ANALYSIS OF
COMPLEXITY EVOLUTION MANAGEMENT AND
HUMAN PERFORMANCE ISSUES IN
COMMERCIAL AIRCRAFT
AUTOMATION SYSTEMS

Sanjay S. Vakil and R. John Hansman

MIT International Center for Air Transportation

Department of Aeronautics & Astronautics

Massachusetts Institute of Technology

Cambridge, MA 02139 USA

ANALYSIS OF COMPLEXITY EVOLUTION MANAGEMENT AND HUMAN PERFORMANCE ISSUES IN COMMERCIAL AIRCRAFT AUTOMATION SYSTEMS

SANJAY S. VAKIL AND R. JOHN HANSMAN, JR.

Abstract

Autoflight systems in the current generation of aircraft have been implicated in several recent incidents and accidents. A contributory aspect to these incidents may be the manner in which aircraft transition between differing behaviours or "modes." The current state of aircraft automation was investigated and the incremental development of the autoflight system was tracked through a set of aircraft to gain insight into how these systems developed. This process appears to have resulted in a system without a consistent global representation.

In order to evaluate and examine autoflight systems, a "Hybrid Automation Representation" was developed. This representation was used to examine several specific problems known to exist in aircraft systems. Cyclomatic complexity is an analysis tool from computer science which counts the number of linearly independent paths through a program graph. This approach was extended to examine autoflight mode transitions modelled with the Hybrid Automation Representation. A survey was conducted of pilots to identify those autoflight mode transitions which airline pilots find difficult. The transitions identified in this survey were analyzed using cyclomatic complexity to gain insight into the apparent complexity of the autoflight system from the perspective of the pilot. Mode transitions which had been identified as complex by pilots were found to have a high cyclomatic complexity.

Further examination was made into a set of specific problems identified in aircraft: the lack of a consistent representation of automation, concern regarding appropriate feedback from the automation, and the implications of physical limitations on the autoflight systems. Mode transitions involved in changing to and leveling at a new altitude were identified across multiple aircraft by numerous pilots. Where possible, evaluation and verification of the behaviour of these autoflight mode transitions was investigated via aircraft-specific high fidelity simulators. Three solution approaches to concerns regarding autoflight systems, and mode transitions in particular, are presented in this thesis. The first is to use training to modify pilot behaviours, or procedures to work around known problems. The second approach is to mitigate problems by enhancing feedback. The third approach is to modify the process by which automation is designed. The Operator Directed Process forces the consideration and creation of an automation model early in the design process for use as the basis of the software specification and training.

3

Acknowledgements

Table of Contents

List of Figures

12

13

List of Tables

18

Chapter 1

Introduction and Motivation

Advances in computation, algorithmic, and sensor capabilities have driven a trend towards more automation in dynamic systems. In particular, the commercial aircraft cockpit has been augmented by automation, causing changes to the task of flying an aircraft. Current advanced commercial transport aircraft, such as the Boeing B777/B747-400, the Airbus A320/A340, and the McDonnell Douglas MD-11, rely on AutoFlight Systems (AFS) for flight management, trajectory control, and interaction with control surfaces (Boeing 1986, 1989, 1997; Honeywell 1992, 1994). These systems have evolved from simple autopilots, such as the single axis autopilots created by Sperry in 1912 (McRuer, 1973) to multiple processor systems capable of sophisticated and interrelated tasks such as those that are used in the Boeing B777 cockpit. These tasks span the range from high-level flight management to low-level control of individual actuators.

Aircraft automation has been designed to improve performance and to increase flight safety. Performance can be increased by allowing more accurate tracking of altitude and path targets, cost can be reduced by flying algorithmically optimized fuel efficient paths, and sensors can be used to warn pilots or deal directly with unsafe situations. Flight safety can be enhanced by automatically performing critical maneuvers, by not allowing the aircraft to perform possibly dangerous maneuvers, or by augmenting the control characteristics to make the aircraft easier to fly. However, automation has also become a potential safety liability. The rapid evolutionary development of autoflight systems in commercial transport aircraft is suspected as a contributory factor in a number of incidents and accidents. Hull losses have occurred at France (Strasbourg, 1992), India (Bangalore, 1990), Japan (Nagoya, 1994), and Colombia (Cali, 1995). Numerous autoflight-related incidents have also occurred, including a rapid pitch-up (Orly, 1994), multiple incidents of overspeeds, and numerous large altitude deviations.

As automation systems become more capable, the human element may become a limiting factor in system operation and design. If so, procedures and design processes may need to be modified to acknowledge known limitations. This issue is likely to be particularly critical in

future generations of aircraft automation. Work is currently underway on the next generation of cockpits. Clearly stating the issues and solutions may improve safety and prevent costly fixes once the next generation is flying.

Many modern dynamically controlled systems use humans in a supervisory manner by having them monitor the automation which is performing the task rather than performing the task directly (Sheridan, 1992). Nuclear power plants, process control plants, and air traffic control are additional examples of supervisory systems. This thesis uses the commercial aviation environment as a case study to identify and discuss issues which may be important in other fields which use automation to support humans in supervisory systems.

1.1   Accidents and Incidents

One of the goals of aircraft autoflight systems was an increase in safety. Each successive generation of aircraft has become safer, in aggregate, than the previous generations. Figure 1.1 shows data compiled by Boeing depicting hull loss accident rates in commercial fleets (Boeing Data, 1998). This data is grouped by generation of aircraft airframes and shows a general reduction in the accident rate between generations.

The generations are based on airframe as well as automation capability. The first generation consists of early commercial jet transports, many of which have been retired from service. The second generation is comprised of widebody jets and shows a marked reduction in accident rate. The third generation consists of the first wave of "glass cockpit" aircraft, but does not include those which are fly-by-wire. Finally, the fourth generation consists of most currently manufactured narrow and widebody aircraft. Limited data exists for the most recent aircraft such as the A330, A340, MD-90 (now B717), and the B777. It is important to note that this chart documents "rare" events, and so the statistical relevance is minimal and care must be taken when observing trends.

In spite of the overall reduction in accident rate, flight crew error still appears as the dominant factor in hull loss accidents. The impact that pilots have on aircraft safety have been recognized for some time. Figure 1.2 show a breakdown of data from 2032 incidents reported over 1959-1997, also generated from Boeing data (Boeing Data, 1998). Over this time period, flight crew

irst Generation

727 .2
rident *
C-10 *
AC 1-11 .9
C-9 .5
737-1/-2 .5
-28 .6

econd Generation

C-10 .2
-1011 .5
300 .0 .6
747-1/-2/-3/-SP

hird Generation

D-80 .3
767 .4
757 .5
310 .4
Ae 146 .6
300-600 .2
737-3/-4/-5 .5
-100/-70 .7
747-400 .0

ourth Generation

320/319/321 .0
D-11 .0
340*
330*
D-90*
777*

verall .4

<100 000 Departures
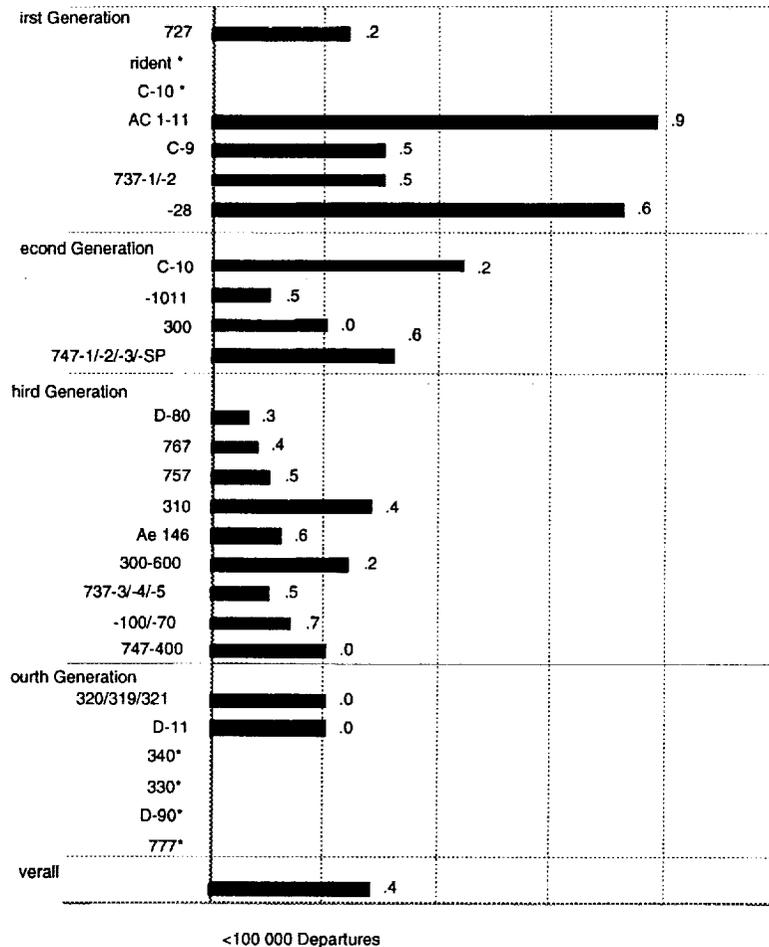
Figure 1.1:   Commercial Fleet Hull Accident Rate (per million departures) 1988-1997 (Boeing Data, 1998)

has remained the primary cause of aircraft hull loss accidents as determined by the investigative authority. The second set of data on this graph shows that the accident causes from 1988-1997 have not changed significantly and that the fraction of accidents attributed to the flight crew has remained largely stable at about 70%.

Within the set of errors attributed to flight crews, automation problems are emerging as a key safety area. The incorporation of new flight automation has resulted in a new set of human factors issues. Sufficient concerns have been raised to warrant government investigation in the form of the *1996 FAA Report on the Interfaces Between Flightcrews and Modern Flight Deck Systems.*
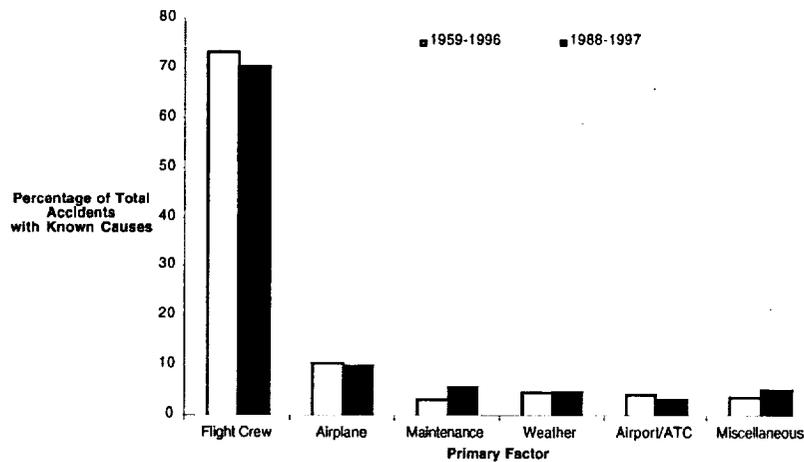
Figure 1.2:   Primary Causes of Aircraft Accidents (Boeing Data, 1998)

This document also discusses human factors and interface issues, which include mode awareness problems (Vakil, 1995), incomplete pilot understanding of automation (Sarter 1992; Weiner, 1988; Vakil 1996; Javaux 1998; others), and loss of automation situation awareness (Endsley, 1994, 1995). In contrast to mechanical aircraft failure, these problems appear to be based in confusion between the pilots' expectations of the autoflight system and what the system is actually doing.

Review of Aviation Safety Reporting System

Flight crew automation issues were examined through the use of the Aviation Safety Reporting System (ASRS), a volunteer mechanism for documenting problems in flight operations with a degree of amnesty. A search was performed on the ASRS database by researchers at the MIT Aeronautical Systems Laboratory (Vakil, Vaneck, and Midkiff, 1995) from the years 1990-94 with a set of keywords designed to elicit problems related to mode awareness. The keywords consisted of the following: annunciation, annunciator, FMC, flight management computer, FMS, flight management system, CDU, mode, capture, arm, automatic flight system, vertical, horizontal, and program. A total of three hundred ASRS reports were returned by the keyword search. After analysis, 184 were categorized as appropriate to flight crew automation issues.

The most commonly reported errors were "Programming Errors," "Mode Transition Problems," and "Insufficient Understanding of Automation." It can be argued that dominance of

22
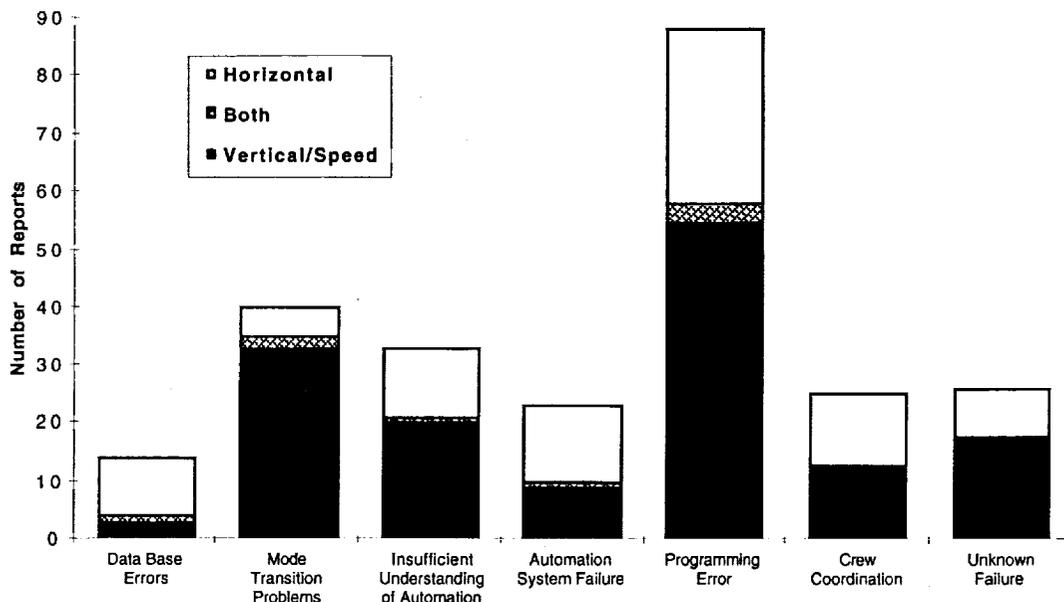
Figure 1.3:   Breakdown of ASRS reports into Perceived Causes and Flight Domain

the "Programming Errors" category may be overstated, since a single typographical error could cause an ASRS filing. However, if a such a minor error can lead to a filing, it may be indicative of an additional concern: the usage of automation can allow relatively minor errors on the part of the human to have significant repercussions. While this is not a new phenomenon, automation may have made these sorts of errors more likely to occur.

The dominant causal areas are of particular importance because they suggest there can be confusion between the pilots' expectations of the automation and what it is actually doing. "Mode Transition Problems" indicate that pilots may not realize when the automation changes its behaviour or the implications of the new behaviour. "Insufficient Understanding of Automation" is equally problematic since it suggests that the pilots may not be able to supervise the automation: in order to effectively monitor automation, a pilot must understand what its intended behaviour should be.

As shown in Figure 1.3, these reports were also categorized by the perceived cause of the problem and by the flight path (vertical/speed, horizontal or both) that was impacted. Since the vertical flight path and the speed are implicitly coupled, they were grouped together. In instances

where the problems spanned multiple causal categories, the reports were counted in each category. In Figure 1.3, it can be seen that vertical/speed problems dominate many of the categories; of all categories 62.7% of the reports were of this type. In particular, the "Mode Transition Problems" category is dominated by vertical/speed problems. The data classified into the "Insufficient Understanding of Automation" also suggests a deficiency in knowledge of the vertical domain automation.

It should be noted that there exists a potential for over-reporting vertical deviations. Air Traffic Control (ATC) radar can measure altitude much more precisely than location. This may result in pilots reporting vertical/speed incidents more often than lateral ones.

1.2   Introduction to Service Problems

As new, complex automation systems are introduced into operation, problems are discovered early during operation and dealt with through training and procedural changes (Weiner, 1985). This process of fixing issues as they appear results in incidents early in the aircraft lifetime, typically after introduction. However, this does lead to a stable set of automation within which all of the problems have been identified. These identified problems can then be dealt with through training, procedural changes, or automation modification. Underlying this process is an implicit higher failure rate early in operational usage, rather than later as mechanical failure appears.

Figure 1.4 shows hull losses or fatal accidents for aircraft from 1959-1997 organized by the number of years since introduction of an aircraft type that the accident occurred. In a manner consistent with Figure 1.1, the number of accidents has been decreasing with successive generations. However, each generation shows a spike a short time after introduction. This increase corresponds with problems that are found early in the operational life of the aircraft that were not foreseen before they were put into active usage.

Figure 1.5 shows the Hull Loss Accident Rate of the worldwide commercial aircraft fleet from 1988-1997 by individual aircraft. Multiple aircraft that have been introduced since 1981 are shown with the accident rate per one million departures. While noting that the statistical significance of the information in this chart is limited since hull loss accidents are rare events, there is a significant difference between the introduction of the Airbus A319/320/321 series as

Accident rate (accidents per million departures)

60
50
40
30
20
10
0

First generation
Second generation
Early widebody
Current generation

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
Years since introduction
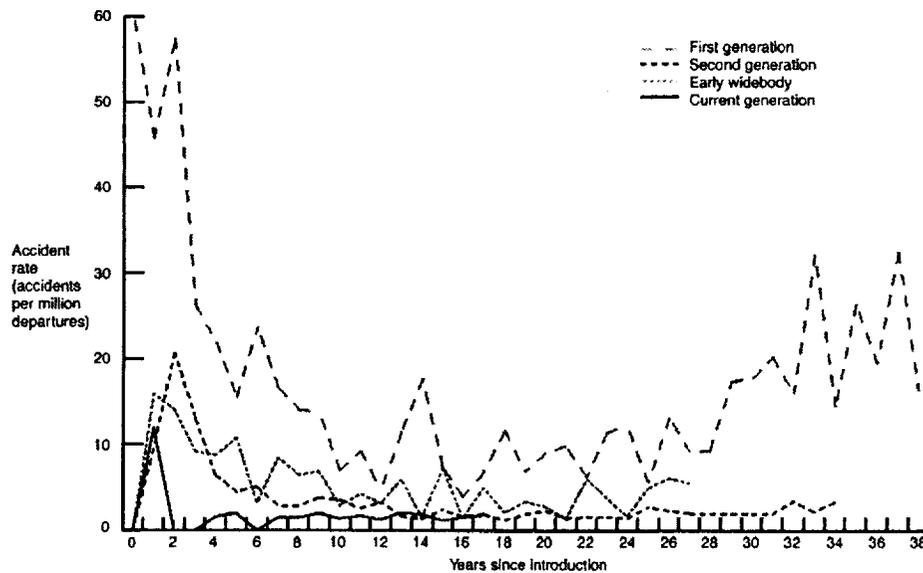
Figure 1.4: Worldwide Commercial Fleet Hull Loss and/or Fatal Accident Rates by Years Following Introduction 1959-1997 (Boeing Data 1998)

compares to other aircraft. Part of the reason behind the anomalous nature of the A320 derivative record is an early hull loss during operational usage at the Habsheim airshow on June 26th, 1988. However, the accident rate of this aircraft also appears to be improving as pilots and airlines gain more experience with its detailed behaviour, and as these details are disseminated. A hypothesis for this improvement is that training material, procedures, and flight crews are becoming more proficient with the aircraft. This is consistent with the nature of the A320, which was the first fully digital commercial fly-by-wire aircraft. It also included numerous departures from previous designs, such as a full authority envelope protection system, a side-stick controller, and non-moving throttles. This experience also explains the lack of hull losses of the recently introduced A330 and A340, which have very similar cockpit automation.

1.2.1 Questionnaire on Pilot Understanding of Boeing B757

Work has also been done to gauge pilot understanding of flight automation. This work looked at pilots' understanding of the Boeing B757 early in its operational lifetime. The results indicate that pilots did not feel that they completely understood the aircraft (Weiner, 1985). The research consisted of a questionnaire ("Phase 1") designed to probe pilots' opinions, experience levels,
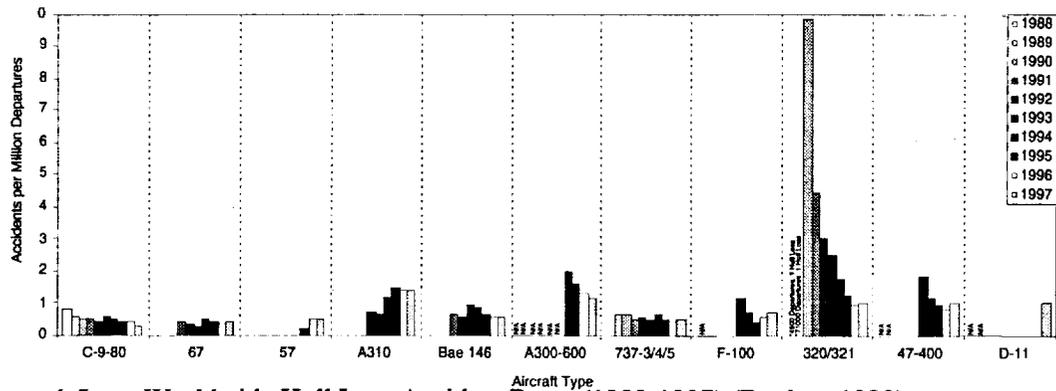
Figure 1.5: Worldwide Hull Loss Accident Rates (1988-1997) (Boeing, 1998)

specific information and viewpoints on the new "glass cockpit" technology that was distributed to Boeing B757 pilots at a pair of carriers.
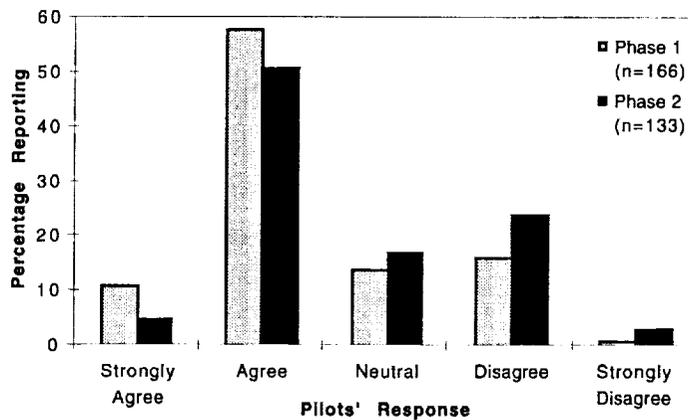
Figure 1.6: In the B757 automation, there are still things that happen that surprise me. (adapted from Weiner, 1985)

A follow-up questionnaire ("Phase 2") was distributed a year later. The second questionnaire is interesting in that it shows insight into the effect of familiarity, practice and experience with the technology. Each questionnaire consisted of a large set of questions organized using the Likert scale to assess the pilot attitude. In this study, five response levels were employed. The response to the statement "In the B-757 automation, there are still things that happen that surprise me." is shown in Figure 1.6. The particularly striking point of this graph is that after a year flying the aircraft, over half of the pilots were still being surprised by the automation.
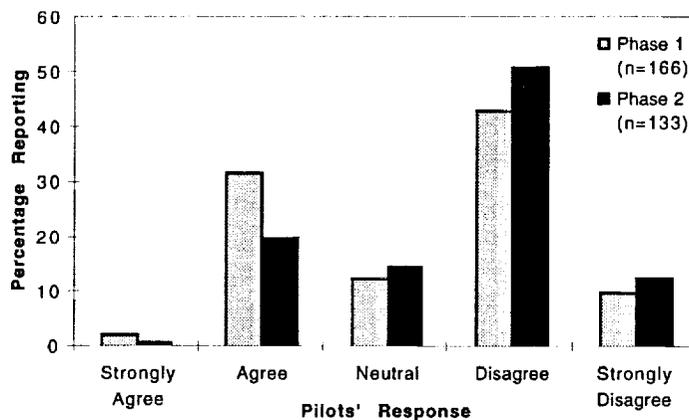
Figure 1.7:   There are still modes and features of the B757 FMS that I don't understand
              (adapted from Weiner, 1985)

While the distribution of responses changed between Phase 1 and Phase 2 of the study, even after at least a year of experience, pilots were being surprised by the automation. The premise that pilots have an incomplete understanding of the automation is further bolstered by Figure 1.7 which shows that pilots felt that there were modes and features of the FMS which they did not understand.

The results of this survey seem to indicate that it may take more time to train pilot to maximal proficiency in new aircraft. The time necessary for this training and the operational experience required have not been determined by this survey. What is necessary is a longitudinal study looking across pilots of varying operational experience to determine when individuals feel that they have mastered the aircraft.

1.3   Motivation

The primary motivation for this research is to gain insight into the underlying causal basis for the human factors issues which have been identified as appearing in new autoflight systems. By understanding the reasons for these issues, mitigation approaches can be identified and suggested. Flight automation systems in successive generations of aircraft have been gaining in capability. This growth in capability has increased the size, and likely the complexity, of new generations of automation. This growth may result in future generations of aircraft which are more susceptible to interface and autoflight systems problems. It is expected that the issues which are identified

through this analysis are early indicators—the leading edge—of these automation problems in future systems. As such, the intention of this work is to serve as a preemptive mechanism to forestall this increase by providing guidance for the design of future generations.

1.3.1 Aircraft Automation as a Leading Indicator to Issues in Other Fields

Aircraft autoflight systems are an effective area in which to study the human factors issues. They can serve as a leading indicator with automation interaction and complexity management and problems in other domains. A number of additional fields may be served by the insights suggested in this work. Nuclear power plants and process control plants represent areas in which workers are trained specifically for their task, albeit not as rigorously as in aerospace. These plants are highly automated in a manner similar to aircraft and in many cases have warning systems and alerts to maximize safety. Air traffic control may also be served by these insights since controllers are highly skilled individuals whose task involves them interacting with automation.

Leading Indicator

Aircraft automation is thought to be an exploratory case study for the identification and consideration of issues which may be important in many other human-automation systems. Pilots are a homogeneously trained group of subjects to investigate, so that fundamental human-automation issues can be seen with fewer confounding factors. The medical and currency requirements on pilots are also stringent. The automation with which pilots interact has rigorous performance requirements due to its life-critical nature.

As a population, commercial airline pilots are homogeneous, intelligent, and highly trained. In the United States, the Federal Aviation Administration verifies that pilots are free of medical disorders, and meet specified educational standards. In addition to initial flight training and check out, commercial pilots are subject to yearly reviews, checkrides, and medical examinations (Federal Aviation Administration 1998). These stipulations are imposed by governmental certification organizations to ensure that the safety of the flying public is not jeopardized. Individual airlines attempt to verify that their pilots do not suffer from drug or alcohol abuse problems. Training is completed and documented on a recurring basis by the airlines. Medical

logs and histories are maintained for each pilot. Stiff competition tends to limit the population to those who are well educated. In general, pilots are held to high standards medically, cognitively, and from the standpoint of co-ordination. Problems in this highly trained and motivated population are likely to indicate problems with other automation areas.

Other Fields

The medical arena is another in which highly skilled and trained practitioners work within a proceduralized environment. In recent years, the drive towards managed medical care has resulted in the adoption of additional proceduralization in order to standardize the care provided to patients. Automation which is used in this field must be designed in a manner which is consistent with a care-giver's model of the task, can be used by a task- rather than technology-oriented audience, and must be able to exist within the procedural environment.

Another important class of operators is composed of those who are not highly trained to specific automation or to the task. Luxury automobiles are an area in which rapid innovation is leading to the adoption of some very interesting automation. Antilock brake systems place a microprocessor between the pedal and the actuator for the brake shoe; one could argue that modern systems provide "brake-by-wire" capabilities. There are other advanced being planned (Port, 1998). "Road-following" is an advanced form of cruise control which allows the vehicle to follow curves and maintain spacing within its lane; other systems can automate lane changes.

1.4   Thesis Argument Overview

The goal of this thesis is to gain insight into the underlying basis for the source of human factors problems which are appearing in commercial autoflight systems and to consider approaches for dealing with these problems. The next chapter will examine the incremental development of these systems, which have been evolving for the past thirty years. Detailed analyses of available aircraft operators manuals, inferred autoflight system behaviour, and accident reports have enabled insight into the structure of autoflight systems and how to segment and extend their behaviour.

29

The results of autoflight system analysis are used to develop the Hybrid Automation Representation, an abstraction of the autoflight system which captures both the continuous and discrete behaviour of these systems. This abstraction bears strong resemblance to modern hybrid models being researched in the control area. The Hybrid Automation Representation will be compared to other similar modeling efforts underway. One of the strengths of this model is that it can be used to measure the "cyclomatic complexity" of autoflight system behaviour. Cyclomatic complexity is a rigorous measure from theoretical computer science of the number of linearly independent paths through a system, which has been extended to allow the measure of an aspect of automation complexity, namely the number of independent paths through which a transition can occur.

Chapter 4 will present a survey designed to validate the applicability of cyclomatic complexity by examining the cyclomatic complexity of autoflight system mode transitions of those transitions identified as difficult by pilots. A survey was conducted on the World Wide Web and made accessible via the Internet. Pilots were instructed to detail autoflight system mode transitions which they found to be most complicated. A subset of these transitions was then analyzed in order to characterize their cyclomatic complexity.

Chapter 5 will use hybrid automation representation to examine several types of underlying causal factors that have been identified through focused interviews and accident/incident reports. The use of the Hybrid Automation Representation in identifying some types of the accidents and incidents *a priori* will be discussed. One of the results of the pilot survey was the identification of the Altitude Capture mode transition as problematic. A case study will be presented of Altitude Capture behaviour in which problems are identified via the hybrid automation representation and were then verified through high fidelity simulator testing.

