	fast	medium	climbing	nil	0.519685
71	positiveSmall	fast	high	climbing	nil	0.566929
72	positiveSmall	fast	high	climbing	0.598425	0.645669
73	positive	verySlow	low	climbingExtreme	nil	0.133858
74	positive	verySlow	low	nil	0.251969	0.511811
75	positive	verySlow	medium	nil	nil	0.818898
76	positive	verySlow	medium	climbing	0.0787402	0.503937
77	positive	verySlow	high	diving	nil	0.0393701
78	positive	verySlow	high	descending	nil	0.779527
79	positive	slow	low	nil	nil	0.968504
80	positive	slow	low	nil	nil	0.354331
81	positive	slow	medium	climbing	nil	0.566929
82	positive	slow	medium	diving	nil	0.692913
83	positive	slow	high	climbing	nil	0.503937
84	positive	slow	high	diving	nil	0.015748
85	positive	medium	low	climbing	nil	0.527559
86	positive	medium	low	climbing	nil	0.598425
87	positive	medium	medium	climbing	nil	0.125984
88	positive	medium	medium	descending	nil	0.637795
89	positive	medium	high	nil	nil	0.566929
90	positive	medium	high	nil	nil	0.503937

Table A.10 (Continued)

91	positive	fast	low	nil	nil	0.503937
92	positive	fast	low	climbing	nil	0.598425
93	positive	fast	medium	climbing	nil	0.503937
94	positive	fast	medium	climbing	nil	0.125984
95	positive	fast	high	climbing	nil	0.629921
96	positive	fast	high	climbing	nil	0.692913

Block: terrain hazard						
Rule#	at	calt	cae	th	gamma	dos
1	diving	extremelyLow	negative	veryHigh	nil	0.503937
2	diving	extremelyLow	negative	nil	nil	0
3	diving	extremelyLow	negativeSmall	veryHigh	nil	0.889764
4	diving	extremelyLow	negativeSmall	veryHigh	nil	0.661417
5	diving	extremelyLow	positiveSmall	veryHigh	nil	0.503937
6	diving	extremelyLow	positiveSmall	veryHigh	nil	0.503937
7	diving	extremelyLow	positive	veryHigh	nil	0.0629921
8	diving	extremelyLow	positive	veryHigh	nil	0.503937
9	diving	veryLow	negative	veryHigh	nil	0.543307
10	diving	veryLow	negative	veryHigh	nil	0.503937
11	diving	veryLow	negativeSmall	nil	nil	0.503937
12	diving	veryLow	negativeSmall	low	nil	0.503937
13	diving	veryLow	positiveSmall	low	nil	0.519685
14	diving	veryLow	positiveSmall	high	nil	0.503937
15	diving	veryLow	positive	high	nil	0.968504
16	diving	veryLow	positive	nil	nil	0.692913
17	diving	low	negative	low	nil	0.527559
18	diving	low	negative	nil	nil	0.755905
19	diving	low	negativeSmall	high	0.015748	0.661417
20	diving	low	negativeSmall	high	nil	0.0629921
21	diving	low	positiveSmall	low	nil	0.535433
22	diving	low	positiveSmall	low	0.251969	0.551181
23	diving	low	positive	nil	nil	0.818898
24	diving	low	positive	nil	nil	0.251969
25	diving	medium	negative	high	nil	0.503937
26	diving	medium	negative	low	nil	0.566929
27	diving	medium	negativeSmall	nil	nil	0.527559
28	diving	medium	negativeSmall	veryHigh	0	0.0787402
29	diving	medium	positiveSmall	veryLow	0.0393701	0.88189
30	diving	medium	positiveSmall	veryLow	nil	0.645669
31	diving	medium	positive	veryLow	nil	0.133858
32	diving	medium	positive	low	nil	0.763779
33	descending	extremelyLow	negative	nil	nil	0.015748
34	descending	extremelyLow	negative	veryLow	nil	0.637795
35	descending	extremelyLow	negativeSmall	high	nil	0.850394
36	descending	extremelyLow	negativeSmall	nil	nil	0.732283
37	descending	extremelyLow	positiveSmall	nil	nil	0.700787
38	descending	extremelyLow	positiveSmall	low	nil	0.0314961
39	descending	extremelyLow	positive	low	nil	0.519685
40	descending	extremelyLow	positive	low	nil	0.755905
41	descending	veryLow	negative	low	nil	0.80315
42	descending	veryLow	negative	high	0.0787402	0.700787
43	descending	veryLow	negativeSmall	low	nil	0.913386
44	descending	veryLow	negativeSmall	high	nil	0.0708661
45	descending	veryLow	positiveSmall	low	nil	0.629921
46	descending	veryLow	positiveSmall	high	0	0
47	descending	veryLow	positive	low	0	0.645669
48	descending	veryLow	positive	nil	nil	0.503937
49	descending	low	negative	low	0.913386	0.787402
50	descending	low	negative	high	nil	0.637795
51	descending	low	negativeSmall	nil	nil	0.834646
52	descending	low	negativeSmall	veryLow	nil	0.677165
53	descending	low	positiveSmall	nil	nil	0.519685
54	descending	low	positiveSmall	nil	nil	0.598425
55	descending	low	positive	low	nil	0.755905
56	descending	low	positive	nil	nil	0.519685
57	descending	medium	negative	low	nil	0.897638
58	descending	medium	negative	nil	nil	0.133858
59	descending	medium	negativeSmall	low	0.283465	0.228346
60	descending	medium	negativeSmall	low	0.251969	0.692913
61	descending	medium	positiveSmall	nil	0.314961	0.645669
62	descending	medium	positiveSmall	veryLow	nil	0.771653

Table A.10 (Continued)

63	descending	medium	positive	nil	nil	0.645669
64	descending	medium	positive	veryLow	nil	0.251969
65	climbing	extremelyLow	negative	veryLow	nil	0.629921
66	climbing	extremelyLow	negative	nil	0.0787402	0.519685
67	climbing	extremelyLow	negativeSmall	veryLow	0.251969	0.755905
68	climbing	extremelyLow	negativeSmall	veryLow	0	0.511811
69	climbing	extremelyLow	positiveSmall	veryLow	nil	0.677165
70	climbing	extremelyLow	positiveSmall	veryLow	0	0.19685
71	climbing	extremelyLow	positive	veryHigh	nil	0.259843
72	climbing	extremelyLow	positive	nil	0	0.818898
73	climbing	veryLow	negative	nil	nil	0.511811
74	climbing	veryLow	negative	veryLow	nil	0.543307
75	climbing	veryLow	negativeSmall	veryLow	nil	0.519685
76	climbing	veryLow	negativeSmall	veryLow	0.503937	0.511811
77	climbing	veryLow	positiveSmall	veryLow	nil	0.102362
78	climbing	veryLow	positiveSmall	nil	nil	0.251969
79	climbing	veryLow	positive	high	nil	0.503937
80	climbing	veryLow	positive	veryLow	nil	0.787402
81	climbing	low	negative	veryLow	nil	0.551181
82	climbing	low	negative	nil	nil	0
83	climbing	low	negativeSmall	veryLow	nil	0.629921
84	climbing	low	negativeSmall	veryLow	nil	0.88189
85	climbing	low	positiveSmall	nil	nil	0
86	climbing	low	positiveSmall	veryLow	0.314961	0.763779
87	climbing	low	positive	veryHigh	nil	0.566929
88	climbing	low	positive	veryLow	0	0.771653
89	climbing	medium	negative	veryLow	nil	0.141732
90	climbing	medium	negative	nil	nil	0.0314961
91	climbing	medium	negativeSmall	veryLow	0	0.692913
92	climbing	medium	negativeSmall	veryLow	0.0629921	0.377953
93	climbing	medium	positiveSmall	nil	nil	0.503937
94	climbing	medium	positiveSmall	veryLow	nil	0.535433
95	climbing	medium	positive	veryLow	nil	0.598425
96	climbing	medium	positive	veryLow	0.125984	0.0708661
97	climbingExtreme	extremelyLow	negative	high	nil	0.125984
98	climbingExtreme	extremelyLow	negative	veryHigh	nil	0.787402
99	climbingExtreme	extremelyLow	negativeSmall	high	0	0.527559
100	climbingExtreme	extremelyLow	negativeSmall	veryHigh	nil	0.503937
101	climbingExtreme	extremelyLow	positiveSmall	high	nil	0.661417
102	climbingExtreme	extremelyLow	positiveSmall	nil	nil	0.755905
103	climbingExtreme	extremelyLow	positive	high	nil	0.535433
104	climbingExtreme	extremelyLow	positive	veryHigh	nil	0.503937
105	climbingExtreme	veryLow	negative	high	nil	0.582677
106	climbingExtreme	veryLow	negative	high	nil	0.826772
107	climbingExtreme	veryLow	negativeSmall	high	nil	0.503937
108	climbingExtreme	veryLow	negativeSmall	high	nil	0.582677
109	climbingExtreme	veryLow	positiveSmall	low	0.0314961	0.00787402
110	climbingExtreme	veryLow	positiveSmall	nil	0.023622	0.377953
111	climbingExtreme	veryLow	positive	nil	nil	0.0314961
112	climbingExtreme	veryLow	positive	high	nil	0.535433
113	climbingExtreme	low	negative	high	0	0.913386
114	climbingExtreme	low	negative	high	0.503937	0
115	climbingExtreme	low	negativeSmall	high	nil	0.732283
116	climbingExtreme	low	negativeSmall	low	nil	0.811024
117	climbingExtreme	low	positiveSmall	low	nil	0.00787402
118	climbingExtreme	low	positiveSmall	low	0	0
119	climbingExtreme	low	positive	high	nil	0.535433
120	climbingExtreme	low	positive	high	nil	0.519685
121	climbingExtreme	medium	negative	veryLow	nil	0.133858
122	climbingExtreme	medium	negative	nil	nil	0.787402
123	climbingExtreme	medium	negativeSmall	high	nil	0.511811
124	climbingExtreme	medium	negativeSmall	nil	nil	0.0629921
125	climbingExtreme	medium	positiveSmall	high	nil	0.559055
126	climbingExtreme	medium	positiveSmall	nil	0.0629921	0.755905
127	climbingExtreme	medium	positive	high	nil	0.015748
128	climbingExtreme	medium	positive	nil	nil	0