Chapter 6 will discuss mitigation techniques which have been identified to address aircraft automation problems. The first is the use of procedures and changes in training as a means to mitigate some automation problems resulting from complexity. Directed additional training may be useful to allow pilots to build more robust mental representations of automation. Procedures can be used to "work around" automation issues. The second approach is to enhance feedback to change the nature of interaction with automation and allow more accurate mental representations

when the existing displays are not sufficient or appropriate. Adding feedback in the aircraft may allow a more accurate representation of automation state to be determined. The third approach is to explicitly manage the complexity of the system so that it is more consistent with human capabilities and limitations by modifying the process by which these systems are designed. An Operator Directed Process will be presented as a development process which considers the human pilot's limitations and capabilities early in the design process.

Conclusions and recommendations will be presented in Chapter 7.

Chapter 2

Evolution of Autoflight System Complexity

Autoflight systems have developed in an evolutionary manner over the past fifty years. During this time, the complexity of these systems has increased as they have been made capable of performing additional functions, enabled by advances in sensors, computational capability, and new algorithms. This complexity is hypothesized to be a contributory factor in aircraft incidents, as suggested in Section 1.1. The evolutionary growth of these systems is examined for a particular family of aircraft to investigate the manner, and order, in which functions were added. Other factors, such as the size of software in the autoflight system, and the number of controls with which the pilot has to interact are discussed. The material for this analysis is based on public information sources, such as aircraft manuals, focused interviews with pilots and airline check pilots.

2.1   Modes and Transitions within Autoflight Systems

Autoflight systems have developed incrementally based on the adoption of new technologies. One way to track the incremental growth is to examine the number of independent, quasi-steady-state behaviours available to pilots as documented in the flight operations manuals. Control block diagrams are an effective representation of closed loop control, where the system is typically controlled to a target value. However, each controller is limited to a single behaviour. In order to allow the multiple behaviours necessary during various stages of flight, autoflight systems include multiple controllers for each flight domain, only one of which is active at any time.

Modes are a mechanism to allow disparate behaviours to coexist within a single system. Disparate behaviours will appear when new functionality is added to system which cannot be parsed as an extension to existing function, or cannot be constructed by combining existing functions. In the case of autoflight systems, new modes were needed when necessary behaviours could not be generated by existing closed loop controllers. New modes were added in the form of new controllers. Dividing the system into separate controllers can allow selection among multiple behaviours. The active mode defines the active controller to determine the behaviour of the

33

system. Figure 2.1 shows this abstraction of the autoflight system graphically. Modes allow the autoflight system to be decomposed into separate behaviours which can be selected. Only one controller can be active at a time.
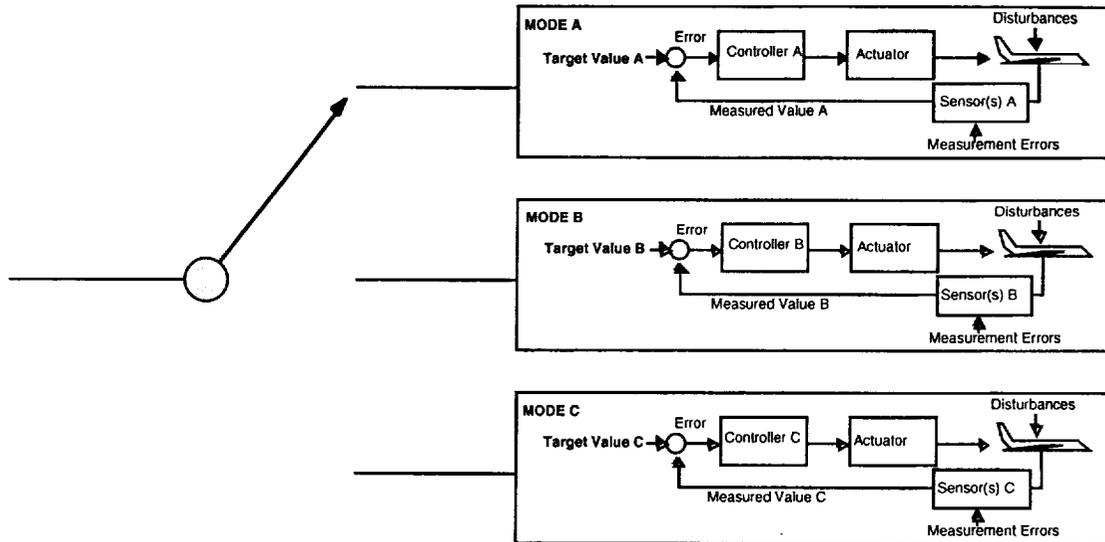
Figure 2.1:   Abstraction of Autoflight System

Modes also allow the incremental adoption of the functions and behaviours into pre-existing systems. They also allow a single system to be more capable by allowing it a larger set of behaviours to deal with more environments, situations, procedures, and scenarios. Systems which have evolved incrementally are more likely to require to take advantages of modes as a tool to manage complexity, as seen in the previous section where new autoflight systems tended to carry forward the majority of older modes. Incremental evolution is often characterized by the adoption of new functions. If new functions need to be added which are not an extension to existing function, or cannot be constructed by combining existing functions, a new mode must be created to encapsulate this new behaviour.

In aircraft, a single mode is unable to allow all the various behaviours required in flight. As an example, consider the recent additions to vertical aircraft automation. A Vertical Speed (V/S) mode has been available since the B727, shown in Figure 2.5. This mode controls the vertical rate of the aircraft, usually by referencing barometric pressure. As such, Vertical Speed is an "air-referenced" control mode. An alternate vertical control strategy is to control to a particular flight

34

path angle, which is fundamentally different in that it is an ground-referenced mode because it measures the angle between the flight vector and the ground. In order to add this capability, a new mode needed to added; more recent aircraft also have a Flight Path Angle (FPA) mode, seen in Figure 2.9.

Modal Structure of Autoflight Systems

Aircraft automation has been parsed to consist of an irreducible set of *base modes* which are used in quasi-steady-state conditions, have an invariant set of targets, and correspond to an unambiguously defined automation behaviour. This definition is consistent with the manner in which pilots model the system, and how engineers model the system—each base mode corresponds to single controller, generally a state level controller (Vakil 1996). A *macro mode* consists of a specific sequence of base modes where a specific order of transitions is expected based on procedural or nominal usage. Each base mode in the macro mode sequence has its own set of targets, so that the automation's set of targets varies over the course of the macro mode. Transitions among the base modes are made based on the mode transition criteria, such as altitude or indicated air speed. An example of a macro mode is the Autoland sequence, which transitions (in the vertical channel) between Altitude Hold, Glide Slope Capture, Flare, and Rollout with a different set of targets in each base mode. Other examples include Vertical Navigation or Profile, Lateral Navigation, Flight Level Change and Autoland.

Transitions Between Modes

Each possible transition between modes consists of a starting mode, an ending mode, *conditional* statements which must be satisfied in order to effect the transition, and the *target value* of the new mode. *Conditions* determine whether the transition will occur. A transition will only occur if all of the conditions are satisfied. Transitions among modes can be caused by various factors, including intervention by the human operator (pressing a button), environmental changes (winds), or due to specific conditions being met (reaching a waypoint or speed limit). Each of these is one element in the set of conditions which must be satisfied before a transition will occur from the starting mode to the ending mode. Individual conditional statements can be combined in a Boolean manner to create more sophisticated interaction. In modern aircraft automation, these statements can become quite complex.

From the standpoint of the pilot, transitions can be further grouped into three categories. A *commanded transition* is active as soon as the selection is made: the transition condition consists solely of the selection itself. An example is moving the Altitude Hold switch to the on position, thereby activating the Altitude Hold behaviour. An *uncommanded transition* is one that is not directly activated by the pilot: the transition conditions consist of elements not under the control of the pilot. These transitions are usually some type of envelope protection, or failure in the automation. Another example is a transition caused by overspeed protection in more modern aircraft. Finally, *armed transitions* occur when, after arming, a mode engagement occurs at some further condition. At least two conditions are necessary: pilot selection and the occurrence of an external condition. An example is the use of Glide Slope Capture to transition to a descent mode after the aircraft intersects with the ILS glide slope signal.

Concerns with Modal Automation

As discussed in Section 1.1, automation problems have been identified as a key safety area. Within this area it appears that modal automation, and mode transitions in particular, are of particular concern. Figure 1.3 shows the that a number of ASRS reports were related to mode transitions. In addition, it appears that pilots have suitable experience with continuous time behaviour of aircraft automation and that is well understood. By contrast, numerous researchers have raised concerns regarding the discrete modal behaviour is more modern aircraft (Sarter 1992, Weiner 1988, Vakil 1996, Javaux 1998, others).

There is also evidence that pilots model the system in a modal manner. If this is found to be a widely adopted representation, it may be the appropriate form upon which to base the training material. Focused interviews showed that pilots have adopted a modal representation of automation behaviour, as described by mode transition diagrams (Javaux, 1999b). Note that this is different from a detailed Finite State Machine type of representation of the underlying automation, but rather an organization of the behaviour into separate modes. The differences appear in the parsing of what constitutes a mode, a trigger event, or a conditional clause. These differences acknowledge the operational viewpoint rather than the design viewpoint. This thesis is going to focus on examining mode transitions in aircraft autoflight automation.

2.2 Autoflight System Evolution

This section will examine the growth in number of modes to support the hypothesis of incremental growth. Based on the open literature and training materials for each aircraft, an estimate was made of the number of independent modes in a series of aircraft as shown in Figure 2.2 (American Airlines 1997, American Airlines 1994, Boeing 1989, Honeywell 1992). These diagrams shows that the number of modes available for use by the pilot has been increasing in a linear manner. The data may be incomplete from the standpoint of system design, but is a measure of the number of modes articulated to pilots. The number of modes may be undercounted. In particular, the high level Airbus PROF and Boeing VNAV modes consist of a set of submodes. These submodes are difficult to directly compare as the manufacturers have parsed the submodes differently. Therefore these modes could not be counted separately. This implies that the mode count in Figure 2.2 is conservative for these aircraft, since modes associated with trajectory control are underrepresented.
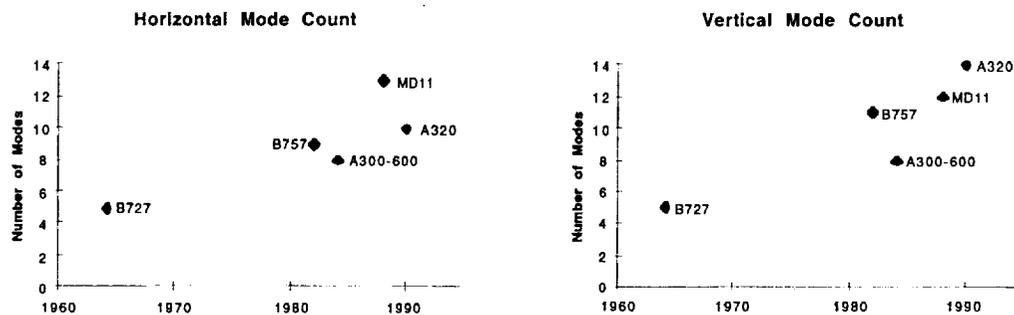
Figure 2.2:   Horizontal and Vertical Mode Counts in Selected Aircraft

The following sections will show the growth of aircraft modes in a more detailed manner organizing modes by the control level at which loop closure is accomplished. Four generations of aircraft will be examined, consisting of the Boeing B727, B737, B757, and B777.

2.2.1 First Generation Automation: B727 (1964)

The Boeing B727 is the representative aircraft for the discussion of first generation of transport jet automation (American 1997). It had limited ability to control its lateral and vertical flight path, but did not have autothrottle capability. As shown in Figure 2.3, in this aircraft, the

vertical modes are not coupled to the speed modes. The Turn/Pitch knob was used to control the attitude of the aircraft. Altitude Hold was the available state control mode and Glide Slope Track was used to control the trajectory during approach. Glide Slope Arm maintained the current altitude (in an identical manner to Altitude Hold) until the glide slope was acquired. At that point, the system transitioned to Glide Slope Track.

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
	Turn/Pitch Knob	Altitude Hold	Glide Slope Track	
		Glide Slope Arm		

Figure 2.3:   Vertical Modes in the B727

Lateral modes are shown in Figure 2.4. The Boeing B727 included an automatic "yaw damper," which acted as a stability augmentation system. This device counteracted the Dutch Roll mode to which swept wing aircraft are susceptible. The Turn and Pitch knob was used to control the roll of the aircraft. State control allowed the selection of a heading, maintenance of a heading, and the ability to track a VOR signal. Trajectory control was used during approach to follow a ILS localizer and Glide Slope signal.

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
Yaw Dampers	Turn/Pitch Knob	Heading Select	Localizer	
		Heading Hold		
		VOR Track		

Figure 2.4:   Lateral Modes in the B727

2.2.2   Second Generation Automation: B747 (1973)

The Boeing B747-100/200 will be used as a representative aircraft for the discussion of second generation autoflight systems (Boeing 1985). Figure 2.5 shows the vertical and speed modes in this aircraft. Sections that are shaded existed in the previous generation.

In addition to those in the B727, the B747 had several new modes, some of which were introduced by the inclusion of an autothrottle. The Turbulence mode was added to provide the ability to hold altitude through turbulent weather conditions and functioned in a manner similar to Altitude Hold. IAS Speed used the pitch of the aircraft to maintain a specified airspeed. Vertical Speed allowed descents at a specified rate. The Speed mode controlled to a target velocity by

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
	Turn/Pitch Knob	Altitude Hold	Glide Slope Track	
		Glide Slope Arm	Perf. Management Sys.	
		Turbulence		
		IAS Speed (pitch)		
		Vertical Speed		
		Speed		
		Altitude Capture/Select		

```
┌──────────────────┐
│Retained from B727│
│   New on B747    │
└──────────────────┘
```

Figure 2.5:   Vertical/Speed Modes in the B747

closing the loop around the throttles and was typically used in conjunction with the Altitude Hold and Vertical Speed modes. Altitude Capture or Select was used to smoothly transition to a level flight path after a climb or descent.

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
Yaw Dampers	Turn/Pitch Knob	Heading Select	Localizer	
		Heading Hold	Perf. Management Sys.	
		VOR Track		

```
┌──────────────────┐
│Retained from B727│
│   New on B747    │
└──────────────────┘
```

Figure 2.6:   Lateral Modes in the B747

The more interesting new mode was the addition, on certain B747s, of the "Performance Management System" (PMS) which provided trajectory control during the cruise segment of flight. This is in contrast to the Glide Slope Track and Localizer modes which was only available during approach. PMS was an early version of Area Navigation (RNAV). Synthesized information from multiple ground-based navigation aids was fused with onboard Inertial Navigation Systems (INS). This enabled the aircraft automation to know its location laterally and vertically at any point in time to a much higher degree of accuracy than previously possible. The RNAV capability increased the number of functions it was possible to have the aircraft perform, including enabling it to automatically fly between waypoints defined laterally and vertically.

2.2.3   Third Generation Aircraft: B757 (1983)

The third generation of jet transport aircraft incorporated multiple radical changes from previous generations, many driven by a Presidential Task Force which allowed widebody aircraft to be flown by two person crews. In order to reduce the workload on the smaller crew, airline manufacturers automated more aircraft systems and graphical displays were used rather than analogue dials to allow more rapid retrieval of information. The Boeing 757/767 was the first of

39

the breed of "glass cockpit" aircraft and will be used as the example in this section (Boeing, 1988). This cockpit design is also very similar to those used in successive Boeing aircraft.

Figure 2.7 shows the Vertical/Speed modes in the B757. Shaded modes are those from previous generations, and those crossed out are modes which were not carried forward to the third generation.

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
	Turn/Pitch Knob	Altitude Hold	Glide Slope Track	Stall protection
	Control Wheel Steering	Glide Slope Arm	Perf. Management Sys.	Overspeed protection
		Turbulence	VNAV Path	
		IAS Speed (pitch)	VNAV Speed	
		Vertical Speed	VNAV Altitude	
		Speed		
		Altitude Capture/Select		
		EPR		
		Takeoff		
		Flare		Retained from B727
		Go Around		Retained from B747
		Flight Level Change		New on B757
		Thrust Hold		Removed

Figure 2.7:   Vertical/Speed Modes in the B757

The B757 exchanged the Turn/Pitch knob for Control Wheel Steering, which can allow direct control over aircraft attitude. It is hypothesized that the Turbulence mode was removed because its function could be handled by a more capable Altitude Hold mode. IAS Speed was replaced with a more capable Flight Level Change mode, designed to allow efficient climbs and descents. Additional modes, such as Thrust Hold, were added to allow more complete control over the autothrottle. Engine Pressure Ratio (EPR) mode was used to allow fuel efficient flight. Takeoff and Go Around mode were used to control the thrust setting to predefined levels during critical flight segments. The Flare mode was used during autoland maneuvers. The PMS was replaced with a more capable Vertical Navigation mode consisting of Path, Speed, and Altitude submodes. Automatic envelope protection also appeared in this generation. Stall protection automatically added power when approached a stall condition. Overspeed conditions were dealt with by reducing the throttle to an idle setting and controlling the aircraft to a maximum safe airspeed.

In the lateral domain, additional state level modes were added to assist during critical maneuver near departure and approach. Rollout was added to assist in post-touchdown

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
Yaw Dampers	~~Turn/Pitch Knob~~	Heading Select	Localizer	
	Control Wheel Steering	Heading Hold	~~Perf Management Sys~~	
		VOR Track	LNAV	
		Rollout		
		Takeoff		
		Go Around		

Retained from B727
Retained from B747
New on B757
~~Removed~~

Figure 2.8:   Lateral Modes in the B757

maneuvers. A Lateral Navigation (LNAV) mode replaced the PMS system in the B747. The LNAV system was also augmented through the use of a "moving map" display which showed the aircraft position graphically in the context of ground based navigations aids.

2.2.4   Fourth Generation Aircraft: B777 (1995)

The most modern generation of aircraft represent the fourth generation of automation and have been introduced since 1988, starting with the A320 (Honeywell, 1992), and continuing with the B777 (Boeing 1997). These aircraft differ from previous generations in that they are Fly-By-Wire rather than cable-actuated: control signals are carried via electrical impulses rather than over mechanical or hydraulic linkages. In practice the distinction to the pilot can be made to be minimal, but a fundamental change is that the inputs from the pilot are now always processed by a computer before the actuation occurs. Automation has become a necessity to fly these aircraft. This capability has been enabled by the adoption of much higher bandwidth digital buses and by placing additional computational power into the aircraft. The latter enables signal processing to occur fast enough to allow interaction with low level flight control.

In the Vertical/Speed domain, Figure 2.9 shows that few additional modes have been added. Shaded modes are those from previous generations, and those crossed out are modes which were not carried forward to the fourth generation. Fly-by-wire (FBW) is used for stability augmentation and serves to interpret any manual control from the pilot. Flight Path Angle mode is used to fly a ground-referenced descent path. This is in contrast to Vertical Speed, which flies an air-referenced descent.

The lateral modes have been augmented by the FBW system as well. A new attitude control mode, ATT: Hold Engage, is used to maintain a roll immediately upon engaging the autopilot. Track Select and Track Hold are the ground-referenced equivalents to Heading Select and

41

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
FBW System	Turn/Pitch Knob	Altitude Hold	Glide Slope Track	Stall protection
	Control Wheel Steering	Glide Slope Arm	Perf. Management Sys.	Overspeed protection
		Turbulence	VNAV Path	
		IAS Speed (pitch)	VNAV Speed	
		Vertical Speed	VNAV Altitude	
		Speed		
		Altitude Capture/Select		
		EPR		
		Takeoff		
		Flare		
		Go Around		
		Flight Level Change		
		Thrust Hold		
		Flight Path Angle		

Retained from B727
Retained from B747
Retained from B757
New on B777
Removed

Figure 2.9:   Vertical/Speed Modes in the B777

Heading Hold. Finally, Envelope Protection has been extended to the lateral domain. AutoBank Limiting limits the bank angle during aggressive high altitude turns to prevent loss of altitude.

Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
Yaw Dampers	Turn/Pitch Knob	Heading Select	Localizer	Auto Bank Limiting
FBW System	Control Wheel Steering	Heading Hold	Perf. Management Sys.	
	ATT Hold Engage	VOR Track	LNAV	
		Rollout		
		Takeoff		
		Go Around		
		Track Select		
		Track Hold		

Retained from B727
Retained from B747
Retained from B757
New on B777
Removed

Figure 2.10:   Lateral Modes in the B777

2.2.5   Technical Factors in Evolutionary Growth

Multiple technical factors have contributed to the evolutionary growth of aircraft. Advances in control theory have resulted in a capability for more optimal control. Advances in multivariable control have generated systems capable of smooth transitions utilizing blended control. Servomechanism work has created control surfaces with better response, especially when coupled with the switch from hydraulic to electrical actuation. Increases in computing power and memory densities have enabled more complex flight paths to be calculated and flown and, when used in conjunction with advanced display technology, have created moving map displays to increase situational awareness. The move from analogue to digital flight controls has allow a large number of changes, both in physical signal transmission, and in the interpretation of pilot inputs. For

example, the A320 utilizes its fly-by-wire system to allow the aircraft to respond in a consistent manner while it moves through its flight envelope.

Perhaps the most significant change, from the standpoint of number of modes available in the aircraft, is the transition between beacon- and area-based navigation. A notional diagram of beacon-based navigation is shown in Figure 2.11. Physically fixed waypoints, shown by the VORTAC symbol are used to define the available paths of the aircraft. Typically, the radio receivers in the aircraft were only capable of tuning into a single beacon. The single target of these systems was a frequency and, perhaps, navigational radial to track. Once the aircraft crossed a waypoint, the aircraft crew had to tune to a new frequency to track a new waypoint. This meant that pilots were responsible for managing the trajectory of the aircraft.
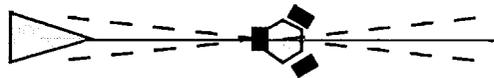
Figure 2.11:   Notional Diagram of Beacon-based Navigation

Area navigation significantly increased the number of types of targets to which autoflight systems could be controlled and created a much richer set of conditions upon which transitions could occur. Synthesized information from multiple ground-based or satellite navigation aids was fused with onboard Inertial Navigation Systems (INS). This enabled the aircraft automation to know its location laterally and vertically at any point in time to a much higher degree of accuracy than previously possible. The RNAV capability increased the number of functions it was possible to have the aircraft perform, including enabling it to automatically fly between waypoints defined laterally and vertically.

Using the RNAV information, targets could be selected from a much broader set. The aircraft behaviour could be made contingent on this much larger set of external elements. As a result of this new sensing capability, the automation was capable of control during the entire trajectory of the aircraft. The automation was also capable of automatically transitioning between modes. These transitions were initiated based on specific condition criteria, such as an altitude, speed or location measured by the sophisticated INS and RNAV systems.
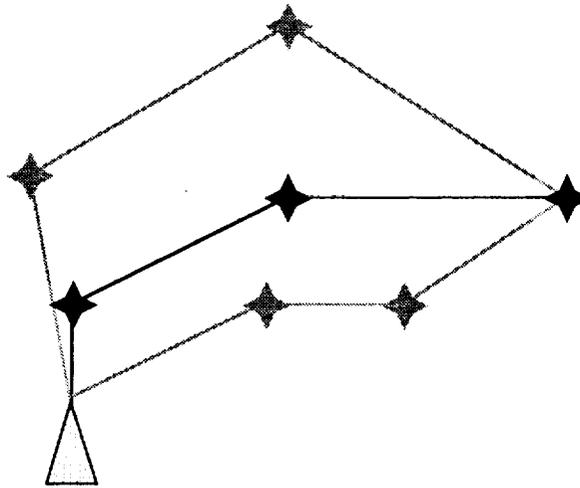
Figure 2.12:   Notional Diagram of Area Navigation

Allowing the automation to make transitions between modes enabled higher level behaviour from the system. As an example, without this capability, an aircraft in a climb mode would have to be monitored until a target altitude was attained. If the system automatically transitions, the pilot can engage the climb mode and enter both a target vertical rate and armed altitude. When this altitude is attained, the aircraft will level off and transition to an Altitude Hold mode, using the altitude as a target. Lateral navigation is similarly extended, since sequences of lateral waypoints can be used to generate successive heading targets, allowing flight along a predefined flight path. This can be very useful, especially if the pilot is able to enter such a flight path during low workload situations.

2.3   Growth in Complexity

The complexity of autoflight systems has been cited as a concern by multiple researchers (Sarter 1992, Hutchins 1996, Degani 1994, others). The term "complexity" has proven to be difficult to define; it is not clear which measurable elements of the autoflight system are appropriate to use as a metric of complexity. However, multiple measures of the size of autoflight systems are consistent in demonstrating rapid growth in the number of controls, displays, and computer software.

2.3.1 Measures of Complexity

Though it is not obvious what metrics are appropriate to measure the complexity of these systems, three metrics are presented in this section. The number of controls and switches is an appropriate measure of the complexity as measured by the human to computer interaction. The complement to this measure is the number of displays in the cockpit, providing computer to human interaction. The size of the software aboard modern commercial transports is presented as an indication of the size of the underlying automation which must be supervised by the pilot.
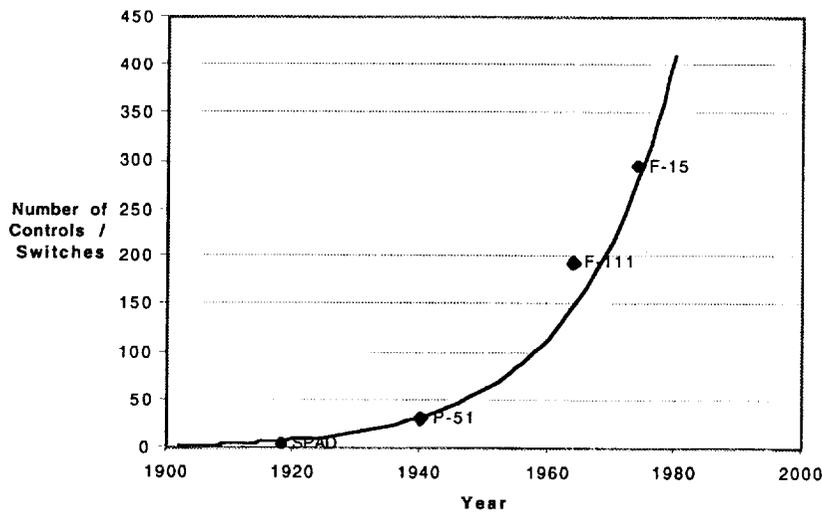
Figure 2.13:   Growth of Controls/Switches in Cockpits (Ostgaard, 1981)

Displays and Controls in Cockpit

The number of controls and switches needed in a cockpit provide some indication of the growth of complexity in aircraft, since the number of functions which can be handled autonomously is linked to the number of controls and displays necessary. Figure 2.13 (Ostgaard, 1981) shows the count of the number of control and switches in some representative military aircraft. The solid trend line indicates the development rate where the number of controls and switches double every 11 years. While not an exact fit to the small set of datapoints, the number of controls is growing quickly in the military domain.

45

Figure 2.14 (Weiner, 1988) shows a similar count of the number of displays in the cockpit. What is interesting in this graph is that the number of displays increases and subsequently decreases. In addition, the rate of increase is notably slower than that shown in Figure 2.13. The diagram shows that the maximum number of displays was reached in about the early 1970s and is now decreasing.
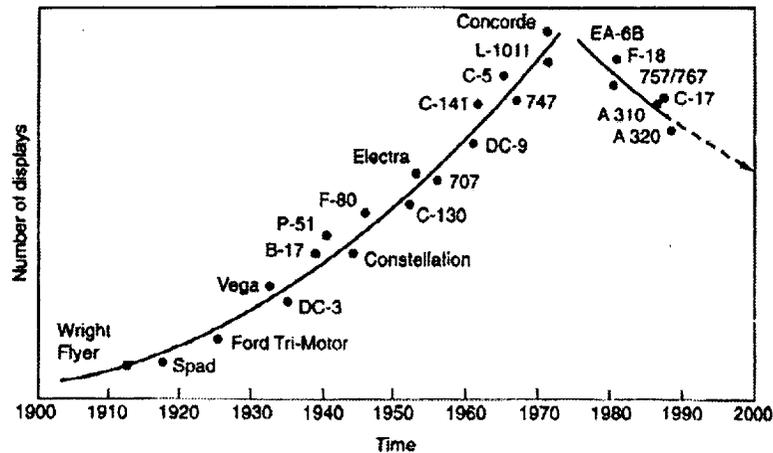
Figure 2.14:   Growth of Displays in Cockpits (Weiner, 1988)

It appears that simply measuring the number of displays is inadequate. Newer aircraft have multifunction interfaces which display multiple pages. While the pages do not take up additional space in the cockpit, they nonetheless increase the number of "displays," as defined by the information shown to the pilot rather than by their physical attributes. Accessing this information may, in fact, be more difficult, since appropriate data may require additional effort to view.

An alternate approach is to show more elements of information on a single page or display, resulting in the use of multiple symbols. Primary Flight Displays (PFDs) have multiple indications on the speed tape to indicate specific limits and targets for the aircraft. Figure 2.15 shows a partial list of the possible tags which can appear on a modern PFD. In order to utilize these additional pieces of information, pilots must be trained to interpret them correctly. These multi-function displays, which underlie the decrease in the number of physical displays, were driven by the need to more effectively utilize the limited space in the cockpit.