Table A.11 Trained Selected FHEIS

name:	stall hazard (sh)
type:	intermediate
terms:	veryHigh, veryLow, low, high

name:	current error (ce)
type:	intermediate
terms:	negativeLarge, negative, positiveLarge, positive, zero

name:	past error (pe)
type:	intermediate
terms:	negativeLarge, negative, positiveLarge, positive, zero

name:	past vertical speed (pvs)				
type:	input				
terms:	veryNegative, negative, postive, veryPositive				
veryNegative:	type: #z	typical: -5953.08	right: -1307.92		
negative:	type: #lamda	typical: -1307.92	left: -5953.08	right: 2997.07	
positive:	type: #pi	typical: 2997.07	typicalEnd: 4604.76	left: -1307.92	right: 5976.54
veryPositive:	type: #s	typical: 5976.54	left: 2997.07		

name:	current vertical speed (cvs)				
type:	input				
terms:	veryNegative, negative, positive, veryPositive				
veryNegative:	type: #z	typical: -3325.51	right: -557.185		
negative:	type: #pi	typical: -557.185	typicalEnd: -469.122	left: -3325.51	right: -381.232
positive:	type: #lamda	typical: -381.232	left: -557.185	right: 5155.42	
veryPositive:	type: #s	typical: 5155.42	left: -381.232		

name:	past altitude error (pae)				
type:	input				
terms:	negative, negativeSmall, zero, positiveSmall, positive				
negative:	type: #z	typical: -20425.2	right: -10024.4		
negativeSmall:	type: #lamda	typical: -10024.4	left: -20425.2	right: 1197.46	
zero:	type: #lamda	typical: 1197.46	left: -10024.4	right: 26720.4	
positiveSmall:	type: #pi	typical: 26720.4	typicalEnd: 31220.6	left: 1197.46	right: 32194.5
positive:	type: #s	typical: 32194.5	left: 26720.4		

name:	overspeed hazard (oh)
type:	intermediate
terms:	veryHigh, veryLow, low, high

name:	current altitude error (cae)				
type:	input				
terms:	negative, negativeSmall, zero, postiveSmall, positive				
negative:	type: #z	typical: -27541.5	right: -1676.44		
negativeSmall:	type: #pi	typical: -1676.44	typicalEnd: -235.009	left: -27541.5	right: 34.2148
zero:	type: #pi	typical: 34.2148	typicalEnd: 498.148	left: -1676.44	right: 855.328
positiveSmall:	type: #pi	typical: 855.328	typicalEnd: 7094.22	left: 34.2148	right: 21246.3
positive:	type: #s	typical: 21246.3	left: 855.328		

name:	current vertical speed acceleration (cvsa)			
type:	input			
terms:	veryNegative, negative, positive, veryPositive			
veryNegative:	type: #z	typical: -2010.26	right: 101.173	
negative:	type: #lamda	typical: 101.173	left: -2010.26	right: 156.891
positive:	type: #lamda	typical: 156.891	left: 101.173	right: 362.17
veryPositive:	type: #s	typical: 362.17	left: 156.891	

name:	overall hazard (h)
type:	intermediate
terms:	veryHigh, veryLow, low, high

name:	terrain hazard (th)
type:	intermediate
terms:	veryHigh, veryLow, low, high

Table A.11 (Continued)

name:	function performance (fp)			
type:	output			
terms:	veryLow, low, high, veryHigh			
veryLow:	type: #z	typical: 0.234604	right: 0.997067	
low:	type: #lamda	typical: 0.997067	left: 0.234604	right: 2.50244
high:	type: #lamda	typical: 2.50244	left: 0.997067	right: 9.98045
veryHigh:	type: #s	typical: 9.98045	left: 2.50244	

Block: overall hazard						
Rule#	sh	oh	th	h	gamma	dos
1	veryLow	veryLow	veryLow	veryLow	nil	1
2	veryLow	veryLow	veryLow	nil	nil	0.496063
3	veryLow	veryLow	low	veryLow	nil	0.80315
4	veryLow	veryLow	low	veryLow	nil	0.448819
5	veryLow	veryLow	high	low	nil	0.425197
6	veryLow	veryLow	high	nil	nil	0.19685
7	veryLow	veryLow	veryHigh	nil	nil	0.338583
8	veryLow	veryLow	veryHigh	high	0.88189	0.19685
9	veryLow	low	veryLow	veryLow	nil	0.779527
10	veryLow	low	veryLow	low	nil	0.228346
11	veryLow	low	low	veryLow	nil	0.0472441
12	veryLow	low	low	low	0.0629921	0.968504
13	veryLow	low	high	nil	nil	0.102362
14	veryLow	low	high	veryLow	nil	0.377953
15	veryLow	low	veryHigh	veryHigh	nil	0.929134
16	veryLow	low	veryHigh	veryLow	nil	0.102362
17	veryLow	high	veryLow	nil	0	0.125964
18	veryLow	high	veryLow	veryLow	nil	0.566929
19	veryLow	high	low	high	0.0944882	0.488189
20	veryLow	high	low	high	nil	0.850394
21	veryLow	high	high	low	nil	0.102362
22	veryLow	high	high	nil	0.133858	0.866142
23	veryLow	high	veryHigh	low	nil	0.700787
24	veryLow	high	veryHigh	high	nil	0.299213
25	veryLow	veryHigh	veryLow	high	nil	0.299213
26	veryLow	veryHigh	veryLow	veryHigh	nil	0.700787
27	veryLow	veryHigh	low	high	nil	0.133858
28	veryLow	veryHigh	low	nil	nil	0.598425
29	veryLow	veryHigh	high	high	0.0787402	0.503937
30	veryLow	veryHigh	high	nil	nil	0.535433
31	veryLow	veryHigh	veryHigh	nil	nil	0.535433
32	veryLow	veryHigh	veryHigh	veryHigh	nil	0.787402
33	low	veryLow	veryLow	high	nil	0.0629921
34	low	veryLow	veryLow	veryLow	nil	0.992126
35	low	veryLow	low	nil	nil	0.401575
36	low	veryLow	low	veryLow	nil	0.362205
37	low	veryLow	high	low	nil	0.102362
38	low	veryLow	high	high	nil	0.897638
39	low	veryLow	veryHigh	veryHigh	nil	0.889764
40	low	veryLow	veryHigh	high	nil	0.992126
41	low	low	veryLow	veryLow	nil	0.897638
42	low	low	veryLow	nil	nil	0.00787402
43	low	low	low	nil	nil	0.503937
44	low	low	low	low	nil	0.0787402
45	low	low	high	nil	nil	0.96063
46	low	low	high	high	nil	0.622047
47	low	low	veryHigh	nil	0	0.897638
48	low	low	veryHigh	low	0	0.0708661
49	low	high	veryLow	high	nil	0.661417
50	low	high	veryLow	nil	nil	0.645669
51	low	high	low	low	0	0.818898
52	low	high	low	nil	nil	0.401575
53	low	high	high	high	nil	0.88189
54	low	high	high	nil	nil	0.755905
55	low	high	veryHigh	veryHigh	0.755905	0.645669
56	low	high	veryHigh	low	nil	0.00787402
57	low	veryHigh	veryLow	high	nil	0.338583
58	low	veryHigh	veryLow	nil	nil	0.614173
59	low	veryHigh	low	nil	nil	0.708661
60	low	veryHigh	low	high	0.015748	0.267717
61	low	veryHigh	high	veryLow	nil	0.645669

Table A.11 (Continued)

62	low	veryHigh	high	high	nil	0.464567
63	low	veryHigh	veryHigh	veryHigh	0.771653	0.929134
64	low	veryHigh	veryHigh	nil	nil	0.787402
65	high	veryLow	veryLow	high	nil	0.19685
66	high	veryLow	veryLow	low	nil	0.80315
67	high	veryLow	low	high	0	0.228346
68	high	veryLow	low	low	nil	0.897638
69	high	veryLow	high	high	nil	0.503937
70	high	veryLow	high	nil	nil	0.566929
71	high	veryLow	veryHigh	veryHigh	nil	0.503937
72	high	veryLow	veryHigh	nil	nil	0.503937
73	high	low	veryLow	veryLow	nil	0.889764
74	high	low	veryLow	low	0.015748	0.0314961
75	high	low	low	nil	nil	0.755905
76	high	low	low	low	nil	0.0472441
77	high	low	high	high	nil	0.125984
78	high	low	high	high	nil	0.818898
79	high	low	veryHigh	veryHigh	nil	0.645669
80	high	low	veryHigh	veryHigh	0.535433	0.543307
81	veryHigh	veryLow	veryLow	veryHigh	0.0708661	0.708661
82	veryHigh	veryLow	veryLow	veryLow	nil	0.622047
83	veryHigh	veryLow	low	nil	nil	0.393701
84	veryHigh	veryLow	low	high	nil	0.0314961
85	veryHigh	veryLow	high	veryHigh	nil	0.811024
86	veryHigh	veryLow	high	veryHigh	nil	0.763779
87	veryHigh	veryLow	veryHigh	veryHigh	nil	0.125984
88	veryHigh	veryLow	veryHigh	veryHigh	nil	0.566929
89	veryHigh	low	veryLow	nil	nil	0.0629921
90	veryHigh	low	veryLow	high	nil	0.393701
91	veryHigh	low	low	veryHigh	nil	0.700787
92	veryHigh	low	low	high	0	0.299213
93	veryHigh	low	high	nil	nil	0.929134
94	veryHigh	low	high	high	nil	0.19685
95	veryHigh	low	veryHigh	veryHigh	nil	0.503937
96	veryHigh	low	veryHigh	veryHigh	nil	0.503937

Block: past error					
Rule#	**pae**	**pvs**	**pe**	**gamma**	**dos**
1	negative	veryNegative	negativeLarge	0.0944882	0.503937
2	negative	veryNegative	negativeLarge	nil	0.661417
3	negative	negative	nil	nil	0.0944882
4	negative	negative	zero	nil	0.015748
5	negative	positive	negativeLarge	nil	0.700787
6	negative	positive	nil	nil	0.80315
7	negative	veryPositive	negativeLarge	nil	0.677165
8	negative	veryPositive	nil	nil	0.503937
9	negativeSmall	veryNegative	negativeLarge	nil	0.133858
10	negativeSmall	veryNegative	negative	0.125984	0.598425
11	negativeSmall	negative	nil	0.0472441	0.11811
12	negativeSmall	negative	negative	nil	0.653543
13	negativeSmall	positive	negative	0.409449	0.228346
14	negativeSmall	positive	nil	nil	0.80315
15	negativeSmall	veryPositive	negativeLarge	nil	0.692913
16	negativeSmall	veryPositive	zero	nil	0.267717
17	zero	veryNegative	negative	nil	0.299213
18	zero	veryNegative	nil	nil	0.716535
19	zero	negative	negative	0.519685	0.850394
20	zero	negative	negativeLarge	nil	0.614173
21	zero	positive	positive	nil	0.606299
22	zero	positive	negative	nil	0.299213
23	zero	veryPositive	nil	nil	0.606299
24	zero	veryPositive	negativeLarge	nil	0.212598
25	positiveSmall	veryNegative	negative	0.00787402	0.0314961
26	positiveSmall	veryNegative	nil	nil	0.629921
27	positiveSmall	negative	zero	nil	0.535433
28	positiveSmall	negative	positive	nil	0.19685
29	positiveSmall	positive	negativeLarge	nil	0.551181
30	positiveSmall	positive	positiveLarge	nil	0.11811
31	positiveSmall	veryPositive	positiveLarge	nil	0.251969
32	positiveSmall	veryPositive	nil	nil	0.590551
33	positive	veryNegative	positiveLarge	nil	0.125984

Table A.11 (Continued)

34	positive	veryNegative	nil	nil	0.755905
35	positive	negative	positiveLarge	nil	0.88189
36	positive	negative	positiveLarge	nil	0.259843
37	positive	positive	zero	nil	0.755905
38	positive	positive	negativeLarge	nil	0.503937
39	positive	veryPositive	nil	0.503937	0
40	positive	veryPositive	negativeLarge	nil	0

Block: current error						
Rule#	cae	cvs	cvsa	ce	gamma	dos
1	negative	veryNegative	veryNegative	negativeLarge	nil	0.787402
2	negative	veryNegative	veryNegative	nil	nil	0.566929
3	negative	veryNegative	negative	negativeLarge	nil	0.503937
4	negative	veryNegative	negative	nil	nil	0.503937
5	negative	veryNegative	positive	zero	0.0629921	0.937008
6	negative	veryNegative	positive	negative	nil	0.102362
7	negative	veryNegative	veryPositive	negativeLarge	nil	0.80315
8	negative	veryNegative	veryPositive	zero	nil	0.19685
9	negative	negative	veryNegative	nil	0.519685	0.566929
10	negative	negative	veryNegative	zero	nil	0.511811
11	negative	negative	negative	negativeLarge	nil	0.629921
12	negative	negative	negative	positive	nil	0.629921
13	negative	negative	positive	negativeLarge	0	0.661417
14	negative	negative	positive	nil	0.0629921	0.448819
15	negative	negative	veryPositive	negativeLarge	nil	0.574803
16	negative	negative	veryPositive	negative	nil	0.811024
17	negative	positive	veryNegative	zero	nil	0.401575
18	negative	positive	veryNegative	negativeLarge	nil	0.566929
19	negative	positive	negative	negative	nil	0.448819
20	negative	positive	negative	negativeLarge	nil	0.724409
21	negative	positive	positive	nil	nil	0.503937
22	negative	positive	positive	negativeLarge	nil	0.535433
23	negative	positive	veryPositive	zero	nil	0.0708661
24	negative	positive	veryPositive	negativeLarge	nil	0.527559
25	negative	veryPositive	veryNegative	nil	nil	0.755905
26	negative	veryPositive	veryNegative	nil	nil	0.535433
27	negative	veryPositive	negative	negativeLarge	nil	0.889764
28	negative	veryPositive	negative	negative	nil	0.598425
29	negative	veryPositive	positive	negativeLarge	nil	0.464567
30	negative	veryPositive	positive	nil	nil	0.535433
31	negative	veryPositive	veryPositive	negativeLarge	nil	0.905512
32	negative	veryPositive	veryPositive	negative	nil	0.598425
33	negativeSmall	veryNegative	veryNegative	nil	nil	0.622047
34	negativeSmall	veryNegative	veryNegative	negativeLarge	0.80315	0.590551
35	negativeSmall	veryNegative	negative	negative	nil	0.614173
36	negativeSmall	veryNegative	negative	negativeLarge	nil	0.401575
37	negativeSmall	veryNegative	positive	negative	nil	0.598425
38	negativeSmall	veryNegative	positive	negativeLarge	nil	0.401575
39	negativeSmall	veryNegative	veryPositive	negativeLarge	0	0.0944882
40	negativeSmall	veryNegative	veryPositive	negativeLarge	nil	0.259843
41	negativeSmall	negative	veryNegative	negative	nil	0.598425
42	negativeSmall	negative	veryNegative	negativeLarge	nil	0.401575
43	negativeSmall	negative	negative	negative	nil	0.850394
44	negativeSmall	negative	negative	nil	nil	0.393701
45	negativeSmall	negative	positive	negative	nil	0.700787
46	negativeSmall	negative	positive	negativeLarge	nil	0.0472441
47	negativeSmall	negative	veryPositive	negative	nil	0.732283
48	negativeSmall	negative	veryPositive	negative	nil	0.771653
49	negativeSmall	positive	veryNegative	negative	nil	0.401575
50	negativeSmall	positive	veryNegative	zero	nil	0.598425
51	negativeSmall	positive	negative	negativeLarge	nil	0.19685
52	negativeSmall	positive	negative	nil	nil	0.811024
53	negativeSmall	positive	positive	positiveLarge	0	0.0866142
54	negativeSmall	positive	positive	zero	nil	0.88189
55	negativeSmall	positive	veryPositive	positive	nil	0.19685
56	negativeSmall	positive	veryPositive	zero	nil	0.866142
57	negativeSmall	veryPositive	veryNegative	zero	nil	0.771653
58	negativeSmall	veryPositive	veryNegative	zero	nil	0.503937
59	negativeSmall	veryPositive	negative	zero	nil	0.944882
60	negativeSmall	veryPositive	negative	positive	0.503937	0.102362
61	negativeSmall	veryPositive	positive	zero	nil	0.700787