46
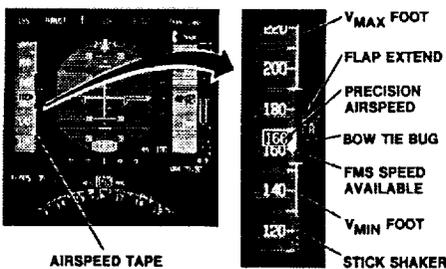
Figure 2.15:   PFD Airspeed/Mach Display (Honeywell, 1992)

Size of Software

In order to support the development of additional functions and displays within the aircraft, the software systems underlying the automation have grown rapidly. One of the indications used by industry to ascertain the size of a project is to estimate the software lines of code (SLOC) required. This metric is somewhat suspect since lines of code do not translate easily between computer languages. Another metric is to look at the actual machine instructions which are generated. This has a similar weakness in that the size may be dependent on the particular computational architecture. However, the trends which appear in Figure 2.16 are based on a single manufacturer's data and are more amenable to comparison (Weener, 1998).

As can be seen, avionics software is growing at an exponential rate. The left graph shows a straight exponential extrapolation of growth in object code, with the size of code doubling every 1.5 years. It is important to note that the rate of growth is heavily influenced by the final datapoint of the B777-200, which is a fly-by-wire aircraft (FBW) in contrast with the other aircraft, which are cable-actuated. To account for this difference, the graph on the left shows two extrapolations based on the hypothesis that the sudden increase in the size of the software reflects the influence of the shift to a FBW system. The solid line is an extrapolation based on the cable-actuated aircraft and doubles at a rate of 2.7 years. Based on the empirical data, the dotted line is the cable-actuated curve translated upwards 42 MB. While this curve increases more gradually, it still predicts that the next generation aircraft will have well over 100 MB of object code.

A corollary of this hypothesis is that the avionics systems in aircraft are presenting the pilot with some subset of the fullest possible amount of information. In order to reduce the information to a manageable form, software interprets and manipulates the raw data. In any sort of interpretive
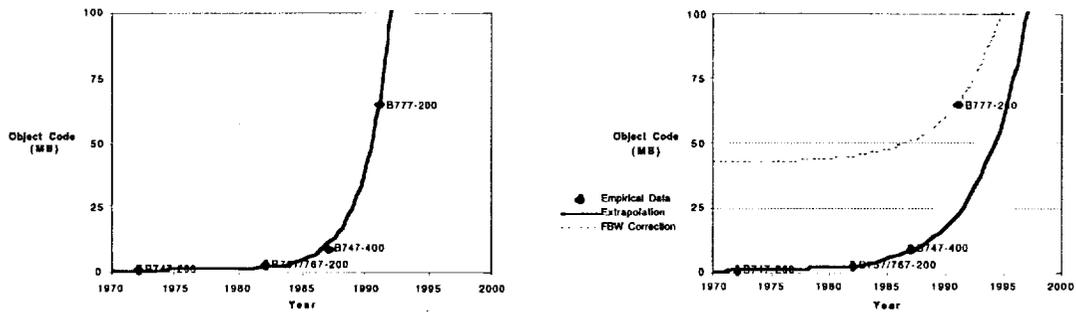
Figure 2.16:   Growth of Software (Weener, 1998)

framework, some information will be hidden from the pilot. This concern of hidden variables and its implications reappears in later in this thesis.

Related to the quantity of software required to run a system are the number of control signals which are distributed by the processing backbone. Figure 2.17 shows the growth of the number of signals in the same group of aircraft. The left graph shows an extrapolation across both FBW and cable-actuated aircraft and has a doubling rate of 2 years, comparable to the increase in software size. The right graph the solid line shows an extrapolation based on the cable-actuated aircraft and then translates this upwards by 7000 signals in order to take into account a single time cost associated with the introduction of the FBW system (Weener, 1998). This curve doubles every 3 years. It likely that the number of signals in a digital system will increase at a faster rate than in a cable-actuated system since the cost associated with each additional digital signal is much lower.
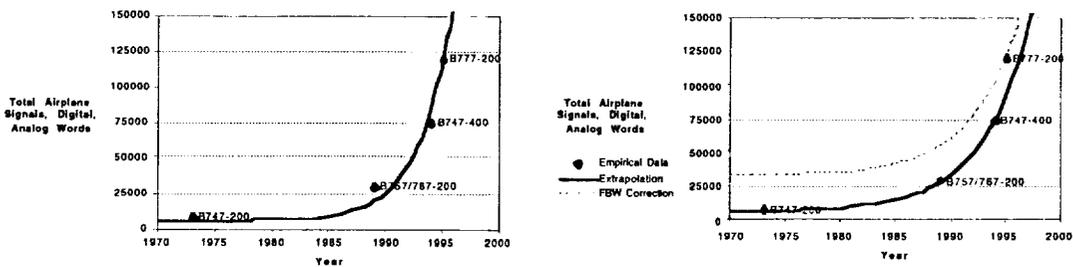
Figure 2.17:   Growth of Signals (Weener, 1998)

48

Finally, it should be noted that this information is based on the small set of available data. While basing an extrapolation on this set is potentially specious, the data does suggest a rapid and exponential rate of growth.

Unconstrained Growth of Software Complexity

An advantage and disadvantage of software is that it is free of many of the physical constraints in design. Conventional mechanical systems are limited in complexity by the necessity to manufacture and maintain the designs. These factors exact a cost from overly complex design. In addition, mechanical systems are constrained by physical attributes. Aircraft must be light enough to fly, chairs must support a weight of 300 lbs, and film must work in standard lighting conditions.

By contrast, software has few physically imposed constraints. Modern processors, software systems, and sensor suites afford a great deal of the flexibility, capability, and the capability for complexity. There is a high cost associated with the initial creation of complex software, and a maintenance cost as the software needs to be upgraded, but the "manufacturing" cost is minimal. The minimal limitations imposed by the physical systems, namely the computing power and memory storage, have been increasing at exponential rates. In the absence of physical constraints, software systems can become excessively complex with little apparent penalty during design. However, the penalty from a complex design may appear during operational use rather than during development.

2.3.2 Apparent Complexity of Autoflight Systems

Aspects of the growth of aircraft autoflight systems are captured in the metrics suggested above. An additional element is the "apparent complexity" of the system: the complexity perceived by the operator of the system. This is hypothesized to be a function of the number of modes in the autoflight systems, the number of transitions among modes, and the nature of transitions among modes: automatic versus manual and whether feedback is provided. These three factors appear to be most critical to the complexity that is apparent to the operator. The following quote from a flight manual demonstrates that the complexity may be a function of the transitions among modes.

"Through the FCU, an immediate climb/descent is initiated by selecting the desired altitude in the ALT SEL window and either pulling the set knob or pressing the LVL/CH P/B to engage the LVL CHANGE mode. Pressing the LVL/CH P/B also disengages PROFILE, however, if PROFILE is engaged, pulling the set knob does not disengage it, rather it initiates an immediate climb/descent to the altitude selected on the FCU. The exceptions are..." (US Airways, 1998)

2.4   Chapter Summary

This chapter has covered an analysis of modern flight automation systems, which consist of two sets of behaviours. The first is the continuous behaviour of autoflight systems which can be represented using control block diagrams. The need for multiple behaviours resulted in independent continuous behaviours appearing incrementally in successive generations of aircraft in an evolutionary manner. The second type of behaviour was discrete switching among continuous behaviours. The need to utilize multiple behaviours drove the adoption of modes as a mechanism to organize disparate behaviours. Mode transition matrices and diagrams were developed as tool with which to analyze the modal (discrete) behaviour of automation. These models are used in the next chapter to develop an automation representation which encompasses these two types of behaviours.

Chapter 3

Hybrid Automation Representation

In order to analyze flight automation, a Hybrid Automation Representation was developed based on the publicly available documentation of current autoflight systems. The lack of a consistent model explaining the behaviour of aircraft automation resulted in a "hybrid" model being used to capture the detailed analysis of these systems.

Based on an analysis of numerous aircraft (Boeing MD-11, B727, B737, B757, and B777, the Airbus A300-600, A310 and 320), autoflight systems appear to composed from two fundamentally different types of behaviour. The first is a "quasi-steady-state" behaviour where the automation controls the aircraft towards some target state in a continuous manner. This behaviour can be completely modelled using control block diagrams at various levels of loop closure. Each additional quasi-steady-state behaviour required an additional controller, modelled by an addition control block diagram. Therefore, in order to support additional functionality additional different controllers or target states were required, though each behaviour could still be modelled by a single control block diagram. It became necessary to segment the automation to organize aircraft capabilities by allowing selection among the active control loops. Each quasi-steady-state behaviour is commonly termed a "mode."

The second type of behaviour requires a set of analytical tools to understand this discrete, "modal" structure. A specific mode is defined by the target that has been set and the manner in which the targets are to be acquired. Where control block diagrams are an effective representation of a single mode, it was necessary to describe discrete transitions among the modes. These transitions are typically initiated by events or when particular conditions are satisfied. The discrete nature of these transitions makes them difficult to model within the continuous representation of the control block diagram. Differing representations, mode transition diagrams and matrices are used in the following sections to represent this level of behaviour. These diagrams also provide the basis for measuring the complexity of these systems.

This hybrid organization of the aircraft automation is similar to hybrid control models being used in modern control theory. In order to allow a system to respond to a wider set of situations, hybrid control systems utilize a set of continuous, but independent, control mechanisms. This provides more flexibility than using a single control mechanism for all possible situations since the appropriate control mechanism can be selected for each situation faced by the system. Hybrid control systems are being researched for use in applications ranging from autonomous vehicle control to process plant control (Godbole, date unknown).

The Hybrid Automation Representation which was developed integrates the continuous representation of control block diagrams with mode transition matrices and diagrams (Vakil, 1998) for discrete representation. One of the goals of this representation was to create a mechanism to evaluate the complexity of flight automation systems *a priori*, based on the underlying structure of the automation. Figure 1.3 implies that a significant number of issues in autoflight systems are related to transitions among modes. Cyclomatic complexity, a measure of the number of linearly independent paths through a system, is presented as a means to characterize the complexity of automation. This measure is directly applicable to the mode transition diagram representation presented.

3.1   Analysis of Quasi-Steady-State Behaviour

Control block diagrams are a common representation of control loop mechanisms used by automation designers and engineers. They are a useful representation of continuous processes which consist of target values, mechanisms to measure the actual system state, and generated error values. Control loops generally drive a system towards the target based on the difference between the current state of the system and the desired target in the presence of disturbances as well as other forces. As such, they are effective at capturing the quasi-steady-state behavior of modes of automation, where specific target values are being attained (Vande Vegte, 1990).

Figure 3.1 shows a simple example of a control block diagram. On the left is shown the set target value. This value is compared to the actual, measured value, as detected by some sensor. The difference between these two is the error measurement and is used to generate a signal to the actuator. This actuator then physically changes the state of the aircraft.
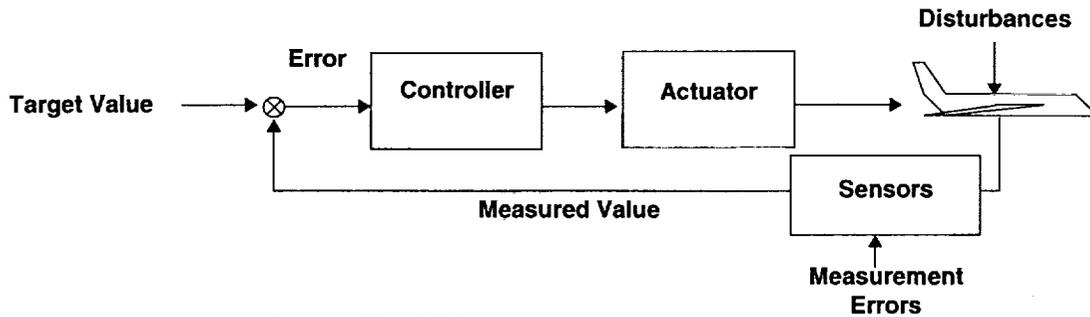
Figure 3.1:   Generic Control Block Diagram

Control block diagrams are a means by which the feedback controllers used in automation are designed and documented. During design, the wealth of previous work done using control block diagrams, especially with linearized processes, can be brought to bear. Using this knowledge, the attributes of the closed system can be determined, in terms of how quickly acquisition will occur, how large the overshoot will be and other characteristics. Since these diagrams are created during system design they are already available and provide a consistent and comprehensible means to depict this type of behaviour to the pilot.

To convey the behaviour of automation to the pilot, completely specifying a control block diagram, in terms of gains, integrations, and control algorithms is unlikely to be necessary. The most important elements to convey to the pilot are the type of target which the controller is using (vertical speed, altitude etc.) and the value of the commanded target, such as the specific vertical speed value. Both of these elements are important in order for a pilot to understand what the system is doing. In the analysis framework being suggested, these two pieces of feedback are associated with the control block diagram representation of the continuous behaviour of the automation.

3.1.1   Attitude Control Loops

Figure 3.2 shows a highly simplified version of the attitude control loop for the roll axis of the aircraft. Similar control loops exist in pitch, yaw, and thrust. In each case a target is specified by some external source. The actual measurement of each axis is determined via gyroscopes, in the case of roll and pitch, or via a more complicated indirect measure in the case of thrust. The difference between these values is used to generate a signal to drive the actuators: ailerons,

elevators, rudder, and throttle. Unlike other axes, which have a variable target value, yaw is always driven to a target value of zero via the yaw damper.
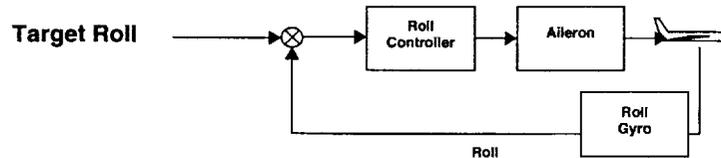
Figure 3.2:   Simplified Roll Attitude Control

In actual systems, maintaining a specified roll, or bank angle, can be much more complicated, as it is dependent on the sensors which are available. Figure 3.3 shows a more realistic control block diagram. In this case, the response of the bank sensor is too slow, and so a roll rate gyroscope must be used for control, requiring the error to be converted into a commanded roll rate actuated by the ailerons. The bank sensor is used for reference, but not used as the primary sensor. Even this diagram is simplified: in the presence of coupling between multiple axes of flight, such as roll-yaw coupling, the commands to the actuators will be include terms from these other axes. As an example, there can be crossover coupling of a bank angle hold autopilot and the yaw damper.
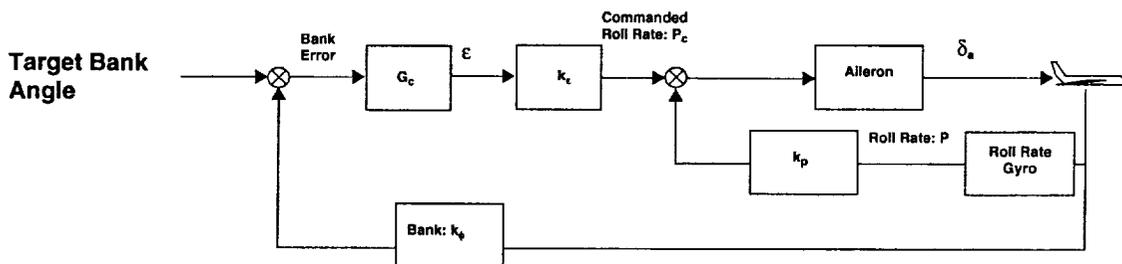
Figure 3.3:   Bank Angle Hold Autopilot (McRuer, 1973)

3.1.2   Velocity Vector Control Loops

At the velocity vector level of control, shown in Figure 3.4, rather than controlling the attitude of the aircraft, the automation controls the velocity vector of the aircraft. This is a level of control as it allows the specification of targets which are more closely aligned with directives from air traffic control. The heading, vertical rate, altitude, and speed of the aircraft can be controlled by setting an appropriate target. The velocity vector controller uses the error between the target value

and the actual value to generates a target for the attitude controller designed to zero the error. The attitude controller then controls to the generated attitude target. Both velocity vector and attitude controllers typically correspond to "base modes" in aircraft: controllers which are used in quasi-steady-state conditions, have a single, scalar and invariant target, and correspond to an unambiguously defined automation behaviour. The simplified block diagram for a specific velocity vector level control mode, Vertical Speed, is shown in Figure 3.4.
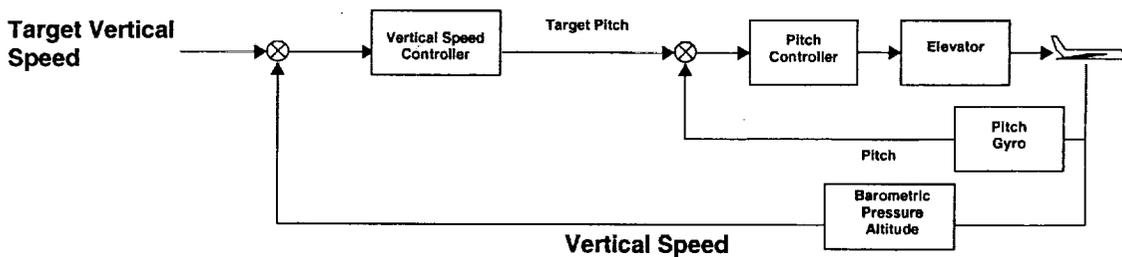
Figure 3.4:   Example of Velocity Vector Control: Vertical Speed

Coupled Modes

Multiple velocity vector controllers can be engaged simultaneously (e.g. vertical path may be controlled by a pitch controller while speed is controlled by throttles and heading by elevators). If multiple controllers are initiated by a single pilot input, these modes are considered "coupled." Modes are coupled when their functions are linked together dynamically or operationally. As an example, the Flight Level Change mode on Boeing aircraft engages both a pitch controller to target the current airspeed and places the throttles into an IDLE or CLIMB setting which enables the fastest possible descent or ascent. Figure 3.5 show the coupled control which occurs when the Flight Level Change mode is selected to climb. This mode engages two controllers. The speed of the aircraft is controlled by the its vertical speed through pitch. The vertical path is controlled by placing the throttles into a "Max Continuous Climb" setting and climbing at the resultant rate.

In general with autoflight systems, if a vertical and lateral mode are coupled, the vertical mode can only be initiated if the lateral mode has already been engaged. As an example, from the Boeing B727, the elevator autopilot switch can only be switched if the aileron switch has already been engaged. Similarly, the Glide Slope Capture mode will not become active until after the aircraft is established on the localizer.
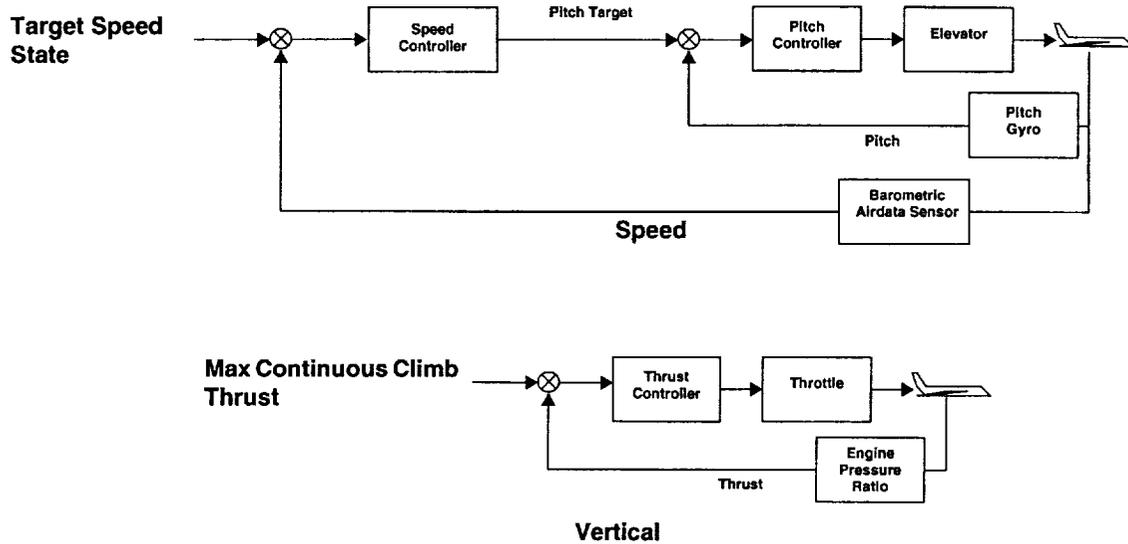
Figure 3.5:   Coupled Control in Flight Level Change Climb

Coupled modes are difficult to capture in this control block diagram representation. Differentiating between a pair of coupled modes and each mode individually involves the manner in which they are initiated rather than their continuous operation. Similarly, the "interlocks" which prevent the engaging of a vertical mode until its lateral counterpart is engaged are difficult to capture within a control block diagram.

Single Input-Single Output Control

In Figure 3.5, the speed controller is shown as a completely independent control loop from the vertical controller. In control terms, this is a Single Input, Single Output (SISO) system, where each output state is controlled by a single input. Typically a pair of thrust and vertical velocity modes engage two independent SISO controllers: the aircraft's pitch controls the vertical speed and the thrust controls the air speed, decoupling the speed and vertical path of the aircraft.

Either the elevators or the thrust can be used to control the vertical path or the speed: reducing thrust while maintaining speed can be used to descend, and pitching the aircraft up while maintaining the vertical path with thrust can be used to reduce speed. For a given mode the control allocation is implicitly selected. Table 3.1 show a set of vertical/speed modes and their associated control allocation for the MD-11. Note that this problem does not exist in the lateral

56

domain, since only a single mechanism, rolling to a new heading, is available to track a lateral target in co-ordinated flight.

Table 3.1:   Representative Vertical and Speed Modes in the MD-11

Vertical and Speed Modes	Speed Allocation	Vertical Allocation
Altitude Hold	Throttle	Elevator
Vertical Speed	Throttle	Elevator
Flight Level Change	Elevator	IDLE or CLIMB Throttle
Glide Slope Tracking	Throttle	Elevator
Go Around	Elevator	CLIMB Throttle

Multiple Input-Multiple Output Control

A more complex system uses multiple controllers with multiple targets by mixing the necessary control signals between multiple actuators. An extension of coupled modes, which can be effectively captured using control block diagrams, are modes which "blend" control across multiple channels. In this case, multiple input and multiple outputs are tied together by the dynamics of the aircraft. Each of the outputs is blended together to control the trajectory. The vertical and speed state of the aircraft correspond to the potential and kinetic energy of the aircraft. In conjunction, both states determine the total energy of the aircraft. With this coupling, speed can be traded for altitude and vice versa, leading to a number of mechanisms to maintain altitude or control the climb/descent rate of the aircraft. This is a Multiple Input-Multiple Output (MIMO) controller, where each output variable is controlled by more than one input. Figure 3.6 shows the MIMO Vertical/Speed control.
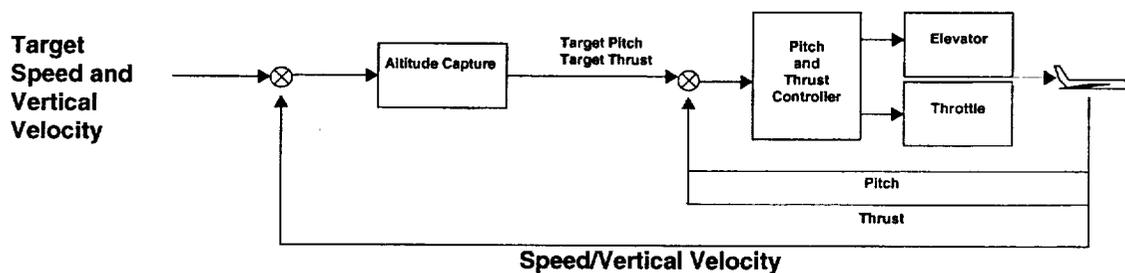
Figure 3.6:   Multiple Input, Multiple Output Velocity Vector Control

Multiple Possible Targets

Velocity vector control is more complex than attitude control because more targets can be commanded. In the vertical path, the target command can be one of many possible types: vertical speed, altitude, defined vertical path (glide slope), pitch, angle of attack, flight path angle and others. Each of these targets defines a different underlying controller, and therefore a different automation mode. Each of these modes must be represented by a separate control block diagram. Table 3.2 shows a selected set of possible modes in the Boeing B737 and the associated controllers.

Table 3.2:   Possible Targets in the Boeing B737

Target	Controller/Mode Selection
Heading	Heading Select
Localizer Signal	Localizer Track
Speed	Speed: IAS or Mach
EPR (Engine Pressure Ratio)	EPR
Glide Slope	Glide Slope Track
Vertical Speed	Vertical Speed
Altitude	Altitude Hold
Altitude	Flight Level Change

Multiple Possible Target Acquisition Means

Velocity vector control consists of multiple modes with which to complete a task. As an example, consider commanding a lower altitude. This target could be attained by reducing the thrust of the engines to decrease speed and therefore lift causing the aircraft to sink or by lowing the nose of the aircraft to descent. Pilots who use velocity vector control must remain aware of the implications of their choice of mode: in this example, the former will not cause the aircraft to gain speed, whereas the later could cause an overspeed condition. Once again, the details of the continuous nature of this mode are captured effectively in control block diagrams, where the actual target can be identified.

3.1.3 Trajectory Control Loops

At the trajectory level, shown in Figure 3.7, the automation controls the trajectory of the aircraft. In the lateral channel, trajectory control functions by measuring the offset from the desired target and generating a signal to correct the heading to reacquire the target trajectory. The measurement can be based on a number of different sensors: an Inertial Navigation System, an Area Navigation System, an en-route navigation aid signal or combinations thereof. In the vertical and speed channels, a similar process has signals generated from a vertical course deviation and thrust profile controlling the vertical speed and airspeed of the vehicle.
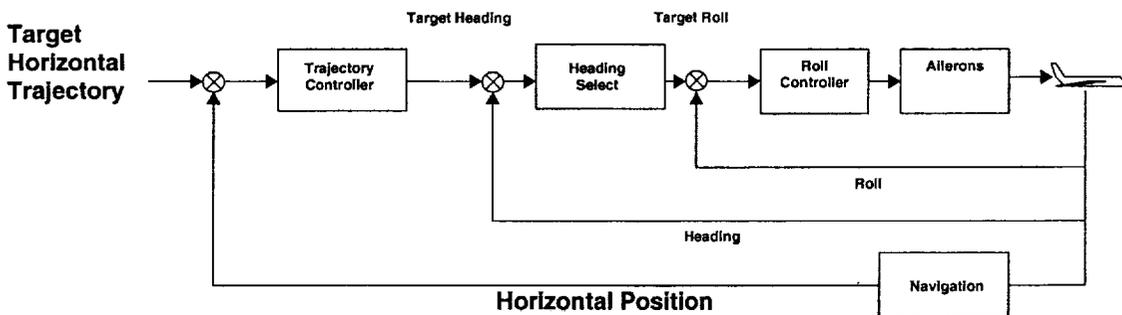
Figure 3.7:   Lateral Trajectory Control

In Figure 3.7, a single lateral trajectory level loop encloses two other inner level loops. To control the aircraft to a trajectory, corrections are made to maintain an appropriate heading to intercept the signal. The heading controller specifies a target to the roll controller which actuates the aileron.

As with velocity vector control, what is less apparent in control block diagrams is that multiple possible controllers are available for each of the flight axes. At any time, only a single trajectory controller is active, implying that the continuous behaviour of the aircraft in one axis can be characterized by a single control block diagram.

3.1.4   Other Control Loops in Modern Aircraft

The newest commercial jet transports incorporate fly-by-wire controls. Rather than pilots or automation providing control signals to the control surfaces via mechanical or hydraulic linkages, the inputs are digitally encoded, transported via a databus, and then decoded at the actuator.

Digitally encoded signals can be transported on much less massive wires than hydraulics or mechanical systems, leading to weight savings.

Digital control also offers the ability to place an intermediary between the inputs from the pilot and the control surface, even during manual control. This intermediary step interprets the inputs of flight controls and then actuates control surfaces. As an example, consider the behaviour of the Airbus A320 near stall. During normal flight, the input from the pilot side stick is interpreted as a standard control law, mimicking the behaviour of conventional aircraft. As the aircraft approaches the high lift region of flight, prior to stall, the input is interpreted in a manner which generates increased positive stability. As shown in Figure 3.8, below a specified speed limit $(1.13V_s)$, the relationship switches from a linear control mode to an angle of attack (α) control mode. In this mode, stick deflection will not correspond to elevator deflection. Instead, elevator deflection will be modulated to prevent stall. The α-control mode has been designed to allow high lift while preventing stall and allowing control authority in other axes. Full stick deflection results in maximal lift, but may not result in full elevator deflection.
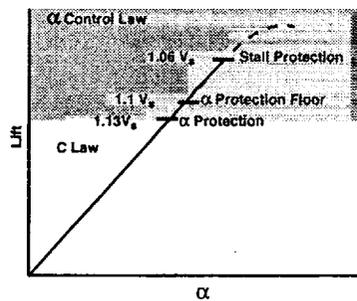
Figure 3.8:   Changing Control Laws

3.1.5   Limitations of Control Block Diagrams

The previous sections have demonstrated how control block diagrams are an effective means to represent a subset of the aircraft automation function. This subset consists of a quasi-steady-state flight segments which are based on a target for each channel (lateral, vertical, and speed) of flight. A single diagram can capture all of the information regarding the continuous behaviour of the aircraft. Multiple diagrams can cover multiple possible behaviours. However, control block diagrams do not effectively represent how aircraft switch between behaviours, deal with modified targets, or respond in the face of performance changes.

3.2　Analysis of Modal Behaviour

Autoflight systems have evolved from systems with few defined behaviours. New systems have a much larger set of modes and large set of associated transitions. This level of the automation cannot be effectively captured because they are fundamentally discrete changes in operation whereas control block diagrams are designed for and largely limited to use in a continuous space. Discrete transitions are necessary to allow aircraft to deal with more scenarios in the flight environment and are necessary to include when describing automation. The next section examines the evolution of autoflight systems as modes have been incrementally added.