Table A.11 (Continued)

62	negativeSmall	veryPositive	positive	positive	nil	0.299213
63	negativeSmall	veryPositive	veryPositive	zero	nil	0.0314961
64	negativeSmall	veryPositive	veryPositive	positive	0.503937	0.393701
65	zero	veryNegative	veryNegative	zero	nil	0.11811
66	zero	veryNegative	veryNegative	negative	nil	0.897638
67	zero	veryNegative	negative	zero	nil	0.228346
68	zero	veryNegative	negative	negative	0.503937	0.551181
69	zero	veryNegative	positive	zero	nil	0.299213
70	zero	veryNegative	positive	nil	nil	0.692913
71	zero	veryNegative	veryPositive	zero	nil	0.566929
72	zero	veryNegative	veryPositive	negative	nil	0
73	zero	negative	veryNegative	zero	nil	0.307087
74	zero	negative	veryNegative	nil	nil	0.503937
75	zero	negative	negative	nil	nil	0.417323
76	zero	negative	negative	nil	nil	0.535433
77	zero	negative	positive	zero	nil	0.598425
78	zero	negative	positive	nil	nil	0.275591
79	zero	negative	veryPositive	zero	nil	0.700787
80	zero	negative	veryPositive	negative	nil	0.362205
81	zero	positive	veryNegative	zero	nil	0.535433
82	zero	positive	veryNegative	positive	0	0.0708661
83	zero	positive	negative	zero	nil	0.716535
84	zero	positive	negative	zero	nil	0.0472441
85	zero	positive	positive	zero	nil	0.503937
86	zero	positive	positive	positiveLarge	nil	0.503937
87	zero	positive	veryPositive	zero	nil	0.299213
88	zero	positive	veryPositive	positive	nil	0.700787
89	zero	veryPositive	veryNegative	zero	nil	0.519685
90	zero	veryPositive	veryNegative	positive	nil	0
91	zero	veryPositive	negative	zero	nil	0.401575
92	zero	veryPositive	negative	positive	nil	0.598425
93	zero	veryPositive	positive	zero	nil	0.307087
94	zero	veryPositive	positive	positive	nil	0.732283
95	zero	veryPositive	veryPositive	zero	nil	0.102362
96	zero	veryPositive	veryPositive	nil	0.503937	0.897638
97	positiveSmall	veryNegative	veryNegative	zero	nil	0.850394
98	positiveSmall	veryNegative	veryNegative	nil	nil	0.401575
99	positiveSmall	veryNegative	negative	zero	nil	0.700787
100	positiveSmall	veryNegative	negative	zero	nil	0.417323
101	positiveSmall	veryNegative	positive	negativeLarge	nil	0.897638
102	positiveSmall	veryNegative	positive	negative	nil	0.102362
103	positiveSmall	veryNegative	veryPositive	negative	nil	0.677165
104	positiveSmall	veryNegative	veryPositive	zero	0.503937	0.503937
105	positiveSmall	negative	veryNegative	nil	nil	0.771653
106	positiveSmall	negative	veryNegative	nil	nil	0.19685
107	positiveSmall	negative	negative	zero	nil	0.771653
108	positiveSmall	negative	negative	nil	nil	0.0944882
109	positiveSmall	negative	positive	zero	nil	0.787402
110	positiveSmall	negative	positive	positive	0.00787402	0.188976
111	positiveSmall	negative	veryPositive	zero	nil	0.700787
112	positiveSmall	negative	veryPositive	nil	nil	0.425197
113	positiveSmall	positive	veryNegative	positive	nil	0.826772
114	positiveSmall	positive	veryNegative	negativeLarge	nil	0.307087
115	positiveSmall	positive	negative	positive	nil	0.637795
116	positiveSmall	positive	negative	positiveLarge	nil	0.299213
117	positiveSmall	positive	positive	positive	nil	0.598425
118	positiveSmall	positive	positive	positiveLarge	nil	0.401575
119	positiveSmall	positive	veryPositive	positive	nil	0.598425
120	positiveSmall	positive	veryPositive	positiveLarge	nil	0.401575
121	positiveSmall	veryPositive	veryNegative	positive	nil	0.110236
122	positiveSmall	veryPositive	veryNegative	positiveLarge	nil	0.401575
123	positiveSmall	veryPositive	negative	positive	nil	0.598425
124	positiveSmall	veryPositive	negative	positiveLarge	nil	0.889764
125	positiveSmall	veryPositive	positive	positive	nil	0.503937
126	positiveSmall	veryPositive	positive	nil	nil	0.629921
127	positiveSmall	veryPositive	veryPositive	positive	nil	0.401575
128	positiveSmall	veryPositive	veryPositive	positiveLarge	nil	0.598425
129	positive	veryNegative	veryNegative	positive	nil	0.614173
130	positive	veryNegative	veryNegative	positive	nil	0.401575
131	positive	veryNegative	negative	positive	nil	0.566929
132	positive	veryNegative	negative	positiveLarge	nil	0.385827

Table A.11 (Continued)

133	positive	veryNegative	positive	nil	nil	0.724409
134	positive	veryNegative	positive	positiveLarge	0.0944882	0.401575
135	positive	veryNegative	veryPositive	positive	nil	0.645669
136	positive	veryNegative	veryPositive	negativeLarge	nil	0.566929
137	positive	negative	veryNegative	positiveLarge	nil	0.622047
138	positive	negative	veryNegative	positiveLarge	nil	0.511811
139	positive	negative	negative	positive	nil	0.503937
140	positive	negative	negative	positiveLarge	nil	0.637795
141	positive	negative	positive	nil	nil	0.96063
142	positive	negative	positive	positive	nil	0.574803
143	positive	negative	veryPositive	positive	nil	0.244094
144	positive	negative	veryPositive	negativeLarge	nil	0.692913
145	positive	positive	veryNegative	nil	nil	0.80315
146	positive	positive	veryNegative	positiveLarge	0.0393701	0.732283
147	positive	positive	negative	positive	nil	0.188976
148	positive	positive	negative	positiveLarge	nil	0.80315
149	positive	positive	positive	positiveLarge	nil	0.275591
150	positive	positive	positive	negativeLarge	0.0314961	0.905512
151	positive	positive	veryPositive	positive	nil	0.503937
152	positive	positive	veryPositive	nil	nil	0.535433
153	positive	veryPositive	veryNegative	positive	nil	0.0708661
154	positive	veryPositive	veryNegative	positiveLarge	0.763779	0.944882
155	positive	veryPositive	negative	nil	nil	0.763779
156	positive	veryPositive	negative	nil	nil	0.897638
157	positive	veryPositive	positive	negativeLarge	nil	0.897638
158	positive	veryPositive	positive	positiveLarge	nil	0.527559
159	positive	veryPositive	veryPositive	positiveLarge	nil	0.503937
160	positive	veryPositive	veryPositive	nil	nil	0.503937

Block: function performance						
Rule#	**pe**	**ce**	**h**	**fp**	**gamma**	**dos**
1	negativeLarge	negativeLarge	veryLow	nil	nil	0.598425
2	negativeLarge	negativeLarge	veryLow	high	0	0.401575
3	negativeLarge	negativeLarge	low	low	nil	0.566929
4	negativeLarge	negativeLarge	low	high	nil	0.307087
5	negativeLarge	negativeLarge	high	low	0	0.80315
6	negativeLarge	negativeLarge	high	nil	nil	0.188976
7	negativeLarge	negativeLarge	veryHigh	low	nil	0.11811
8	negativeLarge	negativeLarge	veryHigh	veryLow	nil	0.88189
9	negativeLarge	negative	veryLow	veryHigh	nil	0.0472441
10	negativeLarge	negative	veryLow	nil	nil	0.188976
11	negativeLarge	negative	low	high	nil	0.80315
12	negativeLarge	negative	low	veryHigh	nil	0.19685
13	negativeLarge	negative	high	low	nil	0.850394
14	negativeLarge	negative	high	veryLow	0	0.448819
15	negativeLarge	negative	veryHigh	nil	nil	0.19685
16	negativeLarge	negative	veryHigh	veryLow	nil	0.80315
17	negativeLarge	zero	veryLow	veryHigh	nil	0.889764
18	negativeLarge	zero	veryLow	low	nil	0.102362
19	negativeLarge	zero	low	veryHigh	0.527559	0.173228
20	negativeLarge	zero	low	high	0.535433	0.19685
21	negativeLarge	zero	high	low	0.015748	0.645669
22	negativeLarge	zero	high	nil	nil	0.629921
23	negativeLarge	zero	veryHigh	nil	nil	0.330709
24	negativeLarge	zero	veryHigh	high	0	0.692913
25	negativeLarge	positive	veryLow	high	0.338583	0.503937
26	negativeLarge	positive	veryLow	nil	0.598425	0.0314961
27	negativeLarge	positive	low	low	nil	0.897638
28	negativeLarge	positive	low	low	nil	0.811024
29	negativeLarge	positive	high	nil	nil	0.897638
30	negativeLarge	positive	high	nil	nil	0.637795
31	negativeLarge	positive	veryHigh	high	nil	0.393701
32	negativeLarge	positive	veryHigh	low	nil	0.220472
33	negativeLarge	positiveLarge	veryLow	veryLow	nil	0
34	negativeLarge	positiveLarge	veryLow	low	nil	0.818898
35	negativeLarge	positiveLarge	low	veryLow	0.11811	0.346457
36	negativeLarge	positiveLarge	low	nil	nil	0.409449
37	negativeLarge	positiveLarge	high	nil	0.133858	0.96063
38	negativeLarge	positiveLarge	high	nil	nil	0.307087
39	negativeLarge	positiveLarge	veryHigh	nil	nil	0.566929
40	negativeLarge	positiveLarge	veryHigh	nil	0.00787402	0.763779

Table A.11 (Continued)

41	negative	negativeLarge	veryLow	high	nil	0.574803
42	negative	negativeLarge	veryLow	low	nil	0.503937
43	negative	negativeLarge	low	veryLow	nil	0.535433
44	negative	negativeLarge	low	low	nil	0.905512
45	negative	negativeLarge	high	nil	nil	0.669291
46	negative	negativeLarge	high	low	nil	0.267717
47	negative	negativeLarge	veryHigh	nil	nil	0.503937
48	negative	negativeLarge	veryHigh	veryLow	nil	0.503937
49	negative	negative	veryLow	low	nil	0.629921
50	negative	negative	veryLow	high	nil	0.511811
51	negative	negative	low	low	nil	0.598425
52	negative	negative	low	high	nil	0.385827
53	negative	negative	high	low	nil	0.503937
54	negative	negative	high	nil	nil	0.503937
55	negative	negative	veryHigh	veryLow	nil	0.929134
56	negative	negative	veryHigh	low	nil	0.19685
57	negative	zero	veryLow	veryHigh	nil	0.929134
58	negative	zero	veryLow	high	nil	0.102362
59	negative	zero	low	veryHigh	nil	0.598425
60	negative	zero	low	high	nil	0.401575
61	negative	zero	high	low	nil	0.503937
62	negative	zero	high	veryLow	nil	0
63	negative	zero	veryHigh	veryLow	nil	0.700787
64	negative	zero	veryHigh	low	nil	0.362205
65	negative	positive	veryLow	high	nil	0.519685
66	negative	positive	veryLow	low	0.251969	0.755905
67	negative	positive	low	high	0.125984	0.19685
68	negative	positive	low	low	nil	0.80315
69	negative	positive	high	low	nil	0.433071
70	negative	positive	high	nil	nil	0.0944882
71	negative	positive	veryHigh	veryLow	nil	0.96063
72	negative	positive	veryHigh	low	nil	0.11811
73	negative	positiveLarge	veryLow	low	0.0314961	0.598425
74	negative	positiveLarge	veryLow	veryLow	nil	0.968504
75	negative	positiveLarge	low	low	nil	0.716535
76	negative	positiveLarge	low	veryLow	nil	0.535433
77	negative	positiveLarge	high	veryLow	0.0787402	0.84252
78	negative	positiveLarge	high	low	nil	0.401575
79	negative	positiveLarge	veryHigh	high	0.0393701	0.204724
80	negative	positiveLarge	veryHigh	low	nil	0.102362
81	zero	negativeLarge	veryLow	low	nil	0.228346
82	zero	negativeLarge	veryLow	nil	nil	0.141732
83	zero	negativeLarge	low	nil	0.314961	0.228346
84	zero	negativeLarge	low	nil	nil	0.346457
85	zero	negativeLarge	high	veryLow	nil	0.968504
86	zero	negativeLarge	high	nil	nil	0.0708661
87	zero	negativeLarge	veryHigh	veryLow	nil	0.535433
88	zero	negativeLarge	veryHigh	veryLow	nil	0.00787402
89	zero	negative	veryLow	veryLow	0	0.283465
90	zero	negative	veryLow	low	0.629921	0.629921
91	zero	negative	low	high	nil	0.724409
92	zero	negative	low	high	nil	0.401575
93	zero	negative	high	high	nil	0.700787
94	zero	negative	high	low	0.125984	0.267717
95	zero	negative	veryHigh	veryLow	0	0.96063
96	zero	negative	veryHigh	low	nil	0.0393701
97	zero	zero	veryLow	nil	nil	0.700787
98	zero	zero	veryLow	high	nil	0.299213
99	zero	zero	low	nil	nil	0.267717
100	zero	zero	low	high	nil	0.244094
101	zero	zero	high	low	0	0.582677
102	zero	zero	high	veryLow	nil	0.692913
103	zero	zero	veryHigh	nil	nil	0.19685
104	zero	zero	veryHigh	low	0.88189	0.425197
105	zero	positive	veryLow	high	nil	0.614173
106	zero	positive	veryLow	low	0.133858	0.84252
107	zero	positive	low	high	nil	0.661417
108	zero	positive	low	nil	nil	0.92126
109	zero	positive	high	nil	nil	0.574803
110	zero	positive	high	low	nil	0.0393701
111	zero	positive	veryHigh	high	0.574803	0.582677

Table A.11 (Continued)

112	zero	positive	veryHigh	veryLow	0.409449	0.582677
113	zero	positiveLarge	veryLow	nil	0.0314961	0.748031
114	zero	positiveLarge	veryLow	veryLow	nil	0.0472441
115	zero	positiveLarge	low	low	nil	0.929134
116	zero	positiveLarge	low	veryLow	nil	0.102362
117	zero	positiveLarge	high	low	nil	0.566929
118	zero	positiveLarge	high	veryLow	nil	0.629921
119	zero	positiveLarge	veryHigh	veryLow	nil	0.897638
120	zero	positiveLarge	veryHigh	low	nil	0.228346
121	positive	negativeLarge	veryLow	high	nil	0.464567
122	positive	negativeLarge	veryLow	nil	nil	0.354331
123	positive	negativeLarge	low	veryLow	nil	0.92126
124	positive	negativeLarge	low	nil	0.173228	0.952756
125	positive	negativeLarge	high	nil	nil	0.866142
126	positive	negativeLarge	high	low	nil	0.102362
127	positive	negativeLarge	veryHigh	veryLow	nil	0.511811
128	positive	negativeLarge	veryHigh	nil	nil	0.503937
129	positive	negative	veryLow	nil	nil	0.0629921
130	positive	negative	veryLow	high	0.559055	0.0551181
131	positive	negative	low	nil	0.535433	0.401575
132	positive	negative	low	veryLow	0.125984	0.724409
133	positive	negative	high	veryLow	nil	0.866142
134	positive	negative	high	nil	nil	0
135	positive	negative	veryHigh	veryLow	0.125984	0.015748
136	positive	negative	veryHigh	veryLow	nil	0.251969
137	positive	zero	veryLow	nil	nil	0.23622
138	positive	zero	veryLow	nil	nil	0.244094
139	positive	zero	low	nil	nil	0.00787402
140	positive	zero	low	high	0.488189	0.88189
141	positive	zero	high	low	nil	0.937008
142	positive	zero	high	nil	nil	0.346457
143	positive	zero	veryHigh	nil	0.535433	0.692913
144	positive	zero	veryHigh	high	0.188976	0.299213
145	positive	positive	veryLow	low	nil	0.417323
146	positive	positive	veryLow	high	0.283465	0.23622
147	positive	positive	low	low	0	0.88189
148	positive	positive	low	nil	nil	0.220472
149	positive	positive	high	veryLow	nil	0.80315
150	positive	positive	high	nil	nil	0.637795
151	positive	positive	veryHigh	veryLow	nil	0.566929
152	positive	positive	veryHigh	veryLow	nil	0.0393701
153	positive	positiveLarge	veryLow	veryLow	nil	0.503937
154	positive	positiveLarge	veryLow	low	nil	0
155	positive	positiveLarge	low	veryLow	nil	0.598425
156	positive	positiveLarge	low	nil	nil	0.401575
157	positive	positiveLarge	high	veryLow	nil	0.944882
158	positive	positiveLarge	high	veryLow	nil	0.299213
159	positive	positiveLarge	veryHigh	nil	nil	0.818898
160	positive	positiveLarge	veryHigh	nil	nil	0.850394
161	positiveLarge	negativeLarge	veryLow	high	nil	0.629921
162	positiveLarge	negativeLarge	veryLow	nil	nil	0.629921
163	positiveLarge	negativeLarge	low	high	nil	0.307087
164	positiveLarge	negativeLarge	low	veryLow	nil	0.0393701
165	positiveLarge	negativeLarge	high	low	nil	0.307087
166	positiveLarge	negativeLarge	high	veryLow	nil	0.968504
167	positiveLarge	negativeLarge	veryHigh	nil	0.889764	0.488189
168	positiveLarge	negativeLarge	veryHigh	nil	nil	0.0944882
169	positiveLarge	negative	veryLow	nil	nil	0.645669
170	positiveLarge	negative	veryLow	high	nil	0.220472
171	positiveLarge	negative	low	nil	nil	0.80315
172	positiveLarge	negative	low	nil	nil	0.125984
173	positiveLarge	negative	high	low	nil	0.401575
174	positiveLarge	negative	high	low	nil	0.15748
175	positiveLarge	negative	veryHigh	high	nil	0.0314961
176	positiveLarge	negative	veryHigh	low	nil	0.133858
177	positiveLarge	zero	veryLow	nil	nil	0.992126
178	positiveLarge	zero	veryLow	nil	nil	0.574803
179	positiveLarge	zero	low	low	nil	0.503937
180	positiveLarge	zero	low	high	nil	0.834646
181	positiveLarge	zero	high	nil	nil	0.559055
182	positiveLarge	zero	high	nil	nil	0.850394