Modes are a mechanism to allow disparate behaviours to coexist within a single system. Disparate behaviours will appear when new functionality is added to system which cannot be parsed as an extension to existing function, or cannot be constructed by combining existing functions. In the case of autoflight systems, new modes were needed when necessary behaviours could not be generated by existing closed loop controllers. New modes were added in the form of new controllers. Dividing the system into separate controllers can allow selection between multiple behaviours. The active mode defines the active controller to determine the behaviour of the system.

The behaviour of modes has two large domains in its characterization, its continuous behaviour and its discrete, transitional behaviour. The continuous behaviour of a mode is entirely captured in the control block diagram. The discrete behaviour of a mode is the manner in which it transitions to other modes. The feedback of this discrete behaviour to the pilot also needs to be considered.

A formalism was developed to represent transitions between modes which is based on the formalism of Finite State Machines (FSMs). *Mode transition diagrams* are used to represent discrete elements of modal automation. FSMs are a standard tool used in the field of computer science to describe and design complex systems, including flight automation. Unlike FSMs, which are used by engineers during design and analysis, mode transition diagrams describe the structure of the automation as experienced by the pilot. FSMs consist of a set of states, a set of transitions between states and the criteria which cause transitions to occur. Modal automation

61

systems can be represented using the same notation which is used for Finite State Machines. In this nomenclature, the states correspond to the modes of the automation, transitions move between modes, and the transition criteria consist of the conditions which must be satisfied. The analogous diagrams are termed mode diagrams and the matrices which are derived are called mode transition matrices.

Figure 3.9 shows the modal structure of a simple autopilot. The circles denote the possible states, and the directed arcs represent the possible transitions. There are a total of six modes in this diagram: Horizontal Autopilot Off (HOFF), Localizer Track (LOC), Vertical Autopilot off (VOFF), Heading Track (HDG) Vertical Speed (VS), and Glideslope Track (GS). In this example, HOFF can transition to LOC or HDG, but not to VOFF, GS or VS. VOFF can be transitioned to from VS or GS, and so on.
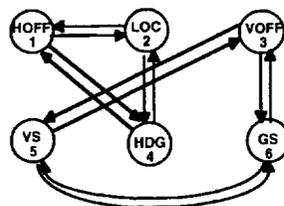
Figure 3.9:   Modal Structure of Simple Autoflight System

The equivalent mode transition matrix is shown in Table 3.3. This is an "allowable" mode transition matrix, where matrix elements correspond to whether a transition is possible or allowed between two different modes. Each row i is the set of transitions which leave mode i. The column j in row i has an entry corresponding to whether a transition exists from mode i to state j. If the mode transition matrix is some matrix T, then T_{ij} is equal to one if a transition exists between states i and j.

Multiple attributes of a modal system can be examined with a mode transition matrix or diagram. The example given above is a straightforward identification of which transitions between modes exist. Alternately, the feedback that was provided to a pilot during a transition could be shown, or the conditions which precipitated the change of mode. In the next sections, the

62

Table 3.3: Transition Matrix for Simple Autopilot System

Mode	HOFF	LOC	VOFF	HDG	VS	GS
HOFF		1		1		
LOC	1			1		
VOFF					1	1
HDG	1	1				
VS			1			1
GS			1		1	

generic mode transition diagram will be refined so as to be more useful in the analysis of autoflight systems.

3.3 Hybrid Automation Representation

As part of this thesis, the Hybrid Automation Representation was developed. This representation attempts to capture both the quasi-steady-state and the discrete behaviour of aircraft automation systems. The representation integrates the continuous representation of control block diagrams with mode transition matrices and diagrams (Vakil, 1998) for discrete representation. Figure 3.10 shows the major elements of this model. To read this diagram, the "From" modes are shown in the rows and then "To" modes are listed in the columns. The transition from Mode A to B is shown to occur when the Elevator Autopilot Lever is in the ON position under the condition that the Aileron Autopilot is also in the ON position. The feedback to the pilot consists of the position of the lever itself.

3.3.1 Detailed Mode Transition Diagrams

The representation of modal automation can be represented in a more detailed form through a diagram focussing on a single mode transition. This representation captures the important elements of transitions in an autoflight system in a more understandable manner. These elements consist of the conditions needed to satisfy a transition, and whether they are commanded or automated, the feedback provided regarding the transition, and the manner in which the target value is specified in the new mode. Figure 3.11 shows an example of a mode transition diagram describing the transition from Mode α to Mode β which shows the major elements of the refined Mode Transition Diagram. Each transition consists of a starting mode, an ending mode, a set of
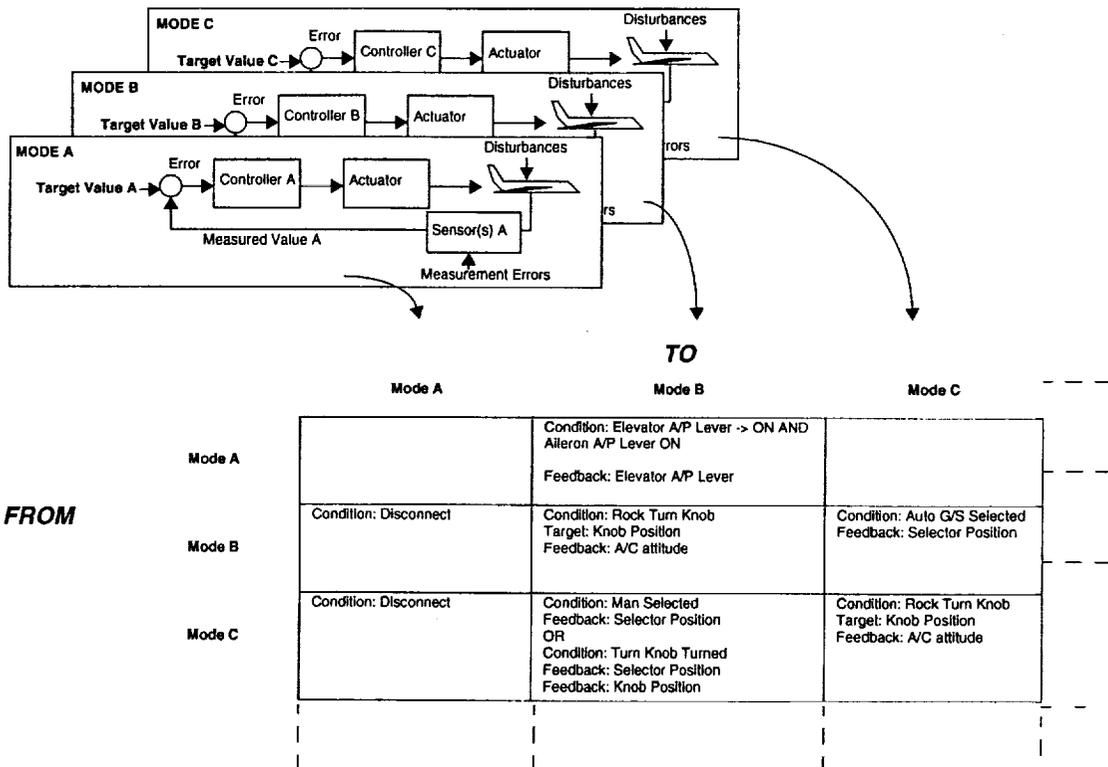
Figure 3.10: Hybrid Automation Representation

conditions to satisfy, the feedback provided during the transition and the new target values for the ending mode. Two types of conditions are shown. The first is a manual or commanded condition, depicted by the pushbutton. The second is the automatic condition depicted by the switch.
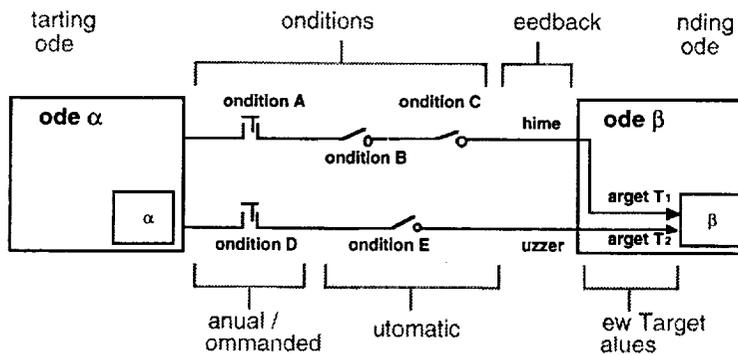
Figure 3.11: Mode Transition Diagram Abstraction

In Figure 3.11 while in Mode α, the target is T_α. One transition path to Mode β is to satisfy Conditions A, B, and C. If these condition are satisfied, a chime will provide feedback, make Mode β the active controller, and specify the target value of that controller to be T_1. Alternately, if manual Condition D is satisfied in addition to automatic Condition E, a buzzer will sound, and Mode β will become active with a target value of T_2.

3.3.2 Example Mode Transition Diagrams

The next several examples show the use of this abstraction to capture elements of the autoflight system. Figure 3.12 shows one transition between Altitude Hold and Vertical Speed, which occurs once the pilot selects the Vertical Speed button on the Mode Control Panel. After this condition is met, the target vertical speed is set to zero feet per minute. Feedback of this change is shown on the Flight Mode Annunciator.
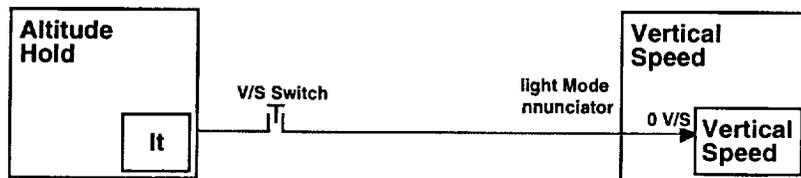
Figure 3.12:   Transition between Altitude Hold and Vertical Speed

Figure 3.13 shows the use of the pitch wheel to change the vertical rate of the aircraft while the Vertical Speed mode is engaged. When the condition of the pitch wheel moving is satisfied, a transition occurs from Vertical Speed back to Vertical Speed with the target being updated to the new value determined by the pitch wheel. The new target is shown on the Mode Control Panel. Note that this example does not change the active mode of the automation, but only the target.
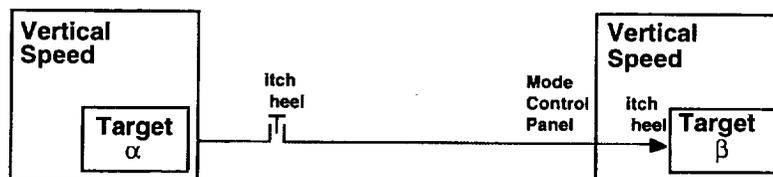
Figure 3.13:   Changing Vertical Speed using Pitch Wheel

Figure 3.14 shows an alternate notation for situations where only the target, and not the mode, change. In this notation, the transition loops back into the same mode, with the value of the new target being specified by the pitch wheel.
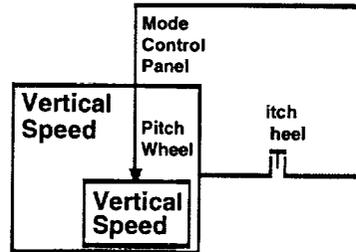
Figure 3.14:   Changing Vertical Speed using Pitch Wheel, Alternate Notation

Figure 3.15 shows an example of an automatic transition to an envelope protection mode from the Vertical Speed mode. If an overspeed condition is satisfied, this will be shown to the pilot on the Flight Mode Annunciator. An automatic transition will occur to the Flight Level Change mode which has both speed and thrust targets. The target speed will be set to Vmax and the target thrust limited to Idle.
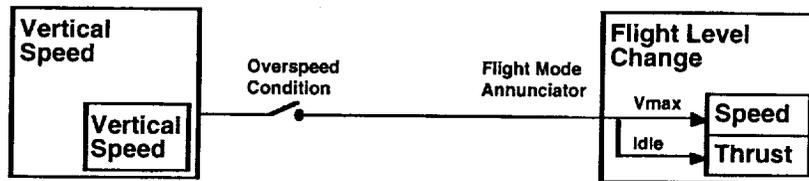
Figure 3.15:   Automatic Transition to Flight Level Change Mode

Each of these previous examples has shown a single transition between modes. Figure 3.16 shows a larger subset of the altitude capture modes from the Boeing B737 (Boeing 1985, 1989). Three modes are shown in this diagram: Vertical Speed, Altitude Capture, and Altitude Hold. Vertical Speed is changed through the use of the Vertical Speed thumbwheel, with the new target vertical speed set based on the thumbwheel position and signal. Other transitions to Vertical Speed can occur from the Altitude Capture mode if the Altitude Selector is moved more than 100 ft with the instantaneous vertical rate of the aircraft being used as the new target value. A change in the Altitude Selector will result in a transition to Vertical Speed, but with a target value of zero.
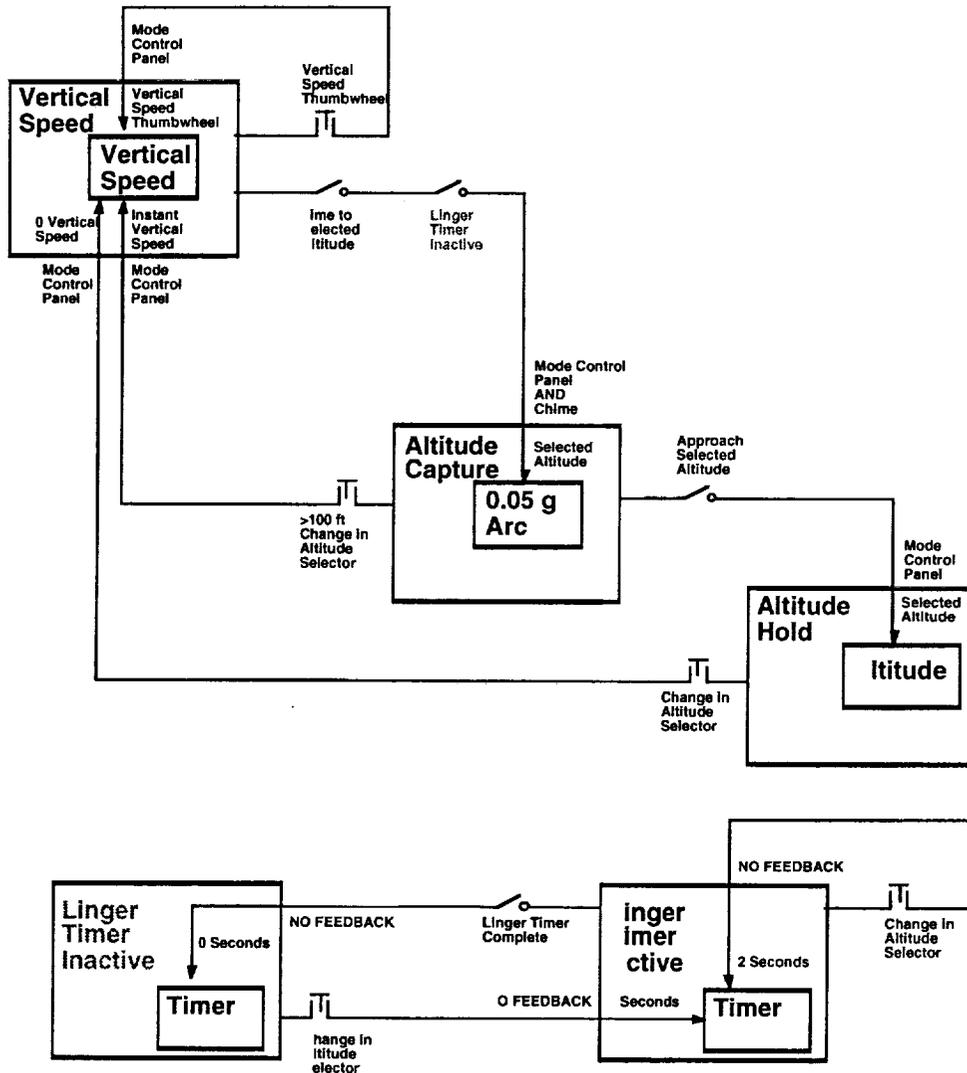
Figure 3.16: B737 Altitude Capture Mode Transition Diagram

The transition from Vertical Speed to Altitude Capture consists of two conditions being met. The criteria is the time to the selected altitude—the transition will occur when the altitude set in the Mode Control Panel is approached. In addition, the "Linger" timer must be inactive. This timer is shown in the bottom half of Figure 3.16 and is an example of an automatic and hidden behaviour in the autoflight system of this aircraft. Much more detail will be presented regarding this mechanism in the next chapter.

3.4 Relation to Other Modeling Efforts

The Hybrid Automation Representation is one example of the efforts underway to model autoflight systems. Several other efforts exist with differing approaches, goals, and representations. These efforts will be discussed to compare them with the approach presented in this thesis.

3.4.1 OFAN

Asaf Degani (1994) has developed a modeling representation called "OFAN," which uses StateCharts to represent the interaction between the different modules in automation. StateCharts are an extension to Finite State Machines utilizing hierarchical structures to allow the modeling of large systems, concurrency to enable the analysis of simultaneous processes, and a broadcast mechanism to allow state changes across multiple concurrent systems. For these reasons, StateCharts are particularly applicable for the modeling of large, complex, reactive systems. Degani illustrates the use of StateCharts to model the environment of the automation, the user task, the interface, the control mechanism and the physical plants.

StateCharts are a powerful tool for describing a complex system. One of the results of the analysis in this thesis is that systems are being designed which are too complex for use by pilots. The tools which allow the analysis of these systems during design do not serve to mitigate the underlying issue related to complexity, though they do serve to allow exploration of complexity concerns. The goal of the analysis done with Hybrid Automation Representations is both to demonstrate the complexity of existing systems (and how this can result in incidents) and to motivate the creation of new systems which are less complex.

3.4.2 Operator Function Model

The Operator Function Model (OFM) is focused on the interaction between an operator and automation in a highly proceduralized environment, such as aviation (Callantine, 1994). The OFM is a structured approach to specify the operator tasks and procedures in a task analysis framework made up of modes and transitions. Using graphical notation, OFM attempts to graph the high level goals into simpler behaviours to allow the supervision of the automation. The power of OFM is based upon several important observations: the event-driven nature of

automation, the proceduralized nature of high risk tasks, and the fact that many of the transitions and decisions made during system operation are discrete in nature. These observations are consistent with those used as the basis of the design of the Hybrid Automation Representation.

The Hybrid Automation Representation has similar goals to the Operator Function Model, but is more focused on modeling the automation rather than the pilot or procedures. As such, the graphical representation can be more straightforward. The HAR also treats continuous behaviour in a manner which appears to be consistent with pilot's mental representations, by using a hierarchy based on the relationship between the discrete and continuous layers of the automation. This is in contrast to the functional decomposition used in OFM. Automatic, uncommanded transitions were found to be an important element in the understanding of automation behaviour. As such, they are highlighted within the HAR representation through the distinction between manual and automatically specified conditional statements on transitions.

3.4.3   Operator Procedure Model

Work has been underway at Honeywell (Shery, 1999) on the Operator Procedure Model, a methodology for the design and verification of "knowledge-based" systems as necessary for elements of the flight task. The goal of this model is to decompose flight missions into subtasks which are meaningful to pilots as determined by the pilots' representation of the tasks. Operational procedures are defined by scenarios, the conditions, context, and situation of the system, and an associated behaviour, which is the response of the system to a given scenario. These scenarios and behaviours are designed through a participatory process with senior pilots, flight tests, and avionics engineers. Operational procedures are captured in tables linking the scenarios (as described by a set of conditions) to behaviours.

The Operator Procedure Model has a great deal of promise in attempting to engage relevant participants early in design. In addition, it does not attempt to coerce the language use in describing system behaviour from that of the pilot to that of the engineer. Unfortunately, for the systems which have been modelled to date using this process, the tables generated are large, complex, and are difficult to examination for errors, inconsistencies, or design weaknesses.

3.4.4 SpecTRM

Work is underway on the development of an analytical tool designed to identify mode problems early in design (Leveson 1997, 1998). SpecTRM (Specification Tools and Requirements Methodology) is a toolkit, including a requirements specification language, for modeling safety critical systems. After casting a design into this model, it can be examined, both by human and by automated processes, for a set of known mode problems. The advantage of allowing automated checking is that it may catch errors, or sections susceptible to errors, not discernible by a human checker. The automated checker uses some fifty completeness criteria to determine whether the system is fully specified. These criteria are based both on a underlying formal mathematical completeness model and on the experience base of a designer of large systems.

SpecTRM has a great deal of promise in automated checking of requirements documents and design verification. What may be even more useful is the gradual adoption of the completeness criteria into use by the designers of systems, and the use of automated testing as a verification process. One of the goals of the Hybrid Automation Representation is to make apparent to designers the design choices that may result in confusion. It does not appear that this goal can be met through the use of SpecTRM.

3.4.5 Simplification Modeling

Denis Javaux (1998) has developed and applied a model of the mechanisms by which humans understand and interact with automation based on the frequential and inferential simplification that occurs over repeated usage. Unlike the other models presented, Javaux's work is designed to provide a theoretical basis, built upon psychological principles, for the manner in which pilots appear to simplify the automation. As such, it provides a basis for some of the predictive statements in the usage of the Hybrid Automation Representation.

Frequential simplification is related to the number of experiences pilots have with a given transition or mode. The more often a particular transition is seen, the more tightly tied it will become to the apparent initiating factors. Other factors which may influence the transition, will be ignored if not experientially reinforced, until the transition is not expected in the presence of these

conditions. The simplification which occurs is based on a lack of experiential interaction. Inferential simplicities are related to inappropriate extrapolation of behaviours: if a change in a particular switch results in a transition in *almost* all modes, it will be inferentially simplified to result in that transition in *all* modes.

3.5 Measuring Autoflight Mode Transition Complexity Using Cyclomatic Complexity

Transitions among modes have been identified as an area of complexity in the aircraft autoflight system in Section 1.1. This category of problems was highlighted in the ASRS review presented in Figure 1.3, in focused interviews with pilots and in an examination of the system documentation. The current section presents a technique with which to analyze transitions between autoflight modes at a detailed level. Cyclomatic complexity is an approach originally developed in structured programming and graph theory.

3.5.1 Cyclomatic Complexity

Cyclomatic complexity is an analysis technique originally used to examine the complexity of structured software written on mainframe computers (McCabe, 1976). In the analysis, cyclomatic complexity determines the number of linearly independent paths through the system. The original goal was to examine the complexity associated with multiple branching code modules or states to gain insight into the impact of structure programming. Part of the contribution of this work is to extend the approach to examine the complexity of any system which can be shown as a linked set of edges and nodes. A node is some type of state or decision point within a system, and an edge is a mechanism to connect nodes. A further contribution is to extend the analysis to be used in the examination of an autoflight mode transition. This is based on the premise that determining whether a transition will occur is dependent on the evaluation of a set of predicating conditions and is analogous to how a structured program is dependent on branching decisions to determine its flow of control.

Further, cyclomatic complexity appears to be a useful analysis tool to examine transition characteristics which are hypothesized to impact the apparent or perceived complexity of autoflight automation. In order to monitor an autoflight system, a pilot needs to be able to track

the evolution of the state of the automation in addition to the state of the aircraft dynamics. Since automation can directly control the behaviour of the aircraft, the state of the automation needs to be understood in order to predict the future aircraft state and to detect when it is inconsistent with what was intended or expected. In order to do this, a pilot must have a representation of the automation itself in order to monitor conformance.

Cyclomatic complexity is a rationale approach to analyzing autoflight mode transitions which counts the number of linearly independent paths. Each path corresponds to a set of evaluations which must be made by a pilot in order to ascertain the future state of the system. Cyclomatic complexity is dependent on the number and structure of the conditional elements in the transitions and is hypothesized to be useful in the analysis of the apparent or perceived complexity of a system. In particular, autoflight mode transitions are thought to have their apparent complexity impacted by the number and structure of conditional elements (Javaux, 1998). These characteristics correspond to those which are identified in the Mode Transition Diagram (discussed in Section 3.3.1), and which are hypothesized to have an impact on the apparent complexity. Figure 3.17 shows these elements. The starting and ending mode are necessary to identify the transition and the total number of modes is hypothesized to impact the system complexity and can be analyzed from the size of the Mode Transition Matrix.
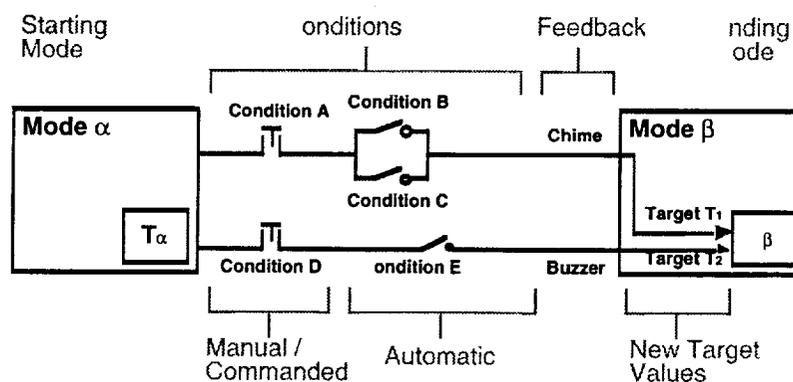
Figure 3.17: Mode Transition Diagram Abstraction

Another hypothesized factor is the number of transitions between modes and the number of different new target values which can be specified by the transition. Recall that the behaviour of a system is defined both by the active mode and its target value. As such, each transition path which

results in a new target value is considered independently. As an example, in Figure 3.17, there are two transitions paths, corresponding to the new target values T_1 and T_2. This also allows the representation, as shown in Figure 3.13, to stay in one mode while changing target values. Finally, the number and structure of the conditions are identified, as they can suppress transitions. Feedback that is provided for the transition is a tool to allow the human operator to monitor the transitions as they occur—it is a mechanism to mitigate the effects of complexity, and is not thought to directly impact the implicit or structural complexity of the transition at a given level of abstraction.

Rasmussen (1986) and others (Norman, 1988) discuss the necessity to effectively interact at a knowledge-based level. At this level of understanding, the pilot must have a model of the automation which can be cognitively "run" in order to predict future aircraft states. Multiple possible outcomes are generated based on this predictive analysis and the most likely of these outcomes is selected. The complexity of this model has an impact on the perceived complexity of the automation. Cyclomatic complexity is a tool to analyze the structure of the model utilized by the pilot in the process of monitoring the automation. Each linearly independent path through the autoflight systems corresponds to a set of evaluations in the process of monitoring.

3.5.2   Analysis Using Cyclomatic Complexity

Cyclomatic complexity is a measure of the number of linearly independent paths through a system of edges and nodes. This is straightforward for a "strongly connected" system, where every mode can reach every other mode through some path. In a strongly connected system, the number of independent paths has been shown to be Equation 3.1 where v is the number of independent paths through the system, e is the number of edges, and n is the number of nodes. Equation 3.1 allows the rapid assessment of the linearly independent paths but is only applicable for strongly connected systems.

$$v = e - n + 1 \qquad\qquad \text{Equation 3.1}$$

If each node only has a single edge, only one path will exist through the system, as shown in Figure 3.18. The single path through the system is from node 1 to 2 to 3 to 4.
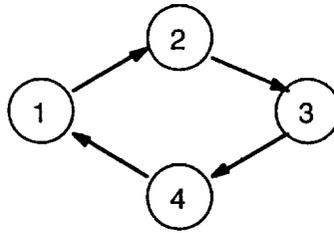
Figure 3.18:   Simple Strongly Connected System: $v = 1$

If an additional edge is added the new set of linearly independent complete set of paths must include the new edge. This results in an increase in cyclomatic complexity corresponding to the number of edges in excess of the number of nodes. Figure 3.19 has two independent paths, corresponding to the original circuit shown in Figure 3.18 and the added edge, creating the direct path from node 1 to 3. This is consistent with Equation 3.1.
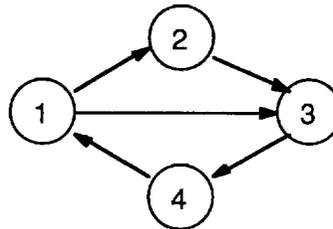
Figure 3.19:   Simple Strongly Connected System with Additional Edges: $v = 2$

Linearly Independent Paths in Program Control Graphs

The original target of cyclomatic complexity was structured programs described by program control graphs (McCabe, 1976). These graphs have a single entry node and a single exit node. Each node can be reached by the entry node and each node can reach the exit node though some set of edges. Program control graphs do not need to be strongly connected, and typically are not, since the exit node does not connect to other nodes. This is apparent if we reverse the direction of the arrows in Figure 3.18 between nodes 1 and 4 and nodes 3 and 4. This is shown in Figure 3.20. Note that there is no path connecting nodes 2 to 4 or nodes 1 to 3. Node 3 in Figure 3.20 is a "terminal node," defined to be the node at the exit of a program graph.