Table A.11 (Continued)

183	positiveLarge	zero	veryHigh	veryLow	0.0314961	0.574803
184	positiveLarge	zero	veryHigh	nil	0.251969	0.204724
185	positiveLarge	positive	veryLow	nil	nil	0.929134
186	positiveLarge	positive	veryLow	low	0.629921	0.732283
187	positiveLarge	positive	low	veryHigh	0.125984	0.897638
188	positiveLarge	positive	low	nil	0.346457	0.598425
189	positiveLarge	positive	high	low	nil	0.259843
190	positiveLarge	positive	high	nil	nil	0.102362
191	positiveLarge	positive	veryHigh	veryHigh	nil	0.377953
192	positiveLarge	positive	veryHigh	low	nil	0.590551
193	positiveLarge	positiveLarge	veryLow	high	nil	0.503937
194	positiveLarge	positiveLarge	veryLow	low	nil	0.283465
195	positiveLarge	positiveLarge	low	high	0.188976	0.866142
196	positiveLarge	positiveLarge	low	high	nil	0.188976
197	positiveLarge	positiveLarge	high	nil	0.141732	0.15748
198	positiveLarge	positiveLarge	high	low	0.283465	0.779527
199	positiveLarge	positiveLarge	veryHigh	veryHigh	nil	0.519685
200	positiveLarge	positiveLarge	veryHigh	high	nil	0.503937

APPENDIX B – Fuzzy System Inputs and Data Split

Appendix B – Fuzzy System Inputs and Data Split

Table B.1[1] Fuzzy System Inputs and Data Split

scenario	normal split	resplit	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	z1	z2	z3	z4
c001	validating	training	-67.83	90.78	32693.00	-7.00	1.00	0.10	321.31	-0.34	-16.00	-2137.15	2.03	0.00	0.00	8.07
c002	validating	overtraining	-1374.99	-3698.89	32819.00	379.00	1.00	-4.43	290.45	1.71	-133.00	11661.10	2.35	0.00	0.00	5.25
c003	overtraining	training	-19.04	25.55	31137.00	7.00	1.00	0.03	310.12	0.21	-33.00	-1035.54	2.00	0.00	0.00	8.85
c004	training	overtraining	-1077.84	2654.25	30969.00	-131.00	1.00	4.28	217.67	-0.41	-307.00	-4265.35	0.00	0.00	0.00	7.12
c005	training	training	-50.66	66.60	32214.00	254.00	0.00	0.08	283.56	-1.17	142.00	2056.31	0.00	0.00	0.00	6.50
c006	validating	overtraining	-91.49	230.40	32535.00	235.00	1.00	0.37	215.30	0.55	93.00	1251.55	0.00	0.00	0.00	4.52
c007	overtraining	validating	-150.37	198.98	28064.00	-66.00	0.00	0.26	289.09	-1.15	-30.00	-3822.61	0.00	0.00	0.00	7.47
c008	overtraining	validating	388.03	-1469.79	26028.00	-182.00	0.00	-2.36	237.73	-0.27	3.00	1354.73	0.00	0.00	0.00	1.83
c009	validating	validating	95.41	-290.90	29356.00	116.00	0.72	-0.76	135.61	0.26	262.00	-2262.17	0.00	1.23	0.00	6.98
c010	validating	training	216.56	-634.66	24621.00	121.00	1.00	-1.05	235.52	1.36	-187.00	8016.36	0.00	0.00	0.00	7.25
c011	validating	overtraining	-366.46	-675.92	26877.00	307.00	1.00	-1.24	204.64	1.37	27.00	7301.80	0.22	0.00	0.00	8.15
c012	training	validating	167.29	155.09	25899.00	69.00	0.00	0.39	148.33	-1.11	170.00	-2123.96	0.00	4.97	0.00	3.00
c013	training	training	-109.64	451.57	20274.00	-86.00	0.00	0.69	274.76	-1.30	-269.00	-3655.69	0.00	0.00	0.00	7.47
c014	validating	validating	176.29	-621.81	21688.00	-2.00	0.00	-1.34	188.62	-0.65	202.00	-2042.75	0.00	0.20	0.00	9.13
c015	overtraining	overtraining	93.38	158.47	19122.00	-58.00	0.54	0.46	146.79	0.10	16.00	-1729.02	0.00	1.50	0.00	7.23
c016	training	training	-101.02	173.28	15892.00	-38.00	0.92	0.43	180.24	1.50	-61.00	-1757.76	0.00	0.00	0.00	9.65
c017	training	validating	-156.19	521.64	11886.00	56.00	0.48	0.83	301.48	-0.21	-35.00	-3515.54	0.03	0.00	0.00	8.32
c018	overtraining	training	33.46	-156.23	9442.00	-18.00	0.49	-0.24	324.93	0.03	49.00	588.22	0.55	0.00	0.00	9.08
c019	overtraining	overtraining	-96.29	76.56	4510.00	140.00	0.39	0.14	299.22	-0.09	30.00	3184.52	0.00	0.00	0.00	5.13
c020	overtraining	training	37.00	1432.57	8794.00	14.00	0.00	3.79	188.02	-2.28	-19.00	-1831.52	0.00	0.85	0.22	4.50
c021	training	overtraining	-57.62	62.28	5429.00	139.00	0.28	0.14	232.11	-0.04	199.00	-2907.45	0.00	0.18	0.00	3.28
c022	training	validating	52.45	1684.02	7201.00	-189.00	0.81	6.38	134.21	-0.12	-294.00	-1873.61	0.00	3.10	0.00	6.30
c023	training	training	79.08	1453.63	7151.00	51.00	1.00	5.14	143.89	1.02	0.00	-1862.59	0.00	1.50	0.00	4.63
c024	overtraining	training	-60.53	-839.22	3684.00	-116.00	1.00	-1.89	238.73	3.57	-289.00	6435.99	2.73	0.00	0.73	2.32
c025	training	validating	138.36	-516.76	3654.00	-106.00	0.00	-1.81	153.01	-0.80	36.00	-1902.18	0.00	2.23	0.65	3.92
c026	validating	overtraining	-150.53	541.11	3195.00	55.00	0.24	2.35	124.08	-0.95	-39.00	2.82	0.00	8.88	2.18	1.23
o001	validating	training	2.59	3648.03	3121.00	-29879.00	0.92	6.89	286.63	-0.33	-30237.00	-4033.23	0.00	0.00	0.00	9.37
o002	overtraining	validating	4.63	1485.74	3041.00	-13059.00	0.84	2.50	322.65	0.70	-13202.00	-1381.57	3.48	0.00	0.00	7.15
o003	training	validating	0.45	-1867.48	3023.00	-11177.00	0.17	-3.20	316.69	0.15	-10907.00	-3589.76	3.18	0.00	2.85	1.75
o004	overtraining	training	-5.00	-4233.64	2867.00	-23733.00	0.00	-6.90	333.87	0.72	-23202.00	-2487.09	8.80	0.00	3.62	1.42
o005	validating	training	-707.79	-3767.61	4382.00	-3818.00	0.00	-6.20	323.48	0.57	-4204.00	10575.50	4.97	0.00	1.70	2.03
o006	training	training	-50.90	-3190.30	4671.00	571.00	0.00	-5.14	329.30	0.09	530.00	7025.46	3.20	0.05	0.98	7.53
o007	training	training	0.23	779.65	5305.00	-8795.00	0.66	1.20	341.90	0.05	-8899.00	964.97	8.77	0.00	0.68	1.30
o008	training	validating	-31.06	-518.04	7277.00	5277.00	0.56	-0.86	307.45	0.83	5026.00	7851.11	2.00	0.00	0.00	6.58
o009	training	training	-8.99	-5400.61	6896.00	-22104.00	0.00	-8.00	347.62	1.04	-21412.00	-4688.22	10.00	0.00	1.63	1.07
o010	overtraining	overtraining	-7.42	-1119.28	7543.00	-22357.00	1.00	-1.70	335.77	2.38	-22212.00	-1247.93	9.53	0.00	1.17	0.32
o011	overtraining	training	-457.38	-1994.83	9112.00	-15188.00	1.00	-3.39	293.10	3.33	-15699.00	11460.30	4.75	0.00	1.13	0.20
o012	training	overtraining	-20.84	-3739.44	10092.00	1792.00	0.00	-5.58	329.77	0.33	1993.00	4227.61	4.98	0.00	0.00	5.83
o013	validating	training	-14.38	-2274.24	12661.00	10061.00	1.00	-3.30	326.44	2.65	10391.00	-4430.82	7.93	0.00	0.87	1.73
o014	overtraining	training	-87.57	-3972.32	13019.00	3319.00	1.00	-6.14	304.50	3.73	3507.00	4372.98	6.15	0.00	0.00	2.22
o015	validating	validating	-48.92	-926.53	18551.00	-12749.00	1.00	-1.33	301.25	1.61	-12914.00	6850.45	3.22	0.00	0.00	0.00
o016	validating	validating	-2115.70	-1816.15	22303.00	9503.00	1.00	-2.35	318.25	1.54	8707.00	13905.30	4.60	0.00	0.00	5.62
o017	training	validating	4.38	-245.35	21680.00	-4620.00	1.00	-0.31	333.58	0.79	-4534.00	-2835.31	7.23	0.00	0.00	1.47
o018	validating	overtraining	16.00	4571.93	24037.00	-2063.00	1.00	6.55	277.44	-1.12	-2499.00	-2289.73	0.00	0.00	0.00	8.85
o019	training	overtraining	15.81	4553.19	24403.00	21303.00	1.00	6.55	274.62	-1.12	20860.00	-1697.45	0.83	0.82	0.00	0.00
o020	validating	overtraining	-1.76	-1356.37	24161.00	-839.00	1.00	-1.70	318.45	1.25	-645.00	-2173.81	5.50	0.00	0.00	0.22
o021	overtraining	overtraining	-1720.57	-2055.39	25588.00	13488.00	1.00	-2.73	293.09	1.83	12782.00	13146.50	4.37	0.00	0.00	4.22
o022	overtraining	training	-15.46	-4840.93	28289.00	16389.00	1.00	-5.40	338.66	1.89	17061.00	-5545.48	9.52	0.00	0.00	1.80
o023	validating	validating	4.28	5134.19	29639.00	6239.00	1.00	6.27	300.19	-1.59	5669.00	-601.68	0.00	0.00	0.00	0.00
o024	training	validating	12.39	143.50	30293.00	1893.00	0.77	0.18	288.89	0.09	1929.00	-1873.03	0.00	0.00	0.00	1.27
o025	overtraining	overtraining	-360.29	-1771.31	32781.00	13181.00	1.00	-2.15	286.25	1.09	12759.00	10460.30	3.75	0.00	0.00	6.48
o026	training	overtraining	-40.13	388.55	33322.00	-1178.00	0.77	0.48	281.11	-0.09	-1383.00	5323.36	0.00	0.00	0.00	6.05
s001	validating	training	1.48	-449.02	2634.00	-1466.00	0.00	-1.99	123.03	-0.79	-1408.00	-379.37	0.00	8.95	1.72	0.00
s002	validating	training	-38.05	3307.65	3903.00	-17897.00	1.00	12.79	138.24	-1.20	-18394.00	4710.47	0.00	4.37	0.28	3.87
s003	validating	overtraining	1.89	-544.78	4124.00	-8176.00	0.12	-2.42	120.07	-0.04	-8100.00	-642.04	0.00	9.22	2.77	0.10
s004	validating	overtraining	3.32	-1627.88	6243.00	-19057.00	0.00	-3.71	226.98	0.17	-18809.00	-3094.33	0.00	0.00	2.57	0.00
s005	validating	validating	55.83	3143.70	7842.00	-12958.00	1.00	12.91	122.66	-1.73	-13211.00	-2027.26	0.00	8.87	0.18	1.95
s006	training	validating	17.59	-1148.58	8913.00	-3987.00	0.05	-4.02	141.63	0.21	-3796.00	-2063.71	0.00	3.53	1.63	1.05
s007	training	training	-22.47	750.68	9705.00	1305.00	0.49	2.07	178.19	-0.05	1100.00	4586.47	0.00	1.15	0.00	3.17
s008	training	overtraining	14.10	2193.07	9896.00	-20604.00	0.79	5.16	208.10	-0.06	-20823.00	-822.92	0.00	0.00	0.00	8.37

[1] Variables x1 to x10 relate to variables defined in table 6.1 and represent numerical inputs to the fuzzy systems. Variables z1 to z4 relate to variables y1 to y4 of table 6.2 and represent the average human assessment for the corresponding measure. Scenario starting with character c, o, s, t belong to the capture altitude, overspeed hazard, stall hazard, and terrain hazard scenario group respectively. Column 2 and 3 indicate how the scenarios were split under normal and resplit conditions.

Table B.1 (Continued)