74
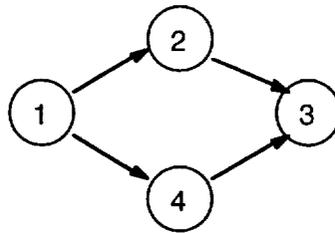
Figure 3.20:   Non-strongly Connected System

Virtual edges must be added in order to make the system strongly connected, thereby allowing the use of Equation 3.1 to count the number of paths. The dotted line from node 3 to 1 in Figure 3.21 indicates the virtual edge between these two nodes which has been added in order to convert it into a strongly connected form. These edges add to the complexity of the system. All nodes now have a path to all other nodes: node 2 can connect to node 4 through nodes 3 and 1. Node 3 is directly connected to node 1.
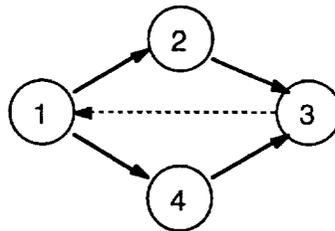
Figure 3.21:   Strongly Connected System through Additional Virtual Edges: $v = 2$

More generally, each terminal node will require a virtual edge in order to make the system strongly connected. Using this generalization, Equation 3.1 can be refined to utilize virtual edges, resulting in Equation 3.2, where v is the number of independent paths through the system, e is the number of edges (transitions), n is the number of nodes (modes), and t is the number of terminal nodes. For Figure 3.21, there are four edges, four nodes, and one terminal node, so the cyclomatic complexity is 2.

$$v = e - n + (t + 1)$$ Equation 3.2

An edge and node diagram with multiple terminal nodes is shown in Figure 3.22. The terminal nodes have been shown as squares to differentiate them from the other nodes and the virtual edges have been shown as dashed. The total number of linearly independent paths in this diagram using Equation 3.1 with 5 edges and 4 nodes, is 2 paths. Using Equation 3.2, the virtual paths need not be counted, but the result of 3 edges, four nodes and two terminal nodes is also 2 paths.
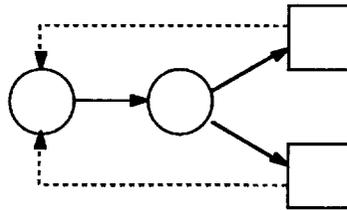
Figure 3.22:   Edge and Node diagram with Multiple Terminal Nodes (squares are terminal nodes)

Extending Cyclomatic Complexity to Mode Transition Diagrams

By mapping Mode Transition Diagrams to the edge and node diagrams used to determine cyclomatic complexity, a measure can be made of the number of the cyclomatic complexity of transitions between modes. This measure corresponds to the number of possible manners in which such a transition could occur—the number of linearly independent paths in the transition.
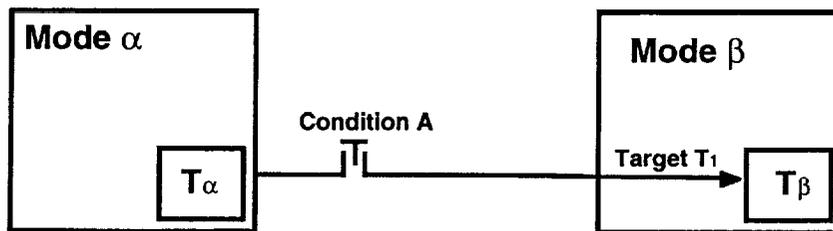
Figure 3.23:   Simple Conditional Transition

To measure the cyclomatic complexity of a mode transition, the starting mode and each combination of ending mode and new target value is considered to be a distinct node. Each *condition* is also a node: each condition node is a check as to whether that condition has been

satisfied. Reducing Figure 3.23 into this set of edges and nodes results in an edge and node diagram such as the example shown in Figure 3.24. A node in this representation can either be a mode or a condition, however, that the starting and terminal nodes correspond to *modes* rather than *conditions*. The distinction between these is that a mode will correspond to a quasi-steady-state behaviour of the system to be modelled using control block diagrams as discussed in Section 3.1. In contrast, conditions do not have an associated behaviour.
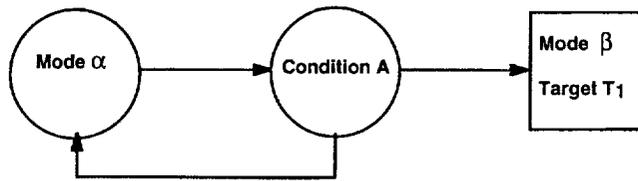
Figure 3.24:   Edge and Node Diagram of Simple Conditional Transition: $v = 2$

The system initiates in Mode α, and branches to Mode β if Condition A is true, or back to Mode α if Condition A is false. The cyclomatic complexity of Figure 3.24 is calculated to be 2 using Equation 3.2 based on three nodes, three edges, and one terminal node.

3.5.3   Measuring the Cyclomatic Complexity of Transitions

In order to be useful, the cyclomatic complexity must be able to analyze the impact of Boolean additions to transitions between modes. Conditions can be combined by ANDs and ORs, and each has an impact on the cyclomatic complexity. This section will examine the impact on cyclomatic complexity of each combination. Cyclomatic complexity must also be sensitive to the use of multiple target values.

Multiple Conditions

Figure 3.23 shows the simplest possible transition, predicated on a single condition and with a single ending mode and new target value. Multiple conditions can be combined by using Boolean operations (ANDing or ORing) in the transition. Figure 3.25 shows a transition in which two conditions, A and B, must be satisfied.
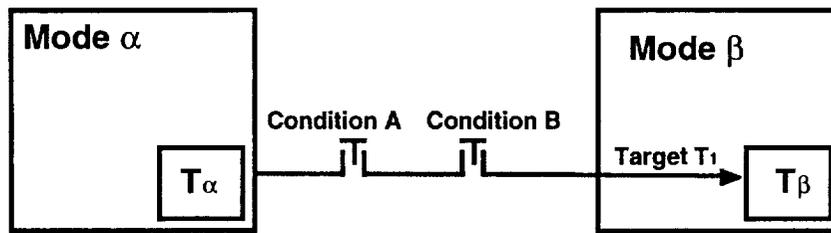
Figure 3.25: Multiple Conditional Transitions

The equivalent edge and node diagram is shown in Figure 3.26. In this diagram, Condition A must be satisfied in order for control to pass to Condition B and then on to Mode β. If either Condition A or Condition B is false, the system remains in Mode α. The cyclomatic complexity of this system results from 5 edges, four nodes, and a single terminal mode: $v = 3$. Each additional conditional element will increase the cyclomatic complexity by one.
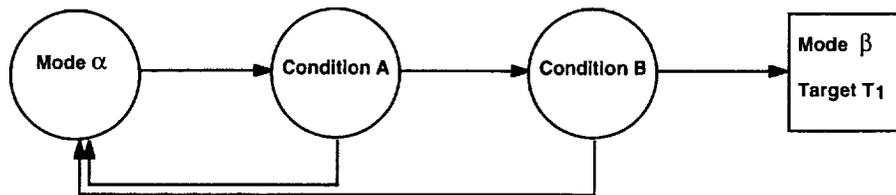
Figure 3.26: Edge and Node Diagrams of Conditional Transition with an AND: $v = 3$

ORing Multiple Conditions

Figure 3.27 shows a mode transition in which either Condition A or B must be satisfied in order for the transition to occur. The transition will occur if either path is completed by Condition A or B being satisfied.
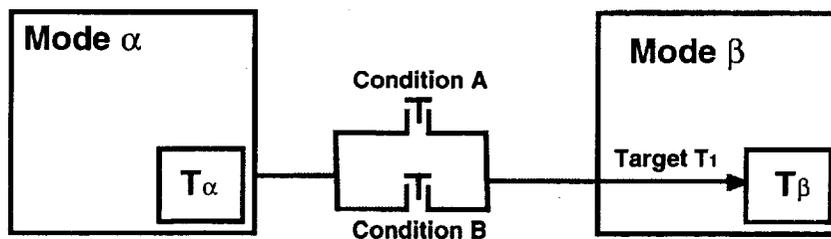
Figure 3.27: Conditional Transition with an OR: $v = 4$

The equivalent edge and node diagram is shown in Figure 3.28. In this diagram, either Condition A or Condition B can be satisfied. Once satisfied, the transition to Mode β will occur. The cyclomatic complexity of this system results from 6 edges, 4 nodes, and 1 terminal node: $v = 4$. Each additional conditional element which is connected by an OR, rather than by an AND, will increase the cyclomatic complexity by 2—the cyclomatic complexity of this example is two greater than Figure 3.25. This is consistent at an intuitive level since the cyclomatic complexity is a measure of the number of linearly independent paths. Each additional path which is added, without a corresponding node, will increase the cyclomatic complexity by one. In the case of an additional condition being added by an OR, the cyclomatic complexity increases by one for the condition, and one for the additional path created by the branching OR.
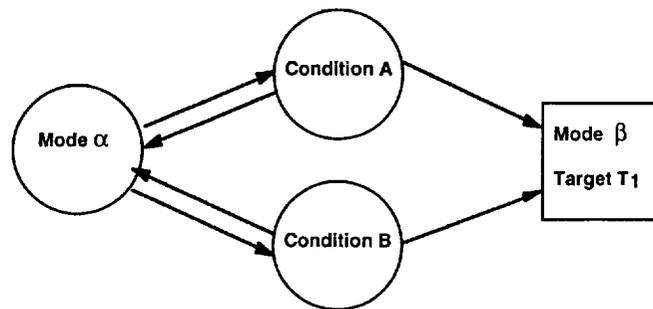
Figure 3.28:   Edge and Node Diagrams of Conditional Transition with an OR: $v = 4$

Multiple Target Values

An additional extension was required to consider mode transitions with multiple new target values. An example of a mode transition diagram with multiple possible target values is shown in Figure 3.29. In this diagram, if Condition A is satisfied, transition to Mode β will occur with a new target value of T_1. If Condition B is satisfied, transition to Mode β will occur with a new target value of T_2.

The edge and node diagram for this mode transition matrix is shown in Figure 3.30. In order to accurately analyze the two distinct new target values used by Mode B, two terminal nodes are created. As defined earlier, the behaviour of the automation is defined by the active mode and its target value. In an identical manner, when measuring cyclomatic complexity a terminal node is
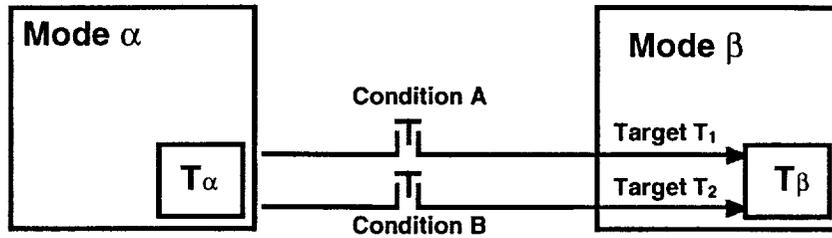
Figure 3.29:   Conditional Transitions with Multiple New Target Values

defined by the active mode and its target value. This also results in needing two virtual edges for the two terminal states to satisfy the "strongly connected" criterion. These are the edge connecting Mode β/T_1 back to Mode α, and connecting Mode β/T_2 back to Mode α. The cyclomatic complexity of this diagram is calculated from the 6 edges, 5 nodes, and 2 terminal modes so $v = 4$.
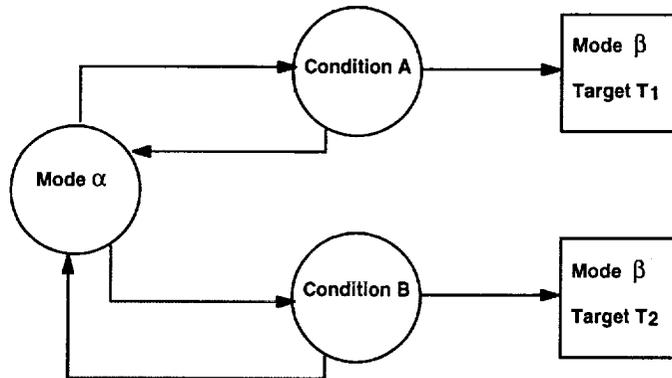
Figure 3.30:   Edge and Node Diagram of Multiple New Target Values: $v = 4$

Combining each of these elements allows the creation of the edge and node diagram for the more complicated mode transition diagram shown in Figure 3.31. The associated edge and node diagram is shown in Figure 3.32. The cyclomatic complexity of this mode transition matrix is based on 13 edges, 8 nodes, and 2 terminal modes: $v = 8$.

Cyclomatic Complexity and Repeated Sets of Conditions

Sets of conditions may appear multiple times within an autoflight system. Each instance of a set of conditions which appear multiple times should not be counted towards the cyclomatic

80
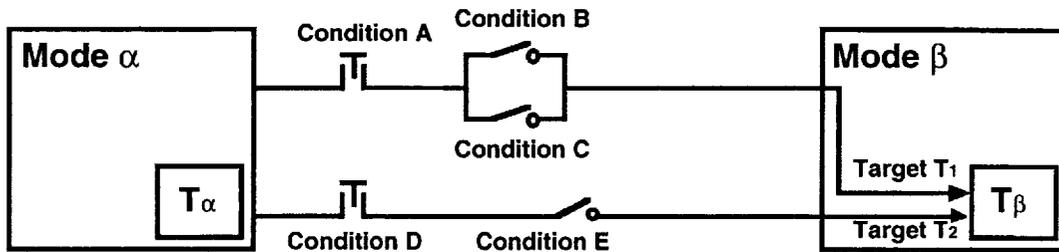
Figure 3.31: Complex Mode Transition Diagram.
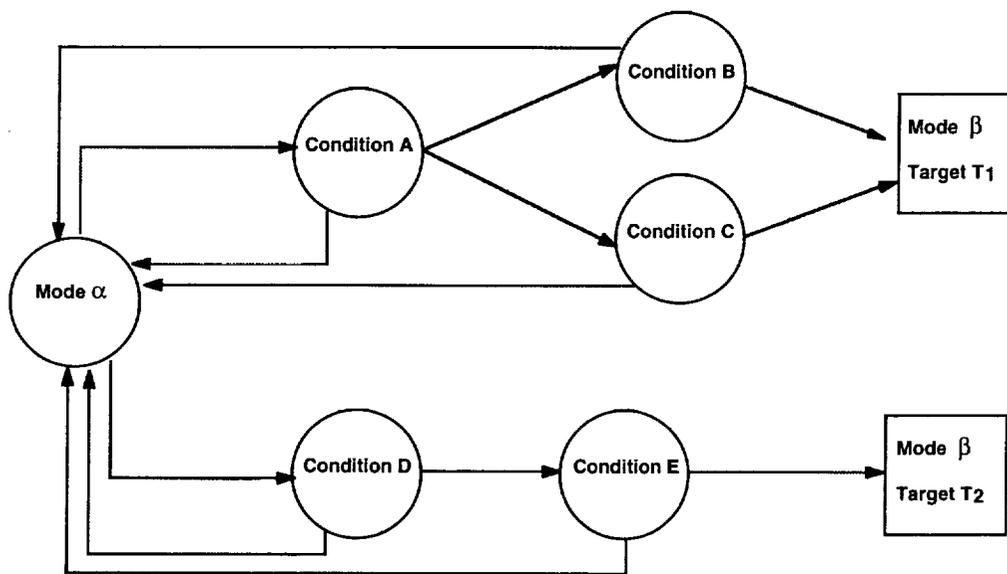
Figure 3.32: Complex Edge and Node Diagram: $v = 8$

complexity of the overall system. Instead, this commonality should be captured explicitly as a subset and used to effect a reduction in cyclomatic complexity.

Figure 3.33 shows a mode transition diagram which has a pair of repeated condition sets. Conditions C and D are arranged in an identical manner in each of the transition paths between these two modes.

A straightforward conversion into an edge and node diagram is shown in Figure 3.34. The cyclomatic complexity of this diagram is very large—10—since it is based on 16 edges, 9 nodes,
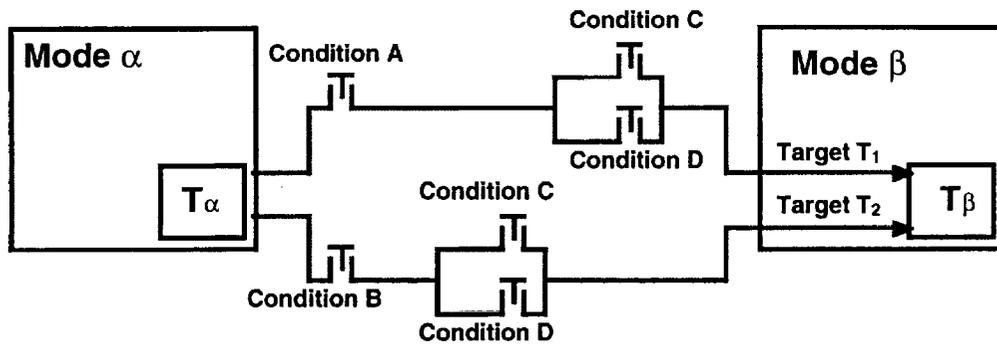
Figure 3.33:   Mode Transition Diagram with Common Conditions

and 2 terminating nodes. This conversion does not attempt to take the commonality between the condition sets into account.
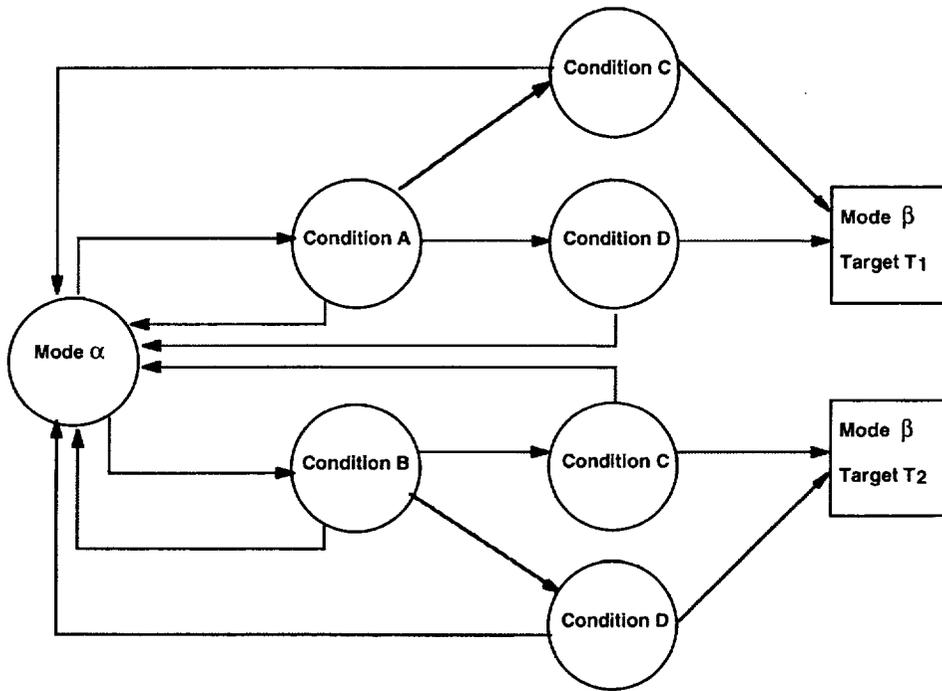
Figure 3.34:   Straightforward Conversion of Common Conditions into Edge and Node Diagram
$v = 10$

Figure 3.35 shows another conversion to an edge and node diagram. In this figure, the common conditions, namely C and D have been placed into a element labelled "Set CD." This

element contains the OR transition, an entry node and two exit nodes, depending on if the Boolean conditions are found to be true or false. In Figure 3.35, the TRUE exit node will lead to Mode β and the FALSE exit node will return to Mode α. Note that the set does not have terminal nodes since it is a "sub element" in the system. As discussed earlier, the only terminal nodes correspond to actual autoflight modes. Since each of the elements within a set is a condition, there are no terminal nodes in a set.

The cyclomatic complexity of the network can then be computed as before, treating any repeated condition sets as single nodes and then adding the contributions of the common conditions (the details within set CD). However, each repeated set need only be counted one time, since it represents a common factor. This corresponds to the program graph approach of capturing subroutines as independents set of edges and nodes.
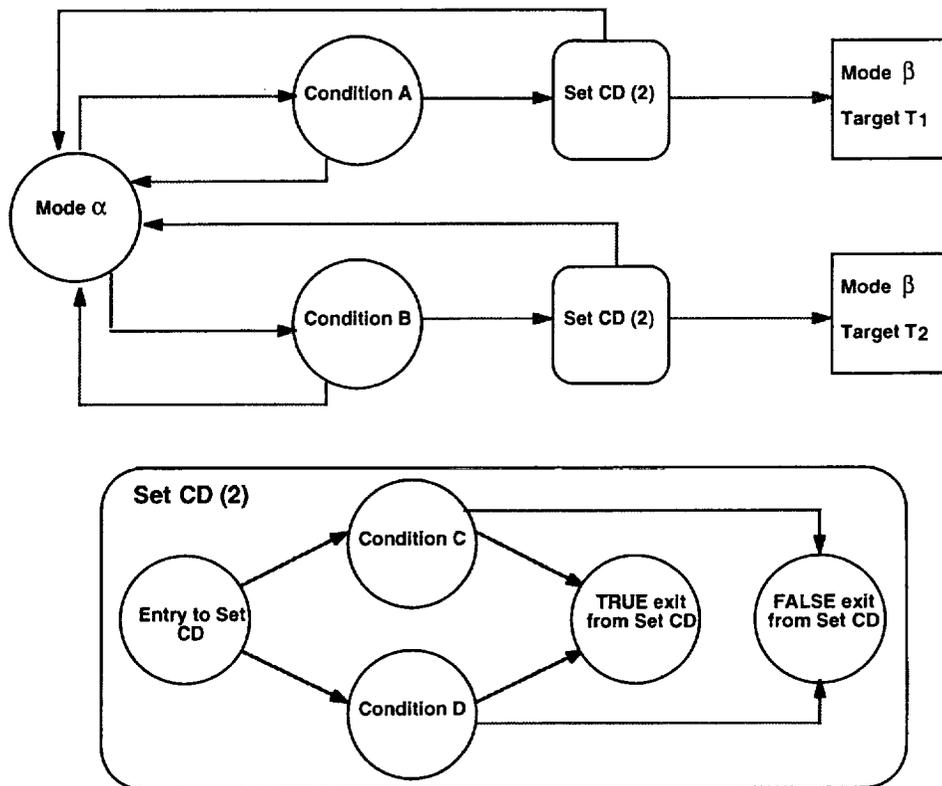
Figure 3.35:   Conversion of Common Conditions into Edge and Node Diagram: $v = 8$

For Figure 3.35, the total cyclomatic complexity includes of the portion associated with the top diagram (10 edges, 7 nodes, and 2 terminal nodes), $v = 6$. Since this is a subset of a strongly connected system Equation 3.1 should be used to calculate the incremental addition of the set. Measuring Set CD results in (6 edges, 5 nodes) $v = 2$. The cyclomatic complexity of the entire system is 8. Note that this is lower than the cyclomatic complexity of Figure 3.34, which was 10.

This lower cyclomatic complexity is contingent on the Set CD being used twice in the system. If the cyclomatic complexity of the transition from Mode α to the Mode β/Target T_1 end state is measured independent of the transition from Mode α to the Mode β/Target T_2, this reduction will not be realized. The cyclomatic complexity of the top transition is (5 edges, 4 nodes, and 1 terminal nodes), $v = 3$, and including Set CD ($v = 2$) totals to $v = 5$. Similarly, counting only Mode α to the Mode β/Target T_2 results in $v = 5$, for a total cyclomatic complexity of 10 as seen earlier in Figure 3.34. By utilizing sets to capture common elements, the sum of cyclomatic complexity of each transition measured independently may not be the total cyclomatic complexity of the set of transitions. Instead, care must be taken to only count common subsets once and to use these common subsets to accurately analyze the cyclomatic complexity of the system. Also, note that the recasting of the system into sets of conditions does not change the level of details of the system; sets simply capture repeated groups of transitions.

Simplified Cyclomatic Complexity Counting

Nodes which have more than one exit edge increase cyclomatic complexity. Conditions increase cyclomatic complexity because they are a decision point with two exits, one when the condition is true and one when it is false. Two edges can also exit a node when two paths are possible to the next nodes, as in a OR condition. Again, each additional edge will increase the cyclomatic complexity by one. In the general case, if n edges exit a single node, the cyclomatic complexity will increase by (n-1).

Using this information and by examining the incremental impact of additional conditions, a simpler analysis can be made of the cyclomatic complexity autoflight systems. The most basic transition diagram, consisting of a single conditional predicate, is shown in Figure 3.23, and has a cyclomatic complexity of 2. The mode transition diagram shown in Figure 3.25 and its associated

edge and node diagrams show that each additional conditional element increases the cyclomatic complexity by one. Each additional path through the system, introduced by the branch in an OR construct also increases the cyclomatic complexity by one. For example, Figure 3.27 shows that allowing two edges to exit a single node, a Boolean OR, also results in a cyclomatic complexity increase of two, one for the branch and one for the additional condition. In a mode transition diagram, each OR will increase the cyclomatic complexity in a similar manner.

Using this information, a simpler measure of the cyclomatic complexity of a mode transition, after consistent elements have been captured in sets of transitions, is shown in Equation 3.3. C is the number of conditions in the transition, B is the number of branches associated with each OR, and t is the number of terminal modes.

$$v_{transition} = C + B + t \qquad \text{Equation 3.3}$$

As an example, Figure 3.36 shows a transition with three branches. The cyclomatic complexity of this transition is calculated based on 4 conditions, one terminal state and the three branches, for a total of 8.
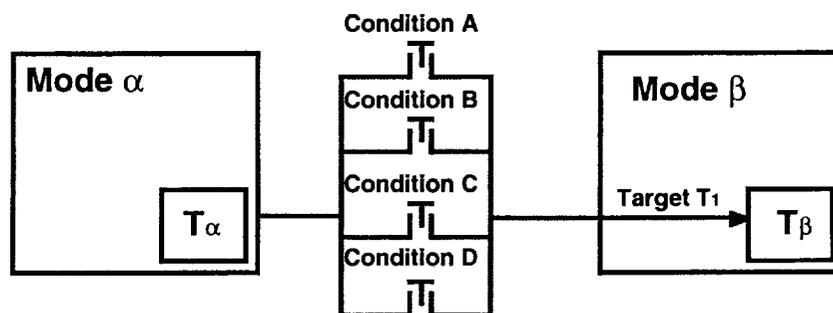
Figure 3.36:   Four-way Conditions Connected by ORs: $v = 8$

The cyclomatic complexity of repeated sets of conditions can be analyzed in a related manner with the difference that each common set is already encapsulated in a single condition, and that these sets do not have any terminal nodes. As an example, Set CD in Figure 3.35 is encapsulated in a single condition in the upper diagram. Therefore, the effect of a each set, regardless of the

internal details, is to increase the cyclomatic complexity as a single condition. In order to acknowledge the single condition represented by each set, the cyclomatic complexity of each set is measured by Equation 3.4. This bookkeeping allows the consolidation of conditions into a set to have the appropriate impact on the cyclomatic complexity.

$$v_{set} = C + B - 1$$

As an example, consider the set of 4-way conditions in Figure 3.36. If these conditions were captured in a common set, the mode transition diagram would be reduced to the one shown in Figure 3.37. The cyclomatic complexity of this system consists of a contribution of 2 from the transitions encapsulating Set ABCD and a contribution of 6 from Set ABCD (based upon 4 conditions, and 3 branches). As shown earlier using Equation 3.3, cyclomatic complexity is also 8, based upon 4 conditions and 3 branches.
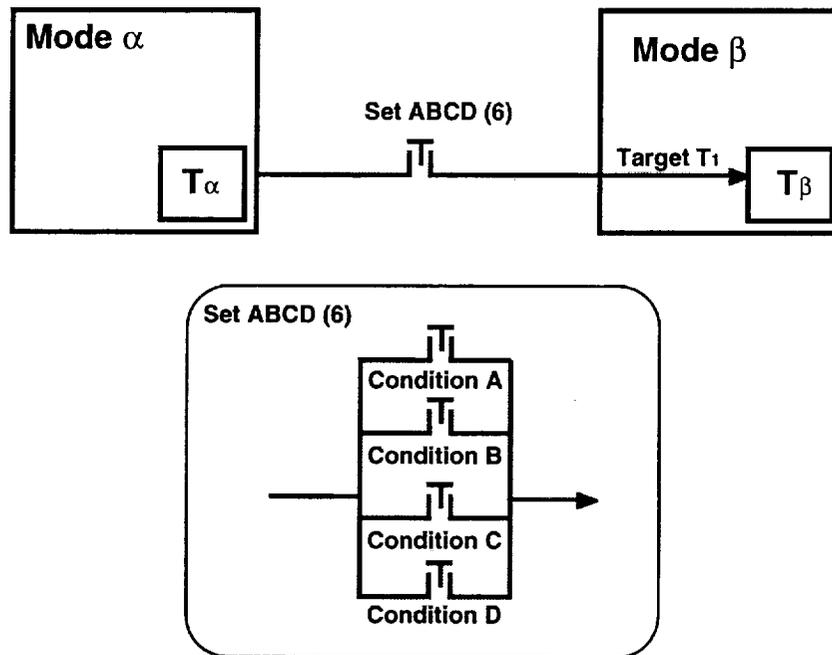
Figure 3.37: Cyclomatic Complexity Measurement of Common Sets: $v = 8$

3.5.4 Level of Abstraction and Apparent Complexity

A characteristic of cyclomatic complexity is that it is sensitive to, and a function of, differing levels of system abstraction. A complex system can be modelled at multiple levels. At a low level of abstraction it becomes difficult to mentally model the system; there are too many variables which must be evaluated. In the extreme case, it would be impossible for a pilot to evaluate the state of the aircraft utilizing the machine code running the autoflight system. At a high level of abstraction, sufficient detail may not be available to accurately predict the future state of the aircraft. As an example, if the abstraction does not take the state of the flaps into account, it may not be sufficiently accurate for monitoring the system.

Note that the repeated set of conditions discussed earlier are *not* a direct means to change the abstraction level of the system. Breaking a system up into sets prevents the overcounting of the number of linear paths through the system, while maintaining a constant level of detail and abstraction. However, if the conditions in a set are related in an operational manner, they may become abstracted and modelled by the pilot as a single conditions. In this situation, the details within the set are not modelled or considered, thereby reducing the apparent complexity of the systems. The cyclomatic complexity is also lower since fewer conditions are modelled. However, a system abstracted at a higher level necessarily has fewer details and may be less able to be accurately monitored. As discussed in Section 3.5.1, the accuracy of the model used to represent the system will impact the ability to monitor the system. A more consistent representation may be able to be abstracted a a higher level with less loss of relevant underlying detail.

This supports the concern regarding the lack of a consistent global model of automation. Without such a model, it is not possible to exploit consistencies in order to allow reductions in system complexity. Further, the lack of such a model implies that abstractions that are created will be more likely to be unable to capture relevant operational detail. If autoflight systems are reaching a limit from the standpoint of the pilot being able to monitor the complex system, then a notional conservative quantity of "apparent complexity" can be hypothesized. When more of this conservative quantity needs to be used to broadly model the inconsistent elements of automation, less is left to deeply model specific modes. In addition, abstractions which need to be made to manage complexity will be less able to capture specific transition details. The lack of such a

consistent, abstractable model may result in the complexity management techniques affecting a loss of understanding of the system. As such, it is felt that the unconstrained growth of these systems may be contributing to the autoflight system safety concerns in modern aircraft.

Abstraction Level Impacting Cyclomatic Complexity

Cyclomatic complexity can be applied at varying levels of abstraction. The level of abstraction will impact cyclomatic complexity and care must be taken to apply this method in a consistent manner. The approach that was used in the course of this research was to analyze based solely on the material available in the Flight Crew Operators' Manual. This information is presented in a manner designed for operational usage and is an appropriate level at which to evaluate these systems because it is likely to be related to the training material and representations which were used to build the pilots' mental models.

Cyclomatic complexity is dependent on some of the details of the representation. This is not only the level of abstraction, but also the particular Boolean operations which are used to construct conditions. Earlier, it was discussed that an additional OR added 2 to the cyclomatic complexity whereas an additional AND only added 1. However, it is possible to use Boolean equalities to convert ANDs to ORs: $\overline{A} \text{ AND } \overline{B} = \overline{A \text{ OR } B}$. As such, the particulars of the representation will have an impact on the measured cyclomatic complexity.

The appropriate representation is at an abstraction level which is both fully accessible and useful to the pilot. This is both a pilot and environmental/contextual issue: the determination of appropriate abstraction level is dependent on the skills, training and aptitude of the intended audience and on the operational requirements of the system. As a basic premise, elements and conditions which have an operational impact must be captured in the appropriate representation. In addition, a system which has an invariant operating regime may be able to be abstracted at a very high level by allowing assumptions to be made by designers. A more dynamic system may require abstraction at a low level in order to provide the flexibility to deal with a changing operating environment. The flexibility required by the operation environment must considered during abstraction level specification.

3.5.5 Complexity Management

As discussed in Section 3.5.4, abstraction of a system at a higher level is one approach to reducing the apparent complexity of the system. The cyclomatic complexity of a system abstracted at a higher level will also be reduced. If pilots abstract systems in some manner in order to manage the apparent complexity it may be possible to pro-actively incorporate these technique into training and design. In order to reduce the operational complexity of flight automation systems, pilots are thought to use techniques to allow modeling a simpler, more tractable, system (Morris 1987, Johnson-Laird 1983). Through discussions with pilots and anecdotal conversations, several techniques have been tentatively identified. It is important to note that these techniques may limit system functionality as pilots attempt to reduce the system to a tractable state.

Reduce Size of System

One broad approach to reduce the apparent complexity is to not utilize portions of the autoflight systems. At the broadest level, entire sets of modes can be ignored—not using VNAV is an example. It may also be possible to avoid the use of particular individual modes. Between modes, particular paths may be used exclusively. By only using a well understood subset, pilots can effectively fly with a simpler system, but lose access to some of the advanced capabilities. This approach reduces the apparent and cyclomatic complexity of the autoflight system by limiting capabilities. Unnecessary modes are effectively removed or ignored. The richness of the automation behaviour is pared down to the operationally relevant subset.

One of the issues hypothesized in the autoflight system is the number of conditional elements which determine when a mode transition will occur. Reducing the number of conditions decreases the cyclomatic complexity. Specifically, in some of instances, there are conditional clauses which are fulfilled the vast majority of the time a transition is commanded to occur. Conditions which are rarely require evaluation will become ignored. This has been termed "frequential simplification" (Javaux 1998). Note that this technique, which is reinforced by learning though repetition, is a serious concern. In many cases, emergency modes or transitions associated with non-nominal transitions have differing responses and behaviours. If these differences are ignored

by the pilots, the effectiveness of these modes in emergency situations may be seriously undermined.

Reduce Possible Paths Through Autoflight System

Another approach is to segment the operation of the automation into less flexible, but well-understood sequences in order to reduce the possible behaviours which must be monitored. Rather than only changing the altitude target of the aircraft, a pilot may choose to change the altitude, reset the speed mode, and then select the vertical mode. This sequence of events always has a known outcome, whereas using the individual modes which make up the chain may not be individually modelled or have a known outcome. Completing such sequence may allow more predictable behaviour out of the system by avoiding rarely used states by constantly resetting the system to a known configuration. By doing so, only particular paths through the overall system may be used. This effectively reduces the number of branches which exist in a mode transition diagram. If entire transitions are removed completely, even more reductions can be realized.

In a similar manner to the previous approach, explicitly using a subset of paths through the automation can reduce the capabilities of the system. In this case, pilots are proactively taking actions to remain within the subset of automation with which they are familiar and comfortable.

Difference Approach

The premise of this technique is to take advantage of known and perceived consistencies within and across aircraft generated by the incremental growth of automation. The fact that successive generations of aircraft automation are largely supersets of previous versions (see Section 2.2) implies that the differences between generations may be limited. Essentially, rather than modeling a new automation system in its entirety, the pilot will make note of the differences between the new system and a known system. Statements such as "This FLCH mode works just like it does in the 737, except..." are typical examples of this technique. Within an aircraft, modes which appear to have consistent behaviours and transitions are considered equivalent, with only the differences noted. The Difference Approach can be used for transitioning pilots, between their current and new aircraft. If a group of pilots is moving from a B737 to an A320, the commonalities between the aircraft can be exploited. In common cockpit designs, such as the

A320/330/340 or B757/767 families, the differences between systems are easily identified. This approach can be related to the measure of cyclomatic complexity by examining the additional complexity of the new system over the old.

While this difference approach can be an effective tool during transitional training, it may be a liability in understanding the fundamental structure of the new system. Extending a known, simple model to explain a more sophisticated system may become overly intricate. If the new system was designed around a new paradigm, such difference modeling may not be appropriate. Systems which are largely similar, but have minor differences are most accessible to this types of complexity management approach. If, however, these differences are in rarely used modes or transitions and do not have experiential reinforcement, the differences may be marginalized, and not distinguishable to pilots when necessary.

Implications of Operational Complexity Management Techniques

By examining the manners in which it is possible to manage the complexity of automation in an operational setting, insight can be gained into how to modify, update, and design such systems. Each of the approaches suggest a manner in which the apparent overall complexity of the system is reduced either through organizing or explicitly ignoring portions of the automation. If it is possible to gain insight into the manner in which pilots select to maintain subsets of functions and mental model, it may be possible to assist in shaping future systems through the creation of more appropriate models. If the manner in which this management is done can be characterized, the complexity management techniques could be adopted to pre-emptively reduce complexity during design stages. Rather than leaving the simplification of the system to individual pilots, it could be handled in a more structured manner through training, the initiation of procedures, more effective feedback, and through the use of modified design techniques.

It is important to note that the approaches suggested to manage complexity are all based, at some level, on making assumptions based on a consistent set of behaviours. In the absence of a consistent model, these techniques will result in a less complete understanding of the system as information about the system is ignored to make it more tractable. Unfortunately, it does not

appear that there is a consistent global model of automation to exploit in order to allow reductions in system complexity.

There are also some concerns about individual pilots reducing the complexity of the system, especially though the use of techniques which ignore parts of the system, conditional elements, or alternate paths through the automation. The learning which occurs during operation necessarily reinforces the modes, behaviour, and conditional elements which are seen most often (Johnson-Laird, 1983). Those which are experienced less often will be the ones which are removed from pilots' representations. A serious issue is that non-nominal or emergency modes are unlikely to be experienced directly with regularity. As such, these modes may become marginalized as a pilot has to deal with the burgeoning complexity of a system. Many automation behaviours exist with which pilots need to have a detailed understanding but will not occur regularly. As such, if poorly implemented these complexity management techniques have the capability of undermining flight safety.

3.5.6   Cyclomatic Complexity Mode Transition Matrices

Cyclomatic complexity can be used to populate a Cyclomatic Complexity Mode Transition Matrix to analyze specific sets of modes. The resulting matrix can be viewed at a detailed level to determine cyclomatically complex transitions, or in an aggregate manner of the overall set of transitions.

Figure 3.38 shows the mode transition diagram for a set of modes involved in altitude capture in the MD-11. The level of abstraction that was used was to model the system based on the contents of the Flight Crew Operators' Manual (Honeywell, 1992). The cyclomatic complexity of each transition is calculated using Equation 3.3. The Vertical Speed to Vertical Speed transition has a cyclomatic complexity of 2, from its single condition and terminal state. The Vertical Speed to ALTCAP transition has a cyclomatic complexity of 5 from 3 conditions and 2 terminal states. The ALTCAP to Vertical Speed transition is more complicated. It consists of 7 conditions, one OR branch and two terminal states, for a cyclomatic complexity of 10. The ALTCAP to Altitude Hold transition has a cyclomatic complexity of 2. Finally, Altitude Hold can transition back to
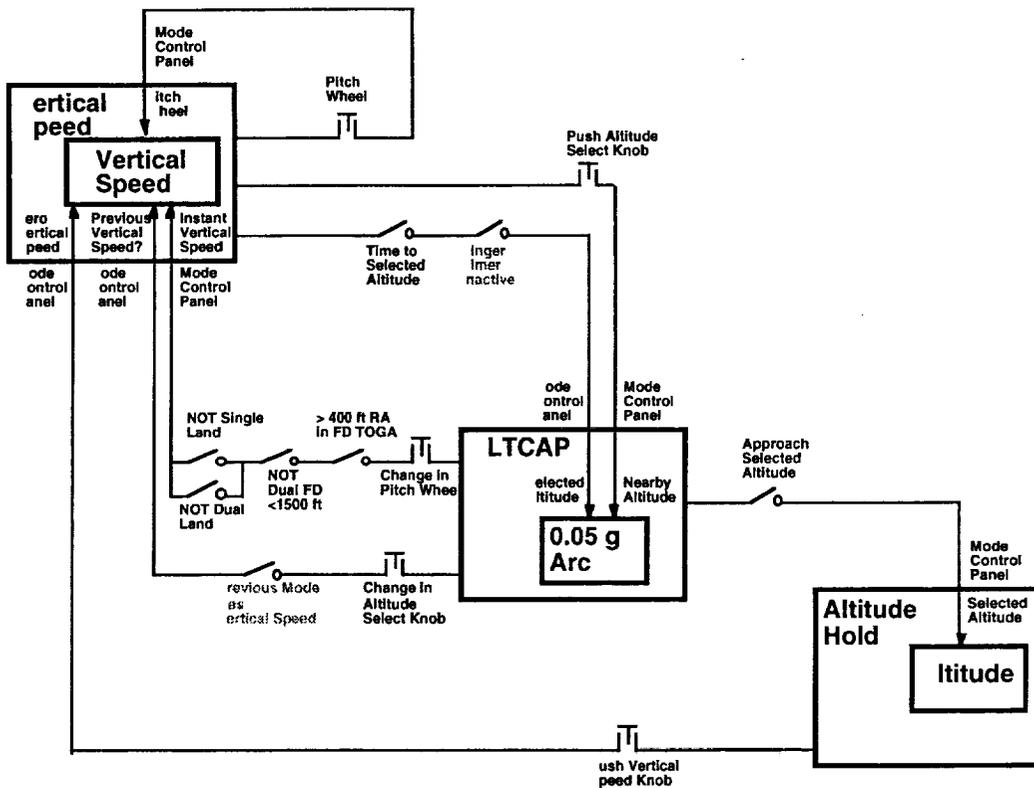
Figure 3.38: Mode Transition Diagram of MD-11 Altitude Change

Vertical Speed manually, with a cyclomatic complexity of 2. The independent transitions which populate the 3x3 Cyclomatic Complexity Mode Transition Matrix are shown in Table 3.4.

Table 3.4: Cyclomatic Complexity Mode Transition Matrix

	Vertical Speed	ALTCAP	Altitude Hold
Vertical Speed	2	5	--
ALTCAP	10	--	2
Altitude Hold	2	--	--

3.6   Chapter Summary

This chapter documented an analysis framework for aircraft automation which captured both the "quasi-steady-state" behaviour and the discrete behaviour to switch among multiple quasi-steady-state controllers. The former can be completely modelled using control block diagrams at

various levels of loop closure. The discrete behaviour is modelled using mode transition diagrams and matrices. Cyclomatic complexity was presented as a rationale basis to analyze mode transitions within the discrete portions of automation which is dependent on the representation and abstraction level of the system being measured.

Chapter 4

Web-based Pilot Automation Complexity Survey

In Section 2.3, numerous measures of complexity were discussed. The appropriateness of measures of complexity is one that needs to be examined carefully in order to determine the contributory elements to pilots perception of complex systems. Cyclomatic complexity was discussed in Section 3.5.1 as an analysis tool for autoflight mode transitions that captured the number of linearly independent paths through a transition. However, it has not been shown previously that there is a relationship between cyclomatic complexity and the apparent complexity from the viewpoint of the pilot. A survey was conducted to identify those modes which pilots found to be most complicated and to analyze them using cyclomatic complexity.

4.1 Survey on Automation Complexity

The previous discussions in Section 1.1 examined autoflight systems from an engineering viewpoint and accident and incident reports from a statistical viewpoint. A survey was conducted of line pilots with the goal of gaining insight into the "apparent complexity" of the automation, and into mode transitions in particular. The apparent complexity is an indication of the viewpoint of the end operator—in this case the viewpoint of the line pilot. While this measure is directly affected by biasing factors, including the experience and training of the pilot, it provides an indication of which transitions and modes are most difficult in practice.

> Based on a review of the Aviation Safety Reporting System (ASRS), we found situations where the autoflight system caught the pilot unaware or had some sort of unexpected behaviour. Many of these situations involved mode transitions. A mode transition occurs any time that the aircraft switches from one mode to another, such as between Vertical Speed mode and Altitude Hold mode. (Appendix D)

The survey was conducted via the World Wide Web, which enabled a broad population of pilots to take part anonymously. The focus of the survey was on the viewpoints of the pilots regarding transitions between modes. To start, pilots were presented with a background explanation of transitions. One of the explanation pages is shown in Figure 4.1. Pilots were asked

95

Types of Transitions

- **Manual transition:** caused by a pilot pressing a switch.
  Example: Boeing B777 in a Vertical Speed descent. When the HOLD button is pressed, the aircraft immediately switches to the Altitude Hold mode and holds the current altitude.

Vertical Speed Descent   Press HOLD button   Transition to ALT HOLD at current altitude

- **Automatic transition:** occurs when the aircraft switches modes without direct pilot intervention
  Example: Transition to Altitude Hold when an aircraft intercepts the altitude shown in the altitude window. During a Vertical Speed manuever in a B737, the aircraft will transition to Altitude Hold mode when this interception occurs

Vertical Speed Descent   Approach Altitude in Altitude Window   Transition to ALT HOLD at Altitude Window Altitude

- **Armed transition:** occurs when the autoflight system has been authorized or armed to make a transition.
  Example of this is the transition from a Glide Slope Armed mode to a Glide Slope Tracking mode. The autoflight system will not switch directly into the tracking mode unless it was previously armed by the pilot.

Glide Slope is Armed for Capture   Glide Slope Capture Occurs   Glide Slope Mode becomes Active

Figure 4.1:   Web-based Survey, Page 3

96

to identify the three most complex transitions, characterize and describe the transitions (shown in Figure 4.2), and complete a set of pairwise comparisons between sets of transitions (Figure 4.3).

First Mode Transition

Transition from [Mode A   ] to: [Mode B   ]

Type of transition (check all that apply):

               Manual ☐   Automatic ☐   Armed ☐

Estimate the number of different paths possible in this transition: [Select estimated number of Paths ⬍]

Describe the transition in as much detail as possible.
Think about explaining this to another crew member.
If possible, include the necessary conditions, possible paths, and outcomes for the transition:

```
Enter transition description here.



```

What makes this transition difficult?

```
Enter what makes the transition difficult here.



```

[Review Transition Types]   [Review Multiple Paths]   [Review Conditions]   [Continue]

Figure 4.2:   Web-based Survey, Page 8-10

Comparing Mode Transistions

We are interested in how complicated you feel difficult transitions are. In this final section, you will be asked to compare the mode transitions that you listed previously against each other. Please rate the following 10 comparisons between pairs of mode transitions.

Mode A to Mode B is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than Mode C to Mode D**

○   ○   ○   ○   ○

Mode A to Mode B is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than Mode E to Mode F**

○   ○   ○   ○   ○

LVL CHG or FLCH to ALT HOLD is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than Mode A to Mode B**

○   ○   ○   ○   ○

LNAV to Heading Select is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than Mode A to Mode B**

○   ○   ○   ○   ○

Mode C to Mode D is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than Mode E to Mode F**

○   ○   ○   ○   ○

Mode C to Mode D is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than LVL CHG or FLCH to ALT HOLD**

LNAV to Heading Select is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than Mode C to Mode D**

Mode E to Mode F is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than LNAV to Heading Select**

LVL CHG or FLCH to ALT HOLD is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than Mode E to Mode F**

LNAV to Heading Select is   Much Less Complicated   Less Complicated   Equally Complicated   More Complicated   Much More Complicated   **than LVL CHG or FLCH to ALT HOLD**

Figure 4.3:   Web-based Survey, Page 11

4.2   Demographic Results

A total of ninety-three responses were generated from pilots flying multiple aircraft types, as listed in Table 4.1. Military pilots constituted four of the responses, with the remaining eighty-nine being commercial air transport pilots. As shown in Table 4.1, the majority of the results were from modern "glass-cockpit" aircraft and from transitional aircraft, such as the more recent variants of B737. Five female pilots and eighty-four male pilots responded; four responses did not fill out the gender field. The average age of respondents was forty-three.

Table 4.1:   Breakdown of Responses by Aircraft Type (total n=93)

Aircraft Type	Number of Responses
Boeing B727	1
Boeing B737-100/-200	3
Boeing B737-300/-400/-500	6
Boeing B737-600/-700/-800	4
Boeing B747-400	6
Boeing B757/B767	17
Boeing B777	2
Airbus A300	2
Airbus A310	1
Airbus A320/330/340	12
Boeing MD-11	2
Other	37

Data regarding the flight hours is shown in Table 4.2, and identifies the majority of respondents being experienced aviators.

Table 4.2:   Flight Hours of Respondents (total n=93)

	Total Flight Hours	Hours in Current Type	Hours in 1999	Hours in 1998
Average	10 250	2 039	584	629
Maximum	27 500	10 000	2 500	1 850
Minimum	150	26	0	50
Standard Deviation	5 750	2 064	310	248

Forty-seven respondents identified themselves as having the rank of captain; thirty-three were first officers. Four identified themselves as "Pilot In Command" (PIC). Eleven respondents

99

identified themselves as instructors. Note that these numbers do not total the number of respondents, since individuals could have multiple positions.

4.3   Pilot Characterization of Complex Transitions

Many mode transitions were identified by multiple pilots in this survey. Table 4.3 show those modes which appeared most often. Note that these involve highly automatic behaviour of the aircraft.

Table 4.3:   Transitions Identified by Respondents

Transition From	Transition To	Number of Times Identified
Flight Level Change	Altitude Capture	12
Heading Hold	LNAV	10
Vertical Speed	Altitude Capture	7
Altitude Capture	Vertical Speed	6
Approach	Go Around	6
VNAV Path	VNAV Path Descent	5

Another manner in which to examine the data is to identify which modes appear most often as the starting mode in a transition. This is an indication of which modes are most complex to leave, but is biased towards those modes which are most commonly used. As shown in Table 4.4, vertical modes dominated the starting modes.

Table 4.4:   Starting Modes Identified by Respondents

Starting Mode	Number of Times Identified
Flight Level Change	20
Vertical Speed	18
Altitude Hold	17
Heading Hold	17
VNAV Path	13
Approach	11

Conversely, Table 4.5 shows those modes in which transitions end. Not surprisingly, Altitude Capture is identified the most often, since it is the mode through which one typically leaves Flight Level Change or Vertical Speed before transitioning to Altitude Hold.

Table 4.5: Ending Modes Identified by Respondents

Starting Mode	Number of Times Identified
Altitude Capture	25
LNAV	15
Vertical Speed	15
Approach	9
Flight Level Change	9
VNAV Path Descent	9

Examining which modes were identified most often (regardless of whether as a starting or ending mode)—results in Table 4.6. Once again this table is dominated by vertical modes which made up 70% of the identified transitions.

Table 4.6: Most Commonly Identified Modes

Starting Mode	Number of Times Identified
Vertical Speed	33
Altitude Capture	31
Flight Level Change	29
Altitude Hold	24
Heading Hold	22
LNAV	22
Approach	20

Another portion of the survey asked pilots whether the mode transitions which had been identified could be characterized as manual, automatic or armed transitions. The results of this question are shown in Figure 4.4. A total of 139 mode transitions were characterized by pilots. This represents approximately 50% of the total possible responses from the 93 respondents. Many pilots only detailed a single transition. Note that a single transition could be characterized by a pilot as part of multiple types. For example, the transition between Vertical Speed and Altitude Hold is characterized as both automatic (if a target altitude is intercepted) or manual (if the HOLD button is pressed). The results are shown in Table 4.4 and are consistent with the hypothesis that the automatic behaviour of aircraft leads to complexity. What was not anticipated is that nearly 50% of the most difficult transitions were identified as manual in nature and only 28% were armed.
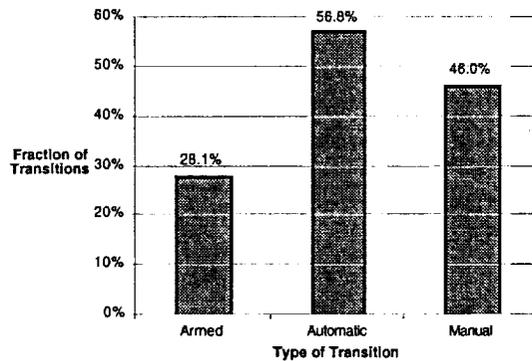
Figure 4.4:   Types of Transitions (n=139)

Pilots were also asked to identify the number of possible paths that could be taken to effect the transition, where a path was described as multiple ways in which a transition could occur. As an example, the transition between Vertical Speed and Altitude Hold can occur along two different paths. Either the Altitude Hold button can be pressed by the pilot, resulting in an immediate leveloff, or the aircraft can approach and leveloff at the altitude in the altitude window. In most cases, each path also has different final states. In this example, the automatic transition captures the value in the altitude window whereas pressing the Altitude Hold button results in the aircraft leveling off at a different altitude than the one shown in the altitude window. The number of paths in each transition is shown in Figure 4.5. The number of paths ranged from 1 to 7, with an average of 2.3 and a standard deviation of 1.2.
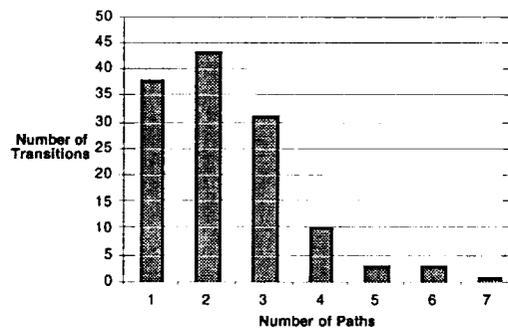
Figure 4.5:   Number of Paths per Transition (n=139)

4.4　Cyclomatic Complexity of Identified Complex Transitions

In order to analyze the cyclomatic complexity of the modes identified by pilots in the survey, it was necessary to identify a subset of responses for which detailed information, in the form of Flight Crew Operators' Manuals, was available. For the sake of consistency, only the information from the Operators' Manual was used; the Flight Crew Operators' Manuals specified the model and level of abstraction used to characterize the system. Additional information which could be brought to bear based on simulator testing was not considered. Each transition was characterized by its components of cyclomatic complexity based on an analysis of the transition as described in the training material and documentation. A total of twenty-nine transitions were analyzed in detail, for those cases which had sufficient autoflight documentation to characterize cyclomatic complexity factors. These were results from B737, B757, and A320 aircraft.

4.4.1　Comparison of Cyclomatic Complexity

Figure 4.6 shows the cyclomatic complexity of the analyzed transitions. The mean value of cyclomatic complexity is 6.45, with a standard deviation of 2.00. For comparison, the cyclomatic complexity of a representative set of non-emergency mode transitions from the B737-300 and B757 were calculated. This set was created by analyzing those all non-emergency modes which explicitly appear in the AutoFlight section of the respective Flight Crew Operators' Manuals. Figure 4.7 shows the cyclomatic complexity of these "typical" transitions along with the pilot-identified complex transitions in the same aircraft. This chart shows that the average cyclomatic complexity of the typical transitions is 3.91, with a standard deviation of 2.30. At a 95% confidence level the typical modes statistically have a cyclomatic complexity 1.4 lower than those transitions identified by pilots in the survey for the same aircraft.

This implies that there is a correlation between the cyclomatic complexity of a mode transition and whether it was considered complex from the perspective of the pilot. It appears that analyzing the cyclomatic complexity of autoflight mode transitions can provide insight into whether these transitions will prove to be operationally problematic for pilots.
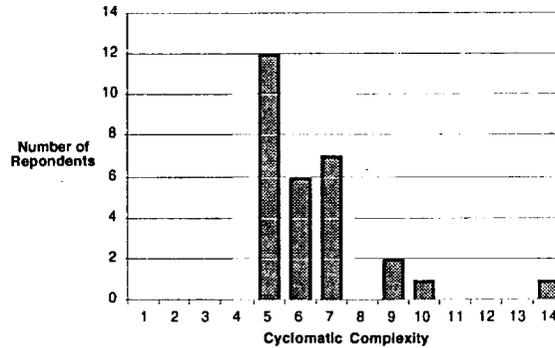
Figure 4.6: Cyclomatic Complexity in Complex Transitions (n=29)
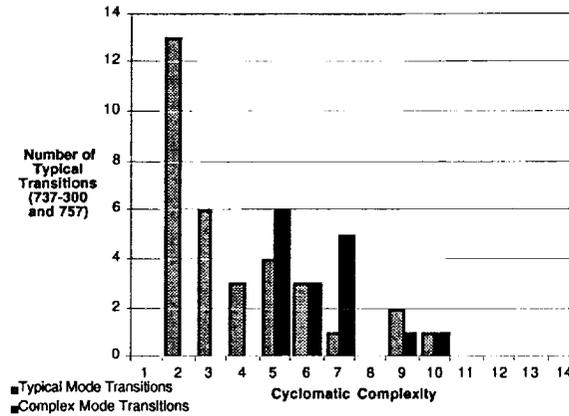
Figure 4.7: Cyclomatic Complexity of Typical Mode Transitions (n=33)

4.4.2 Number of Conditions

Figure 4.8 shows the number of conditions which appeared in analyzed transitions. The mean number of conditions in the transitions was 3.03, with a standard deviation of 1.09.

For comparison, Figure 4.9 shows the distribution of conditions for the typical transitions from the B737-300 and B757 discussed in Section 4.4.1. For the typical transitions, the mean number of conditions in the transitions was 2.00, with a standard deviation of 1.06. Statistically, there were 0.5 fewer conditions in typical mode transitions as compared to the identified complex transitions in these aircraft. Based on this data, there appears to be an indication that number of conditions has an impact in apparent mode complexity.

104
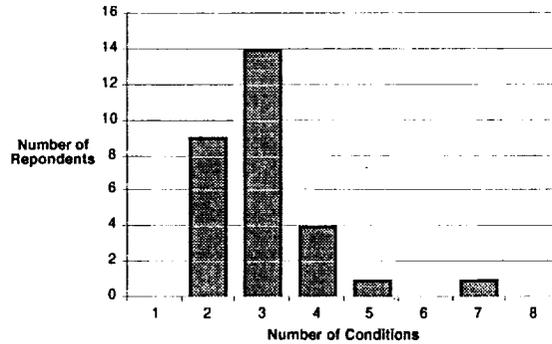
Figure 4.8:   Number of Conditions in Identified Complex Transitions (n=29)
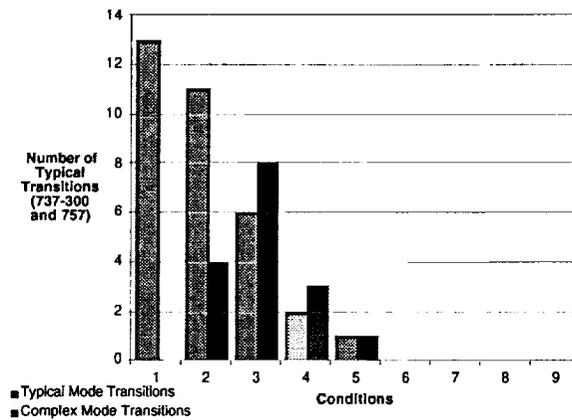
Figure 4.9:   Number of Conditions in Typical Mode Transitions (n=31)

4.4.3   Number of Branches

Figure 4.10 shows the number of branches which appeared in analyzed transitions. The mean number of branches was 1.62, with a standard deviation of 1.01. It is also interesting that the mean value of the number of branches identified directly by pilots in Figure 4.5 is statistically similar to the number of paths (where paths are defined as branches + 1) identified via cyclomatic complexity in Figure 4.10, though the distribution is different.

Figure 4.11 shows the distribution of number of branches in B737-300/B757 typical transitions. The mean value is 0.67 and the standard deviation is 0.96. There are statistically 0.4 fewer branches in typical transitions than in those identified as complex by pilots.