s009	training	validating	-5.59	1673.28	11286.00	3686.00	0.75	4.62	173.22	-0.12	3436.00	2366.42	0.00	2.23	0.00	0.00
s010	overtraining	overtraining	-15.51	4188.96	11535.00	-16665.00	1.00	16.20	123.00	-3.01	-17102.00	-261.04	0.00	9.98	0.00	0.27
s011	training	training	22.60	2662.31	11885.00	2485.00	1.00	9.77	128.98	-0.89	2197.00	731.87	0.00	7.57	0.00	0.15
s012	overtraining	overtraining	29.39	3690.55	12345.00	2845.00	1.00	12.56	138.07	-1.86	2499.00	-938.28	0.00	7.32	0.00	0.18
s013	validating	training	-67.08	3614.95	13631.00	-10469.00	1.00	13.42	124.02	-2.01	-11010.00	4791.00	0.00	8.95	0.00	1.03
s014	overtraining	training	-66.11	845.82	14806.00	-15894.00	0.66	2.96	129.11	-0.04	-16116.00	3698.67	0.00	5.43	0.00	2.33
s015	overtraining	validating	2.79	-1620.26	15074.00	-2326.00	0.00	-3.31	221.87	0.09	-2103.00	-2096.91	0.00	0.00	0.00	0.47
s016	training	overtraining	-36.20	875.04	15806.00	11806.00	1.00	2.99	130.36	0.45	11613.00	3232.53	0.00	1.92	0.00	2.75
s017	overtraining	validating	43.48	1263.42	16289.00	-10911.00	0.80	3.98	140.08	-0.06	-11001.00	-852.32	0.00	3.42	0.00	5.55
s018	training	training	-36.44	7839.23	18542.00	7642.00	1.00	16.36	205.30	-3.70	6885.00	-4134.20	0.00	2.50	0.00	0.13
s019	overtraining	overtraining	-19.16	3069.35	19193.00	-13707.00	1.00	9.45	136.71	-1.25	-14119.00	2963.42	0.00	6.50	0.00	2.38
s020	training	validating	-58.37	8545.05	22715.00	18115.00	1.00	16.75	204.33	-4.06	17192.00	-792.13	0.00	1.35	0.00	0.00
s021	validating	training	39.66	855.50	26495.00	6695.00	0.97	2.61	121.95	0.00	6629.00	-229.41	0.00	7.02	0.00	0.00
s022	training	training	33.62	4179.17	29239.00	7439.00	1.00	7.41	202.85	-1.28	7000.00	0.00	0.00	1.30	0.00	0.00
s023	training	validating	-125.98	2223.98	30819.00	27419.00	1.00	5.93	129.25	-1.02	26945.00	6112.50	0.00	5.83	0.00	2.90
s024	overtraining	training	-1.32	895.79	30442.00	342.00	1.00	2.59	119.83	-0.18	202.00	1651.32	0.00	9.58	0.00	0.15
s025	overtraining	validating	48.42	-50.47	31232.00	17532.00	0.72	-0.12	144.92	0.03	17602.00	-1537.08	0.00	1.85	0.00	0.80
s026	validating	overtraining	23.64	635.95	31757.00	22357.00	1.00	1.61	134.17	-0.07	22300.00	-39.70	0.00	4.05	0.00	0.13
t001	overtraining	training	-30.78	-1626.50	2121.00	2121.00	0.00	-6.86	130.12	1.01	2299.00	-416.70	0.00	1.92	1.90	6.47
t002	training	validating	-29.94	-1905.89	976.00	976.00	0.00	-7.71	137.90	1.31	1197.00	-841.76	0.00	0.85	2.98	5.78
t003	validating	training	37.01	2282.25	2716.00	-24084.00	0.96	8.88	139.86	0.07	-24302.00	-556.27	0.00	0.28	0.18	8.88
t004	overtraining	overtraining	4.37	3559.41	3068.00	3068.00	1.00	14.00	137.52	-1.61	2698.00	-397.02	0.00	5.17	2.72	0.95
t005	validating	overtraining	-22.97	-1596.40	643.00	-10357.00	0.00	-5.81	154.13	0.64	-10202.00	404.92	0.00	0.80	7.87	0.00
t006	training	validating	-11.60	-360.12	599.00	-6401.00	0.23	-1.24	162.60	0.20	-6398.00	1133.94	0.00	0.28	3.38	0.87
t007	overtraining	validating	-15.02	-1659.66	2023.00	-24177.00	0.00	-5.35	170.42	0.54	-24002.00	-111.66	0.27	0.00	2.18	0.30
t008	overtraining	overtraining	-49.52	-1762.73	632.00	-23368.00	0.12	-5.45	181.26	1.39	-23297.00	3301.26	0.00	0.00	8.53	0.00
t009	training	training	-20.01	-2708.88	543.00	-9357.00	0.00	-7.70	197.49	1.63	-9000.00	-2655.15	0.78	0.08	9.57	0.10
t010	training	training	11.18	3743.67	1186.00	-8014.00	1.00	11.62	179.25	-0.66	-8398.00	-1267.61	0.00	1.52	0.78	6.43
t011	validating	overtraining	-20.65	-2896.90	1034.00	-30066.00	0.00	-7.60	212.61	1.59	-29697.00	-2231.89	0.00	0.00	7.00	0.00
t012	overtraining	validating	-70.40	5127.52	1411.00	1411.00	1.00	17.09	166.39	-2.61	685.00	4808.39	0.00	5.02	2.00	0.23
t013	validating	validating	7.10	4033.77	1313.00	-19187.00	1.00	11.74	190.82	-0.71	-19603.00	-1547.01	0.00	1.10	0.83	8.37
t014	training	training	-18.65	-2959.95	742.00	-2058.00	0.00	-7.09	233.61	1.45	-1703.00	-1212.07	0.18	0.00	8.17	0.00
t015	validating	overtraining	-84.41	-1273.48	2722.00	2722.00	0.46	-2.98	233.00	2.04	2565.00	7052.10	0.00	0.00	0.00	7.47
t016	training	validating	-1.09	-1878.71	2339.00	-17261.00	0.00	-4.30	239.11	0.30	-17003.00	-2202.81	0.83	0.02	1.47	1.17
t017	validating	validating	-2.29	1794.33	2608.00	-28692.00	0.59	4.30	227.25	-0.13	-28961.00	3159.12	0.00	0.00	0.00	6.25
t018	overtraining	validating	-11.33	-2849.20	99.00	-8501.00	0.00	-6.50	247.66	1.19	-8111.00	-2885.38	0.42	0.00	10.00	0.00
t019	overtraining	overtraining	-106.80	6742.67	1755.00	1755.00	1.00	18.80	197.87	-3.51	779.00	7492.91	0.00	1.52	0.83	0.02
t020	validating	overtraining	-196.40	-3546.46	2391.00	2391.00	0.54	-7.62	254.52	4.05	2274.00	8055.90	2.70	0.00	2.90	5.63
t021	validating	training	-18.88	-2251.63	1124.00	1124.00	0.00	-5.07	247.38	0.55	1213.00	4392.91	0.33	0.00	2.05	5.70
t022	training	training	-33.43	-742.79	807.00	807.00	0.88	-1.56	267.06	3.26	605.00	7325.82	2.32	0.00	3.05	5.22
t023	training	training	-33.42	6233.59	3369.00	-11631.00	1.00	13.98	240.51	-2.04	-12306.00	-3146.11	0.00	1.52	1.20	6.05
t024	training	overtraining	-39.53	6089.63	1953.00	-3047.00	1.00	14.00	239.40	-1.91	-3902.00	7035.63	0.00	1.18	0.00	6.67
t025	training	training	-16.31	-3513.26	806.00	-14394.00	0.00	-7.30	269.40	1.32	-13995.00	-47.09	0.93	0.00	7.92	0.00
t026	overtraining	training	-3.21	-2510.26	179.00	179.00	0.00	-5.20	272.50	0.48	499.00	-1630.41	0.83	0.00	9.63	0.00

APPENDIX C – Questionnaires and Task Description

APPENDIX C – Questionnaire and Task Description

Pre-Test Questionnaire

Subject # ______________________________ **Date:** ____________________

1. Current seat

☐ captain
☐ first officer
☐ flight engineer
☐ other (e.g., check pilot, instructor):

____

2. Flying experience

List all aircraft you have flown:

____
____
____
____
____

Current aircraft:

____

Total flying time: __________ hours

3. Certificates/Ratings

☐ student
☐ private
☐ instrument
☐ multi-engine
☐ commercial
☐ CFI
☐ ATP
☐ F/E
☐ other: ______________

4. Caffeine and medications

How many cups of coffee, tea, or other caffeine-containing beverage have you had today? _____

Are you taking any medication that is likely to affect your flying and decision making skills?

☐ yes
☐ no

5. Other conditions

Did you get adequate sleep last night?

☐ yes
☐ no

Do you have a cold or other illness that might affect your performance today?

☐ yes
☐ no

1. Are there any other factors that might affect your performance today?

____
____
____
____
____

Post-Test Questionnaire

Subject # ______________________________ **Date:** ____________________

6. **Did you receive adequate training to serve the purpose of this experiment, as you understand it?**

☐ yes

☐ no

Explain:

7. **Rate how realistic you found the scenarios according to the scale below.**

Unrealistic Somewhat Realistic Realistic

0 5 10

Your rating: _____

8. **What factors made the scenarios unrealistic?**

9. **Rate how clearly the metrics were defined according to the scale below.**

Unclear Moderately Clear Clear

0 5 10

Terrain Hazard: _____
Stall Hazard: _____
Overspeed Hazard: _____
Pilot Response Accuracy: _____

10. **What were the difficulties you had with understanding any of the above metrics?**

11. **What other comments, criticisms, and suggestions do you have concerning the research or your experience with the experiment?**

Task Description

In a number of scenarios you are going to see how a pilot is performing the task of ensuring flight safety and capturing a target altitude. Based on the control actions of the pilot, state information displayed on the primary flight display, and the target altitude, you are to rate the pilot's activities and assess potential hazards that may be present at the end of the scenario.

Assumptions and Constraints

The employed simulator models a two jet engine civil transport airplane.

Speeds

Assume the following speeds:
- Stall speed: 120 knots indicated airspeed
- Overspeed: 340 knots indicated airspeed

Environment

Assume the following environment conditions:
- No winds
- No mountains
- Ground impact occurs at sea level (0 ft on altitude tape)

Priorities

Assume the following priorities:
1) Safety
2) Target

Note: Some scenarios may require the pilot to respond to a hazardous situation. In those cases deviation from the target may be acceptable.

Explanation of Assessment Metrics

Terrain Hazard

Terrain hazard quantifies the danger of fatal ground impact you think is present at the end of the scenario. Note: A low altitude stall condition may pose a terrain hazard.

Stall Hazard

Stall hazard quantifies the danger of stall you think is present at the end of the scenario.

Overspeed Hazard

Overspeed hazard quantifies the danger of overspeed you think is present at the end of the scenario.

Pilot Response Accuracy

Pilot response accuracy quantifies how accurate the pilot is responding to a given target altitude and initial situation. Consider the following points:

General:
To what degree is the pilot doing the right thing?

Hazard specific:
Does the initial situation pose any hazards to be resolved and is the pilot acting accordingly?
Is the pilot creating a hazardous situation?
If the initial situation is hazardous is the pilot deviating from the target and is this deviation justified?

Target specific:
Is the airplane heading in the right or wrong direction?
Is the airplane progressing towards the target in a timely manner?
How accurately is the pilot capturing the target altitude?