105
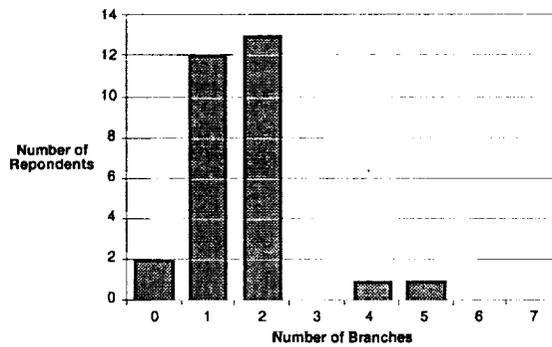
Figure 4.10:   Number of Branches in Identified Complex Transitions (n=29)
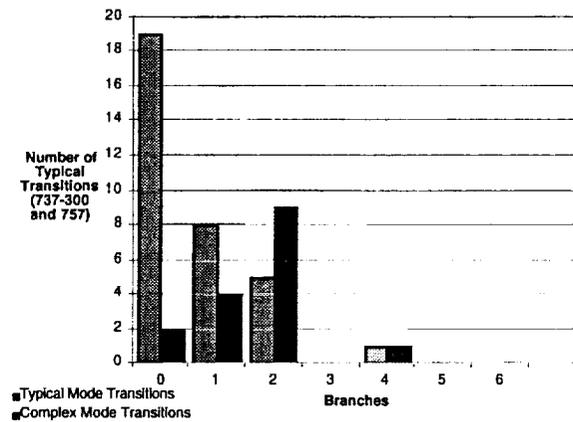
Figure 4.11:   Number of Branches in Typical Mode Transitions (n=33)

4.4.4   Number of Terminal States

Figure 4.12 shows the distribution of terminal states which appeared in analyzed transitions. The mean number of terminal states was 1.79, with a standard deviation of 0.56.

The distribution of the number of Terminal States in B737-300 and B757 typical modes is shown in Figure 4.13. The mean value is 1.24 and the standard deviation is 0.56. There are statistically 0.1 fewer terminal states in typical transitions than in those identified as complex by pilots.
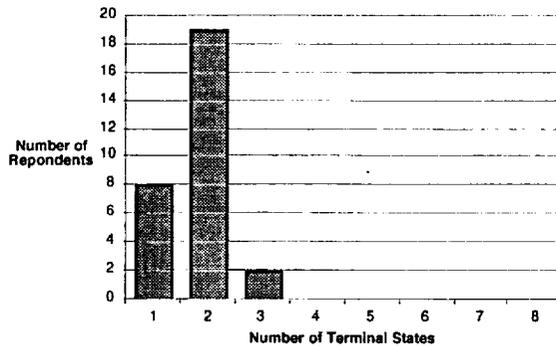
Figure 4.12:   Number of Terminal States in Identified Complex Transitions (n=29)
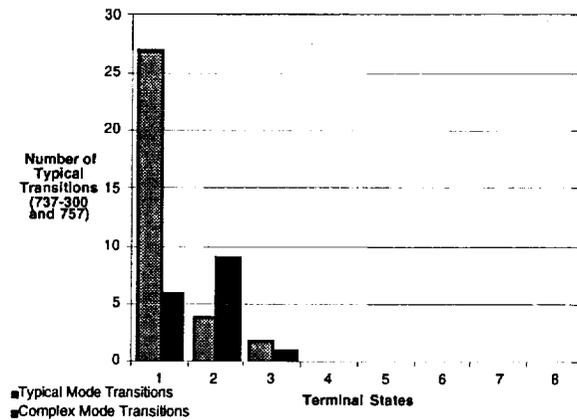
Figure 4.13:   Number of Terminal States in Typical Mode Transitions (n=33)

4.4.5   Pilot Ratings of Identified Complex Transitions

Pilots were also asked to rate the complexity of the identified mode transitions against two common mode transitions with low cyclomatic complexity: Flight Level Change to Altitude Hold and LNAV to Heading Select.

Figure 4.14 shows that the majority of pilots felt that the mode transitions that they identified were more complicated than the transition from Flight Level Change to Altitude Hold. The latter transition has a cyclomatic complexity of 5. Figure 4.15 shows that pilots also felt that the mode transitions that they identified were more complicated than the transition from LNAV to Heading Select. This transition is predicated on a single switch and has a cyclomatic complexity of 2. In addition, the responses to this rating were found to be statistically different that the results shown

107
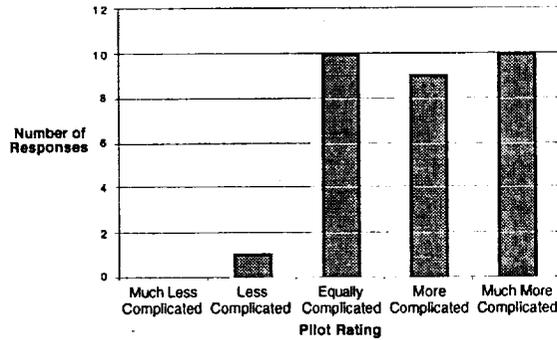
Figure 4.14:   Pilot Ratings of Identified Complex Transitions versus Flight Level Change to
                Altitude Hold ($v = 5$)

in Figure 4.14; pilots rated the identified transition as having a greater difference in complexity from the LNAV to Heading Select than from the Flight Level Change to Altitude Hold.
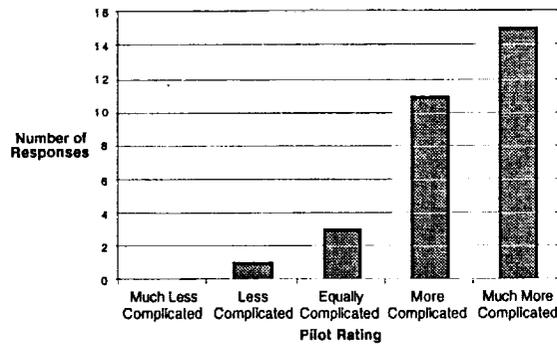
Figure 4.15:   Pilot Ratings of Identified Complex Transitions versus LNAV to Heading Select
                ($v = 2$)

These results were examined in further detail to examine the differences of those transitions which pilots identified as "Equally Complicated" versus "More Complicated" versus "Much More Complicated." Figure 4.16 shows the results of the comparison of the identified complex transition to the Flight Level Change to Altitude Hold transition for both the aggregate cyclomatic complexity and each constituent component: the number of conditions, branches and terminal modes. Figure 4.17 shows the same results for the comparison between the identified complex transition to the LNAV to Heading Select.

108
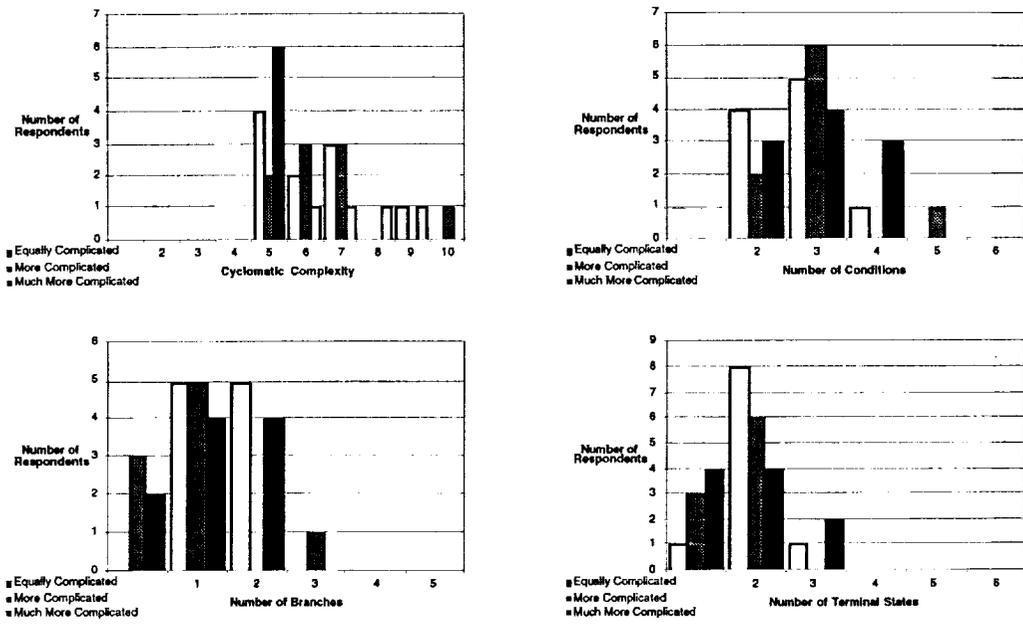
Figure 4.16: Detailed Analysis of Comparison of Identified Complex Transitions to Flight Level Change to Altitude Hold
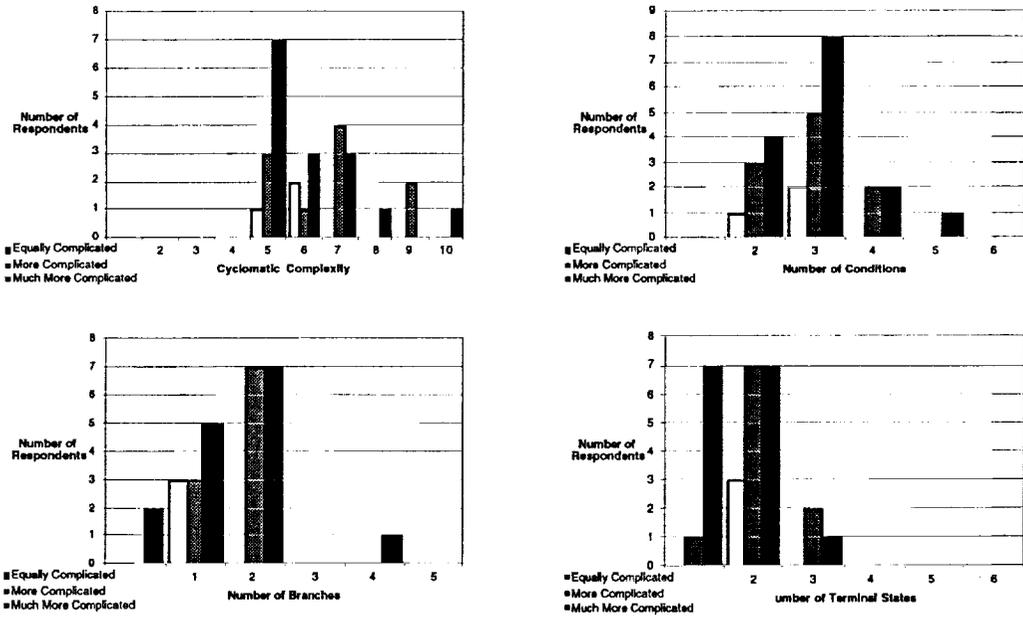
Figure 4.17: Detailed Analysis of Comparison of Identified Complex Transitions to LNAV to Heading Select

Both of the charts and the related statistical analysis do not show a relationship between the cyclomatic complexity of the identified complex modes and the subjective ratings of the pilots. It is hypothesized that this is due to the cyclomatic complexity measures being based (for consistency) on the contents of the Operators' Manual rather than of the pilots' own representations of the transitions. The survey was not designed to gain insight into the models of the automation being used by individual pilots. Instead, the Operators' Manual was used as an surrogate for the models being used by all pilots. For the aggregate results shown in Figure 4.14 and Figure 4.15 this was sufficient, but the inconclusive nature of the detailed analysis in Figure 4.16 and Figure 4.17 may be based on differences in pilots' models (further discussed in Section 3.5.5). It may be possible to investigate this level of detail through focused interviews rather than through a broad survey.

4.5   Chapter Summary

A survey was performed to analyze the cyclomatic complexity of those transitions which pilots characterize as complex. While not exhaustive or conclusive, the results from the survey shows two relevant relationships. The first is that mode transitions which were identified by pilots had a higher mean cyclomatic complexity than a typical set of modes from the B737-300 and the B757. In addition, pairwise comparison results demonstrated that pilots found that mode transitions with higher cyclomatic complexity were more complicated that those with lower cyclomatic complexity.

These results suggest that there may be correlation between the cyclomatic complexity of a mode transition and whether it will be considered complex from the perspective of the pilot. It appears that measuring the cyclomatic complexity of autoflight mode transitions can provide insight into whether these transitions will prove to be operationally problematic for pilots. Detailed pairwise comparisons did not show statistical differences between modes which were compared.

Chapter 5

Analysis of Autoflight System Issues

An examination was made of a cross-section of current autoflight systems with the goal of identifying and analyzing automation problems. This analysis included the ASRS review shown in Figure 1.3, focused interviews with pilot and designers, and an examination of accident reviews. A detailed examination was also done of operators' manuals from multiple aircraft which provided the basis for an analysis of inferred autoflight system behaviour and structure. This inferred structure became the basis of abstractions of the autoflight system, including the Hybrid Automation Representation discussed in the previous chapter. Where applicable, high fidelity simulator tests were run to verify behaviour predicted by the abstractions. Based on this analysis, this chapter will examine problems which have been identified in contemporary autoflight systems, and scrutinize those issues, where appropriate, using the Hybrid Automation Representation.

The examples used in this section are based on transitions to and from the Altitude Capture mode: transitioning from a climb or a descent into level flight. This transition is difficult because reconfigures the aircraft in a short amount of time, typically using Multi-Input, Multi-Output control for a smooth, low acceleration transition. As such, it appeared most frequently as a mode transition perceived as complex by pilots in the survey discussed in Chapter 4. This Altitude Capture Mode was also identified by pilots as complex across a broad set of aircraft: B737, B747-400, B757/767, B777, Lear 3X/6X, ATR-72, A300-600, A320, MD-80/88. For a subset of aircraft (B737-300-800, B757/767, Airbus A320) the behaviour of this mode was further probed through simulator testing. The final section in this chapter will be an examination of transitions to and from Altitude Capture Mode.

5.1 Limitations of Physical Systems

During the design of an aircraft system, physical limitations on the electrical, mechanical, and hydraulic systems may have become apparent. Many of these limitations were related to the dynamics of the system: spool up times for the engines, the motion of control surfaces, and the

finite durations required for physical changes to be made to the aircraft configuration. Control algorithms were limited by the finite bandwidth of the physical system and mechanisms were put in place to prevent unusual behaviours which could lead to instability. Aircraft utilizing the first and second generation of autoflight system have mechanical interlocks and relays to prevent actions which conflicted with these physical limitations. Newer aircraft tend to capture these limitations in their avionics.

5.1.1 Windshear Alert Suppression

An example of this is the suppression of windshear alerts while flaps are in motion in some windshear detection and warning avionics systems. Airflow over the wings during flap reconfiguration is liable to generate false alarm. The response to this condition was to suppress any alerts during this time, under the implicit assumption that alerts would be false. After the flaps have been reconfigured, the detection system would once again become active. The delay introduced by this design decision was identified as a contributory factor in the crash of a DC-9-31 at the Charlotte/Douglas airport on July 2, 1994 (Phillips, 1994). In this accident, the flaps were being deployed from 15° to 40°, which requires 10-12 seconds. Severe windshear was experienced at 275 feet, but the alerting system suppressed the warning for 7 seconds.

5.1.2 Altitude Capture "Linger Timer"

Timing issues are particularly susceptible to physical limitations, and these issues are hypothesized to appear in the interface between digital systems and dynamics. As such, some elements of interface design may be constrained by the limitation on timing. In the Boeing B737-300/400, a mechanism has been put in place to prevent immediate altitude changes. When an altitude change is made, a timer is started which runs for 2 seconds. After the timer has expired, the new altitude target becomes active. This is hypothesized to be in place to prevent the aircraft from chasing transients and functions essentially as a low pass filter, removing high frequency input spikes. This filter serves to match the input from the pilot and autoflight system to the physical mechanics to prevent the autoflight equivalent of a pilot induced oscillation.

Figure 5.1 shows a representation of the behaviour of the B737 during descent or climb while in Vertical Speed mode. The lower section of the figure shows a representation of the "Linger
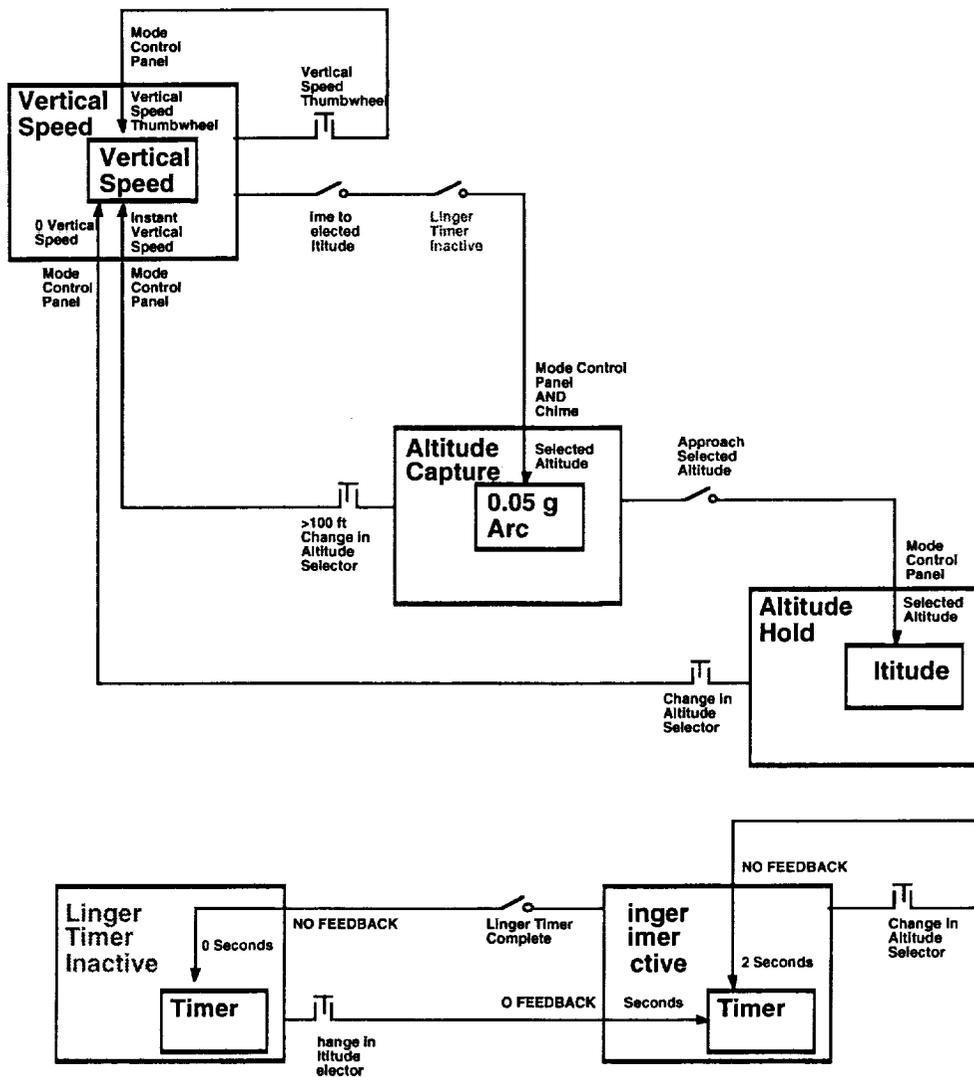
Figure 5.1:   B737 Altitude Capture Mode Transition Diagram

Timer" mechanism described above. When an altitude change is made ("Change in Altitude Selector"), the linger timer becomes active for 2 seconds. After the 2 second duration has expired, the system reverts to the Linger Timer Inactive mode. The timer is reset for another 2 second duration if an altitude change is made while the timer is active.

At the top of Figure 5.1, the transition between Vertical Speed mode and Altitude Capture is predicated on the Linger Timer being inactive: while the Linger Timer is active, an Altitude

Capture cannot occur. This implies that the Altitude Selector must remain static ("Linger") for 2 seconds in order for capture to occur.

This Linger Timer appears to be consistent across the altitude capture autoflight system segments across multiple aircraft and has been verified in the MD-11, B737, B757/767, B777 and the A320. The duration of the timer is not consistent across aircraft and is not documented in the training material, but the fundamental consistency is that a change in armed altitude becomes active after short duration. Simulator testing was required in order to document this behaviour. It is hypothesized that this may be used to maintain controller stability.

In typical transitions, this window of opportunity is not an issue. However, if an altitude change is made close to the time when the aircraft is able to transition to capture mode, it is possible for the automation to respond in unexpected ways. In particular, if the altitude knob is kept in motion, the Linger Timer cannot reset to allow capture to occur. Different aircraft respond in differing manners to the Linger Timer, when transitioning from Flight Level Change to Altitude Capture Mode. In the B737, if the new target altitude is above the aircraft during descent (or below the aircraft during climb) the automation will immediately attempt to attain the new target, however, if it is below the aircraft during descent (or above the aircraft during climb), capture will not occur and the aircraft will travel through the target. In the B757 the aircraft continued on its flight path while the altitude knob was in motion and did not capture. However, when the altitude knob was stopped, the automation attempted to capture the new target altitude.
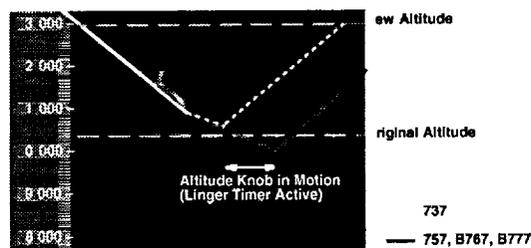
Figure 5.2:   Altitude Capture "Linger Timer"

Figure 5.2 shows this behaviour. The aircraft starts in a Flight Level Change descent towards a target altitude. While the altitude knob is in motion to the new altitude target above the aircraft

the B757/767/777, the aircraft continued in a descent whereas the B737 immediately begins to climb.

Low Altitude Linger Timer Concerns

A final concern with the linger timer is at low altitude when a Missed Approach Maneuver is being attempted. Prior to this attempt, some procedures required placing the target altitude on the runway. If the decision is made to do a missed approach, the target altitude is moved to above the aircraft (to the missed approach altitude) and the aircraft is placed into a climb configuration. It is possible, if the altitude knob lingers at a low altitude, that the aircraft could level at an intermediate, and inappropriate altitude during this maneuver.
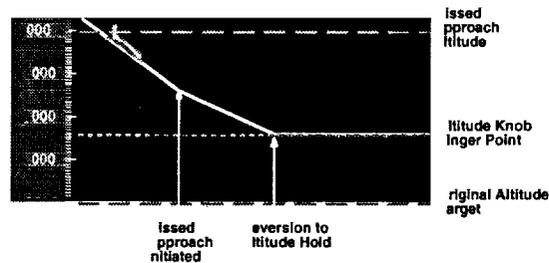
Figure 5.3:   Inappropriate Reversion to Altitude Hold

Figure 5.3 shows this scenario. The aircraft starts the approach with the altitude target on the airfield and then begins to execute a missed approach. As the altitude target is reset, it lingers at some intermediate altitude resulting in an altitude capture and a transition to Altitude Hold.

5.1.3   MD-11 Changes in Vertical Rate

As will be discussed in Section 5.3.2, the MD-11 responds to changes in the vertical rate target by suppressing the capture of a new altitude: motion of the pitch wheel during altitude capture reverts to Vertical Speed for short duration. This may be due to capture controller limitations, rather than a specific design goal.

5.2   Representation Issues

A review was conducted of existing flight automation systems which suggested that the evolutionary development of autoflight systems has resulted in a large and complex structure of

operating modes which do not appear to have a simple, consistent, underlying global model (Vakil, 1996). Research was done through public information sources, such as aircraft manuals, focused interviews with line pilots and check airmen and direct contact with avionics manufacturers (American Airlines 1997; American Airlines 1994; Boeing 1989; Honeywell 1992). No consistent global model of autoflight systems could be identified for the B727, B737, B757, B767, B747-400, A320, A300, MD-11 and F-100. Avionics manufacturers who were contacted were not able to supply a functional model or logic/control diagram. The documentation presented to the FAA is a detailed specification of the implementation of the automation, but not an overall model (FAA, 1996)

The Hybrid Automation Representation captures both the continuous behaviour defined by closed loop control and discrete behaviour defined as transitions between active controllers, but in a hybrid manner. The structure of the model is also based on the *de facto* standard of separation between closed loop control and transitions between modes which appears to be prevalent among the systems reviews. The HAR is not expected to be an appropriate model to provide to pilots, but more as a design tool during development.

Norman (1989) discusses the relationship between the user model, the design model, and the system image, shown in Figure 5.4. The design model is the designer's conceptual model developed during the creation of the system. The user's model is a mental representation created by the user through interaction with the actual system. The system image is the instantiation of the actual physical object and includes the documentation, instructions, labels and so on. All communication between the designer and the user occurs through the system image. An unclear system image—one not including some elements of the design model—will result in an incorrect user model. It is hypothesized that there is no articulated system model in autoflight systems, which leads to difficulty in the creation of the user's model.

In addition, if the design model is not captured in the system image, successive designers will be relegated to the position of generating their own mental representation and become equivalent to users. This relationship implies that the lack of a consistent global representation is a concern not only for the users of the system, the pilots, but also for the designers. The issue is that only the system image, and the underlying implementation details are available to ascertain the purpose of

design decisions and to document the instantiation of design goals. This documentation, termed "rationale capture" (Leveson, 1998), becomes much more difficult without a representation which can be used by designers.
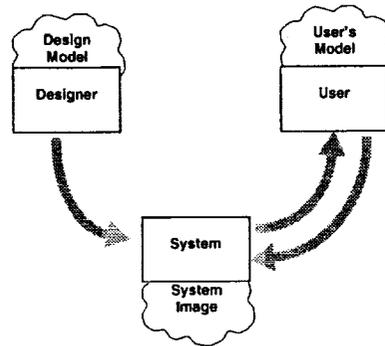
Figure 5.4:   System Image versus Design Model and User's Model (adapted from Norman, 1988)

The lack of a global representation follows from the incremental development of aircraft systems. While original systems may have had a straightforward representation, the consistency of these representations may have been degraded as changes and additions to automation were made by parties unaware of the original representation and consistencies. Part of the goal of capturing the rationale of the system is to make explicit and to widely distribute the underlying consistencies and goals of a system so that they may be maintained. Work has been done (Lehman, 1980) which tracks the development of systems, namely computer operating systems (Brooks, 1975) and telephone switching software (Lehman, 1996). This work has shown that over time systems grow away from a straightforward model and begin to become less consistent. As this occurs, the effort required to add additional capabilities onto the system increases. Without a clear model of the system, each change will increase the entropy of the overall system and increase the cost, time and complexity of successive changes as the design becomes more complicated.

From the standpoint of systems engineering design, this incrementally developed complexity is a costly problem. From the standpoint of a pilot, it is potentially a much more serious issue as it may undermine the ability to effectively monitor automation.

117

5.2.1 Concerns with Mental Models

In the absence of a simple, consistent and communicable model of flight automation, pilots appear to create their own models of the flight automation (Norman, 1988). These ad-hoc mental models have several shortcomings. The most obvious is that the models may not accurately reflect the actual systems. The basis of these models is grounded in both training material provided to the pilots and flight experience. The existing training material is based on a proceduralized, operational model with little causality or connection to the structure of the underlying system. In some cases it has also been shown to be incomplete. Compared to other automation systems, clear mental models of time-critical flight systems are of particular importance. In current aircraft automation, the pilot is given final control and full responsibility. This implies that the pilot must understand, at some level, all automation behaviour in order to intervene effectively and appropriately in emergency situations. It may be the case that a limiting factor on aircraft automation design should be the level of complexity that a pilot can maintain and readily access as a mental model.

Experientially Developed Models

It is hypothesized that pilots develop mental models based on experience interacting with the system. It is expected that the actual mental models used by the pilots are more sophisticated than those put forward during training and are a function of their individual pilot backgrounds. Since these models are created independently by individual pilots, specific ad-hoc models may not be accurate. In addition, the fact that ad-hoc models are created during nominal operations (where the vast majority of pilot experience occurs), they may not hold (or may even be a liability) in emergency situations.

Javaux (1999) has put forward a mechanism based on spreading activation networks which accounts for the development of mental models during nominal conditions. *Frequential Simplification* and *Inferential Simplification* are mechanisms by which the details of a sophisticated transition between modes can be reduced to a less comprehensive prototypical state. Frequential simplification occurs when a sophisticated interaction is simplified during nominal operation by having many of the conditional elements satisfied. Inferential simplification creates an analytical basis for the incorrect application of consistency, when a pattern which appears in

118

some portion of the automation is incorrectly applied to another segment. It is difficult to distinguish which of these mechanisms is in effect, since they are related phenomena. Two examples are presented where the simplified model of aircraft automation is suspected to have been a contributory factor in an aircraft incident.

> At the accident which occurred at Nagoya on an Airbus 300, the crew inadvertently put the aircraft into a go-around mode. The autopilot on the aircraft can be disengaged by forcing forward on the yoke with sufficient force. However, the automation is not completely disengaged and maintains control over the trim of the horizontal stabilizer. The autopilot is programmed to trim the horizontal stabilizer to the flight path of the engaged mode. In the case of the go-around mode, the autopilot trims the horizontal stabilizer to a nose up configuration to allow the aircraft to climb. The crew, believing that they had disengaged go-around, pushed the aircraft nose down to continue the approach. When the crew decided to abort the approach, the aircraft pitched up violently due to the combination of horizontal stabilizer trim and elevator input, stalled and crashed.
>
> ...
>
> In advanced Boeing aircraft, in general, transitions between modes can be initiated by pressing buttons on the Mode Control Panel which select the new mode: pressing the ALT HLD button initiates the Altitude Hold mode. However, the Approach Mode cannot be transitioned out of without switching off the autopilot entirely. Attempting to engage a new mode via the Mode Control Panel is ignored. While this prevents the accidental engagement of a spurious mode during a critical flight segment, the accidental engagement of Approach can lead to serious confusion since the automation will suddenly appear to be nonresponsive to switching modes. (Vakil, 1998)

Cognitive scientists have shown that humans' understanding of the world consists of identifying and explaining patterns (Richards, 1998). These patterns range from consistencies in the behaviour of tools to seeing images in clouds. In situations where no causal basis for an underlying pattern has been presented, humans will grasp onto any apparent pattern in an attempt to explain behaviour. In the absence of some basis on which to identify and take advantage of patterns in automation, incorrect mental models may arise.

5.2.2 Rare modes

A closely related issue is pilot interaction with modes which are rarely used or modes which are used regularly but for short duration. These modes may not be used sufficiently to populate

whatever mental representation the pilot is using. In the case of modes of limited duration, it is unlikely that there is time to consider or model the effects of inputs to the system, simply because little time exists to provide these inputs (Palmer, 1996).

In a larger sense, this problem exists in many aspects of aviation. Much of standard and recurrent training deals with situations which are rare: engine fires, tire blowouts, go-arounds etc. These are dealt with by creating and reinforcing procedural behaviours during simulator training. Similar training and effort is not spent reinforcing models of the automation itself.

Using Mode Transition Matrices to Identify Rare Modes and Transitions

Mode transition matrices can be used to help in organizing and identifying which modes and which transitions suffer from an insufficient experiential basis. Since the mental representations used by pilots appear to be experientially based, the most benefit may be gained by tailoring the automation training to individuals. This may be possible through access to pilot-specific data from data recorders.

The mode transition matrices discussed in Section 3.3 included all possible transitions between modes. These can be termed "Allowable Transition Matrices." In some cases, it may be useful to examine the converse of the allowable-transition matrix. By doing so, the criteria for not allowing access to specific behaviours can be examined. This may be useful in situations where a mode change could lead to a dangerous aircraft configuration. The information in mode transition matrices can also be used to organize information from a modal system. Examples include allowable mode transition matrices, frequency-based transition matrices, relative frequency based, and rarely-used transition matrices. In addition, specific attributes can be highlighted with the mode transition representation, such as the nature of conditional statements or the available feedback.

Table 5.1 is a mode transition matrix adapted from work by Degani (1994) observing mode transitions in operational, revenue-earning flights of a Boeing 757. In this matrix, the cells are populated with the absolute frequency of observed transitions. In contrast with earlier examples, the aircraft autoflight modes include both vertical and lateral behaviours, so that a single mode is

defined as the behaviour of the system in both axes. Mode transitions which were observed with a frequency of less than 2% are not included on the diagram.

	Takeoff Heading Hold	VNAV Heading Hold	VNAV Heading Select	VNAV Spd Int LNAV	VNAV LNAV	VNAV One Now LNAV	Vertical Speed LNAV	Flight Level Ch LNAV	Flight Level Ch Heading Select LNAV	Altitude Hold LNAV	Altitude Hold Heading Select	Flight Level Ch Localizer	Glide Slope Localizer
akeoff eeding Hold	%	%											
NAV eeding Hold		%											
NAV eeding Select			1%										
NAV Spd Int NAV				%									
NAV NAV		%	%		%	%	%	%	%	%			
NAV One Now NAV				%									
ertical Speed NAV							%						
light Level Ch NAV				%				%					
light Level Ch eeding Select							%			%	%	%	
ltitude Hold NAV							%						
ltitude Hold eeding Select							%						%
light Level Ch ocalizer													
lide Slope ocalizer													

Table 5.1: Observed Autoflight Transition Matrix showing Absolute Percentage of Transitions

There are several insights to be gained from the matrix. The first is that certain modes were only observed to transition to a single other mode. For example VNAV-Heading Hold was only observed to transition to VNAV-Heading Select. This does not imply that there is only allowable transition from VNAV-Heading Hold, but does highlight how the modes are used operationally. Since the matrix is based on a finite set of data, all possible transitions were unlikely to be observed. However, by looking at the frequency data, the nature of the mode transition matrix can be seen to be dependent on the operating policies and training characteristics of the particular airline. If data across airlines can be examined, a nominal usage pattern of mode transitions may be apparent.

Other modes have no observed exit transitions at all. The Flight Level Change-Localizer and Glide Slope-Localizer Modes are terminal modes, as can be seen from the empty row in the transition matrix. While there are allowable transitions from these modes (usually to Go Around Mode) they were not observed. Large blank areas can be observed in the matrix, corresponding to sets of transitions which were not observed. In many instances, this is due to certain modes being

highly specialized and so of limited usage. An example is Takeoff-Heading Hold which only used during takeoff and climbout and not later in flight.

Columns and rows which are highly populated represent the modes in which the aircraft automation spends the bulk of its time. In this data VNAV-LNAV has a large number of transitions both in and out. In particular, note the most frequent transition between VNAV-Heading Select and VNAV-LNAV. This is hypothesized to be due to vector-based air traffic control transitioning to flying a predetermined flightplan.

Unused Mode Transition Matrices

Another potential use of mode transition matrices is to highlight the set of mode transitions which are used infrequently by examining the difference between a normalized allowable matrix and a frequency matrix. The difference of these two matrices will be a measure of the infrequency of transitions: those which can occur, but do not appear during normal operation.
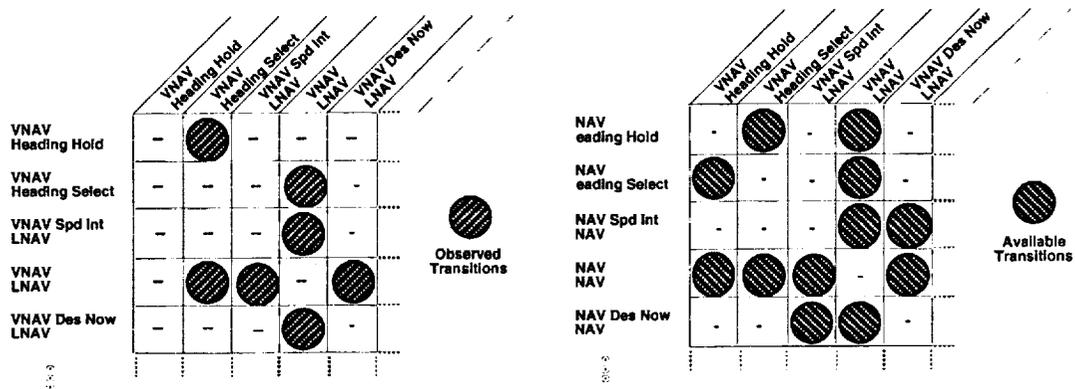
Figure 5.5:   Partial List of Observed and Allowed Transitions

The left matrix in Figure 5.6 shows a subset of the modes observed in Table 5.1 and indicates the transitions which were observed. The matrix on the right show those modes which are allowable. The difference between these two matrices is shown in Figure 5.6. The transitions which are marked in black indicate transitions which are allowed, but which were not observed in operation. These mode transitions are of concern because they indicate non-nominal situations with which a pilot may have little experience. If these transitions are determined to no longer be necessary, this information can be used to guide the simplification of subsequent generations of

automation systems. It may also be the case that the transitions are necessary (such as transitions to a Go Around Mode), but are operationally infrequent. Additional training can be brought to bear on these transitions.
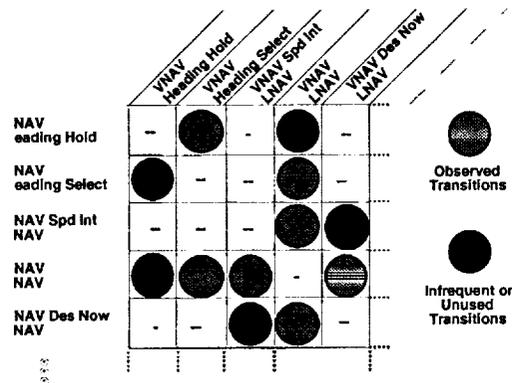
Figure 5.6:   Partial List of Infrequently or Unused Transition Matrix

5.2.3   Lack of Consistency and Predictability

Consistency in autoflight systems refers to how accurately a pattern in interaction or behaviour can be applied to other areas of the automation. An example of a pattern is that pressing the Altitude Hold button while in a Vertical Speed ascent results in the aircraft immediately leveling off and holding the current altitude. If this pattern can be applied in all ascents, or even more generally to all altitude changes, the system has a high level of consistency. In practice, the B737 ignores the Altitude Hold button being pressed while in the glide slope is being actively tracked, undermining the consistency of the system.

Consistency is an issue for the underlying structure of the automation, which can be identified with the Hybrid Automation Representation, but also for the training material. If consistencies are not communicated to pilots, or provide an insufficiently complete representation of the automation to allow the consistency to be apparent, the pilots may not be capable of exploiting the consistency as a means of complexity management.

Identifying Consistency Using the Hybrid Automation Representation

The Hybrid Automation Representation can help in identifying consistent behaviours, and exceptions where inconsistency appears. In particular, the conditions which are used to fully

123

specify a transition, and the new target values which are used after the transitions are appropriate and useful to examine for consistency. Figure 5.7 shows a simple example of how the Hybrid Automation Representation can be used to examine the autoflight system for inconsistency. In this diagram, a number of modes are shown along with their transitions to the Altitude Capture and ultimately Altitude Hold mode. The majority of these modes can transition to capture immediately after the Altitude Hold button is pressed, or after the intended altitude is approached during a climb or descent.
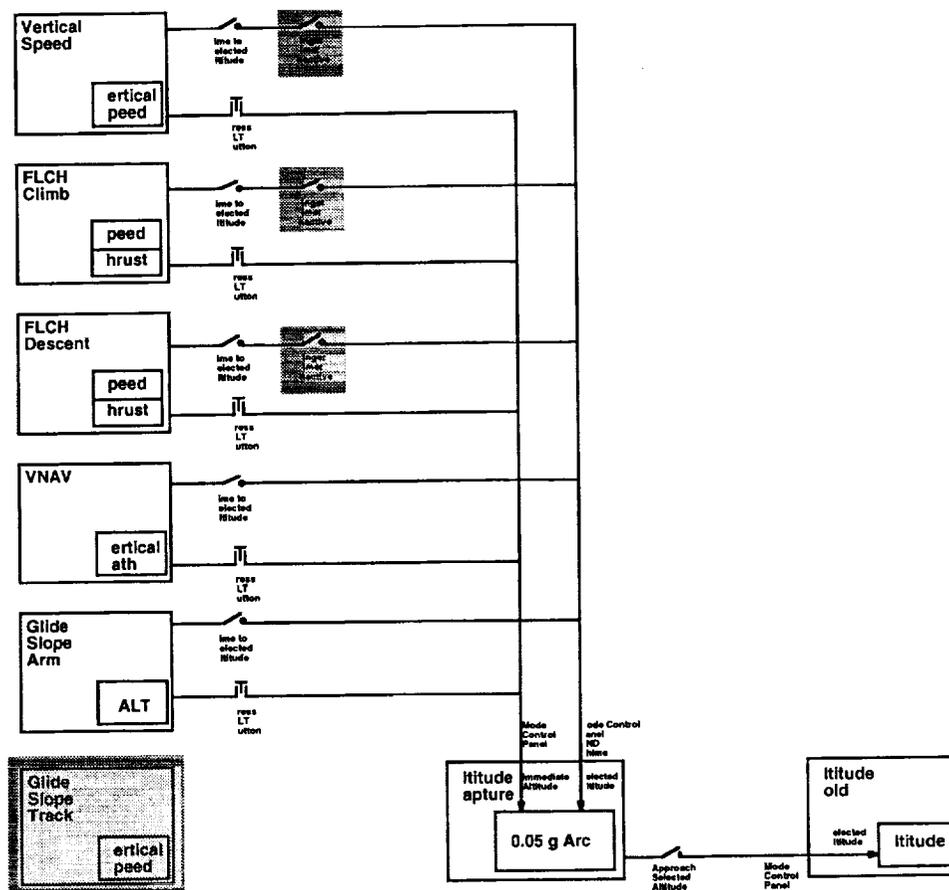
Figure 5.7:   Transitions to Altitude Capture and Altitude Hold Modes

Three sets of transitions can be seen in this figure. The first is an automatic transition which occurs between modes which are designed to be used in conjunction with Altitude Capture, namely Vertical Speed, FLCH Climb, and FLCH Descent. Each of these modes automatically

transition to Altitude Capture when the aircraft approaches the selected altitude and while the "Linger Timer" (highlighted in grey) discussed in Section 5.1 is inactive.

A transition can also occur immediately if the ALT button is pressed by the pilot. The second set of transitions are similar, but do not, based on the available documentation, appear to depend on the "Linger Timer." The transition between VNAV or Glide Slope Arm and Altitude Capture can also occur based on the pilot manually pressing the ALT button.

The final type of transaction is from the Glide Slope Track mode. No transition can be made to from Glide Slope Track to Altitude Capture. Once the aircraft has acquired the Glide Slope and is tracking it, neither approaching the selected altitude nor pressing the ALT button will result in a transition to Altitude Capture. This behaviour is inconsistent with the other transitions.

The diagram shown in Figure 5.7 appears to be a contrived example, in that it shows the inconsistency in a simplistic and immediate manner. However, this example is based on a single column in a Mode Transition Matrix. Each column consists of all of the possible transitions to a mode, similar to how each row consists of each transition from a mode. By examining individual columns and rows, consistency between modes can be made apparent.

Consistency can also be examined directly in the mode transition diagram, but it can be more difficult to identify. Figure 5.8 shows the diagram representing the Flight Level Change Behaviour of the Boeing B737. The inconsistency which can be noted here is that the Linger Timer is a condition in only certain cases. The transition from FLCH Climb/Descent to Altitude Capture is dependent on the state of the timer, but other transitions are not.

Predictability

As autoflight systems have become more capable, there has been a shift to pilots becoming supervisors and managers of automation. This shift towards supervisory flight automation has caused the ability to accurately predict the future state of the automation to become critical in monitoring system conformance. The task of monitoring a system consists of noting and diagnosing differences between the observed behaviour of the system and the behaviour that was expected.
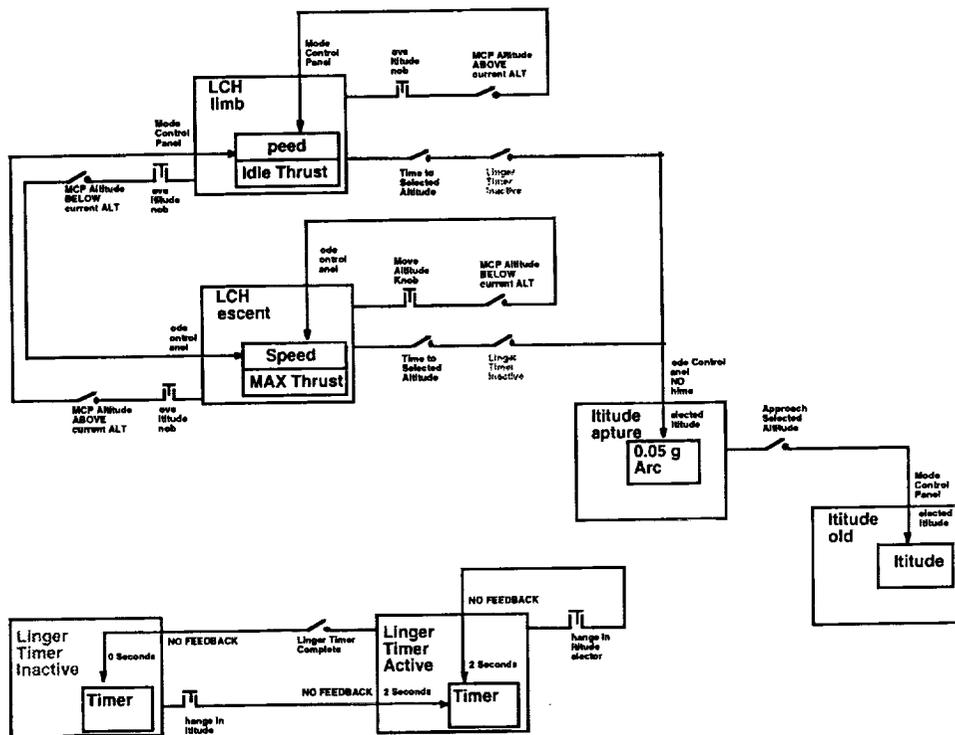
Figure 5.8:   Linger Timer Consistency in B737 During Flight Level Change

The concept of predictability is defined as a measure of how well an operator can anticipate what the system will do at some point in the future. In essence, this is a measure of the complement of how often a system will "surprise" an operator by acting in an unanticipated manner. This concept correlates to consistency: systems which are highly consistent are expected to have a higher level of predictability than those which are not. A preliminary experiment was run to examine the difference in accuracy of prediction between subjects who were using a system with a consistent model (a Reverse Polish Notation calculator) and one which had a less consistent model (a standard "Four Function" calculator). It was found that a consistent model had a statistically more predictable behaviour. The full details of this experiment have been published (Vakil, 1997) and are shown in Appendix B.

The Boeing B737 has an example of poor predictability in the behaviour of the autoflight system to a change in MCP altitude during a Vertical Speed descent. The response of the aircraft to a change in the altitude target is difficult to predict because it is dependent on the rate at which the altitude knob is moved. Figure 5.9 shows the Hybrid Automation Representation of this

126

segment of the autoflight system. Consider a scenario where the aircraft is descending from FL330 in Vertical Speed at -4000 fpm, the MCP altitude is set at FL290 and the aircraft currently at FL310. If the altitude knob is moved quickly to FL350, the aircraft will end up in an open descent with the MCP altitude above it during descent. If the altitude knob is moved very slowly to FL350, the aircraft will capture the current altitude (near FL310) and level off. If the altitude knob is moved at an intermediate, but slow, rate or if the pilot pauses near the current altitude while moving the knob, the aircraft will transition to Altitude Capture and then revert to Vertical Speed with the immediate vertical rate as the target. This will result in a more shallow open descent.
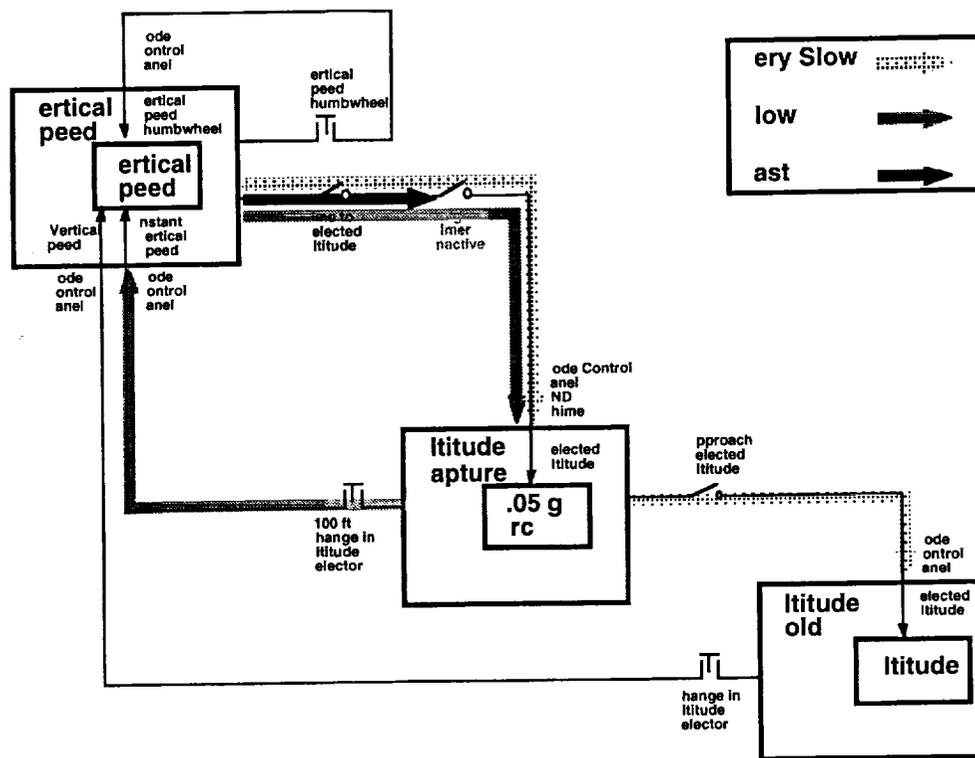
Figure 5.9:   B737 Altitude Capture Mode Transition Diagram

These three behaviours can be seen in the Mode Transition Diagram of the B737 Altitude Capture. If the altitude knob is moved quickly, the transition to the Altitude Capture mode will be suppressed by the Linger Timer (discussed in detail later), which does not allow propagation of changes to the altitude target to occur quickly. If the altitude knob is moved very slowly, the

condition of the time to the selected altitude will be satisfied when the MCP altitude is near the current altitude of the aircraft. A transition will occur to the Altitude Capture mode and then to the Altitude Hold mode. Finally, in the intermediate case, the transition will occur to the Altitude Capture mode, but continued motion will result in a transition back to the Vertical Speed mode with the instantaneous vertical rate becoming the new target. Figure 5.10 shows the trajectory of the aircraft in each of these cases.
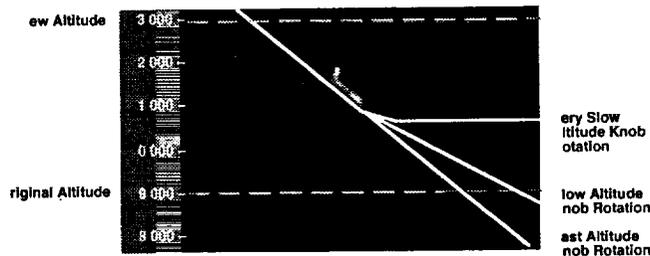
Figure 5.10:   Vertical Trajectory of B737 After Change in Altitude Target

Shift Towards Supervisory Systems

In the case of manual control systems, even those which have been hydraulically or mechanically amplified, the possible behaviours of the system are well understood. Pulling back on the yoke will cause the nose of the aircraft to pitch up. The mapping between the control input and the aircraft response it may become part of a repertoire of skills, rather than knowledge or rules. The understanding required to predict the behaviour of the aircraft in the presence of control inputs and the larger environment includes an understanding of aerodynamics and practice with manual piloting skills. The physics of the situation define the capabilities of the system.

In systems which utilize automation for aircraft control, monitoring requires the ability to track the evolution of the state of the automation in addition to the state of the aircraft dynamics. Since automation can directly control the behaviour of the aircraft, the state of the automation needs to be understood in order to predict the future aircraft state and to detect when it is inconsistent with what was intended or expected. The mental model required for this task is a superset of that required for manual piloting, since it must include some representation of the automation itself in order to monitor conformance. Rasmussen (1986) discusses the capability necessary in order to effectively interact at a knowledge-based level. At this level of

understanding, the pilot must have a model of the automation which can be evaluated in order to predict future aircraft states. Multiple possible outcomes are generated based on this predictive analysis and the most likely of these outcomes is selected. A key element of this process is an accurate and complete model of the automation to be used by the pilot. It is precisely this information which appears to be lacking in flight crew operations manuals and other training and documentation.

5.2.4 Discrepancy Between Pilots' and Designers' Representations

The lack of a global model of the automation may result in discrepancies between the representation of the automation used by pilots and those used by engineers. These discrepancies may arise due to the different manners in which pilots and engineers interact with automation. Pilots typically have an operational viewpoint and consider the automation within the context of procedures and air traffic control. Their parsing of the automation appears to be based on the functionality and the immediate interface as shown on the right side of Figure 5.11. Each segment of the autoflight system is considered to be a separate system: fly-by-wire system, stability augmentation, autopilots and autothrottle, and the flight management system. By contrast, engineers parse the autoflight system into control loops based on the closed loop control being provided. These control loops are defined by the kinematics and dynamics of the aircraft. Engineers are also less likely to have operational insights into the usage of the automation. These differences may contribute to the likelihood of incidents occurring, particularly if each party has differing expectations of the automation based on their viewpoint. Consistencies in one viewpoint may not be consistencies in the other.
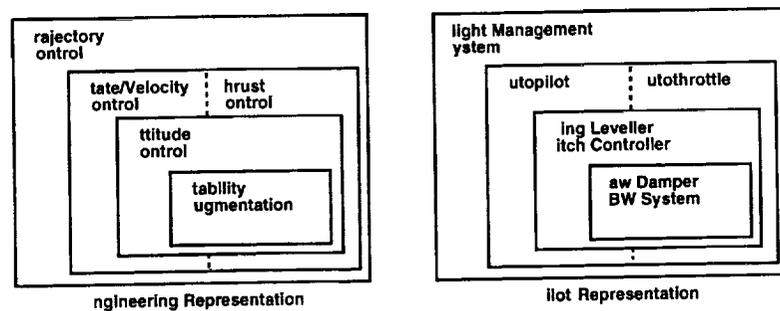
Figure 5.11:   Hypothetical Differing Representations of Autoflight Systems

Differing Expectations

The observed behaviour of automation may be judged by pilots to be inappropriate if it does not fit with the operational task being performed. This can either be due to the behaviour simply being poorly designed, or can occur when the automation behaviour is consistent with an engineering model rather than with the pilots' operational representation of the task.

One example is the 1994 incident in Orly, France, where the A310 to pitched up to acquire an altitude target late in an approach. The automation was designed to attempt to capture the current programmed altitude in the event of a speed violation. In this incident, the crew had programmed a target altitude at the Missed Approach Altitude, during final approach (as per operational guidelines). This target altitude was above the current altitude of the descending aircraft. When the flaps were inadvertently lowered at too high a speed, the AutoFlight System detected an overspeed and pitched the aircraft up in an attempt to capture the Missed Approach Altitude (Sparaco, 1994). In response to this incident, Airbus issued an advisory bulletin warning pilots to follow posted flap limit speeds carefully (Sparaco, 1994b).

The engineering representation of how to deal with overspeed events appears to be to switch to a speed-protected mode and fly towards the Flight Control Unit (FCU) altitude. There is an implicit assumption in place that the aircraft always flies towards the FCU altitude. Operationally, this assumption breaks down during approach, when the target altitude is placed at the initial missed approach altitude, above the descending aircraft. It is important to note that this behaviour appears to be consistent across multiple Airbus aircraft, including the A310, A320, A330, and A340.

The engineering representation of this system is shown in Figure 5.12. In this figure, the transition conditions from the Vertical Speed mode to the speed-protected Level Change (Climb) mode are shown to be an under- or overspeed condition when the altitude on the FCU is *above* the aircraft. Similarly, the transition conditions to the Level Change (Descent) mode are shown to be under- or overspeed condition when the altitude on the Flight Control Unit (FCU) is *below* the aircraft. This behaviour is shown in the top of Figure 5.12 as the white path of the aircraft. When
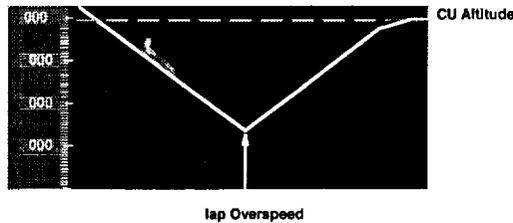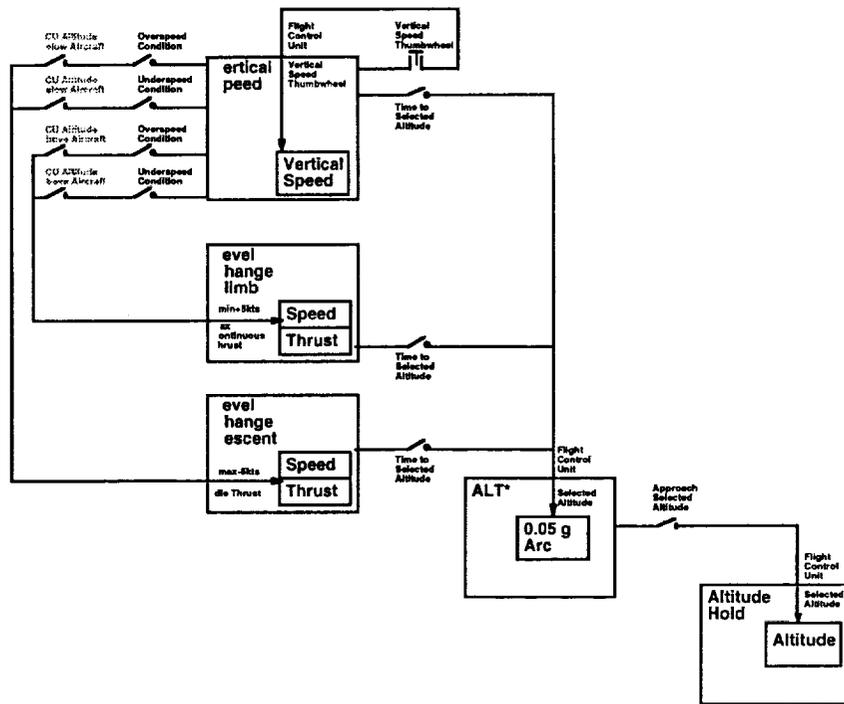
Figure 5.12:   Engineering Representation of A310 Envelope Protection

the overspeed condition occurred, the aircraft went into a climb mode to acquire the FCU Altitude.

By contrast, a Hybrid Automation Representation of the behavior expected by the pilot is shown in Figure 5.13. The finding that pilots expected this behaviour was determined during the evaluation of the Electronic Vertical Situation display where this incident was used as an experimental scenario (Section 6.2, Appendix E). While the structure of the model is identical, the key difference is that the conditions required for the transition to the speed-protected Level Change modes are expected to be based on the current vertical rate of the aircraft. If the aircraft is climbing, it is expected that a Climb mode will become active. If the aircraft is descending, it is

131
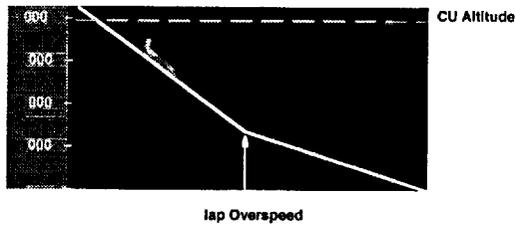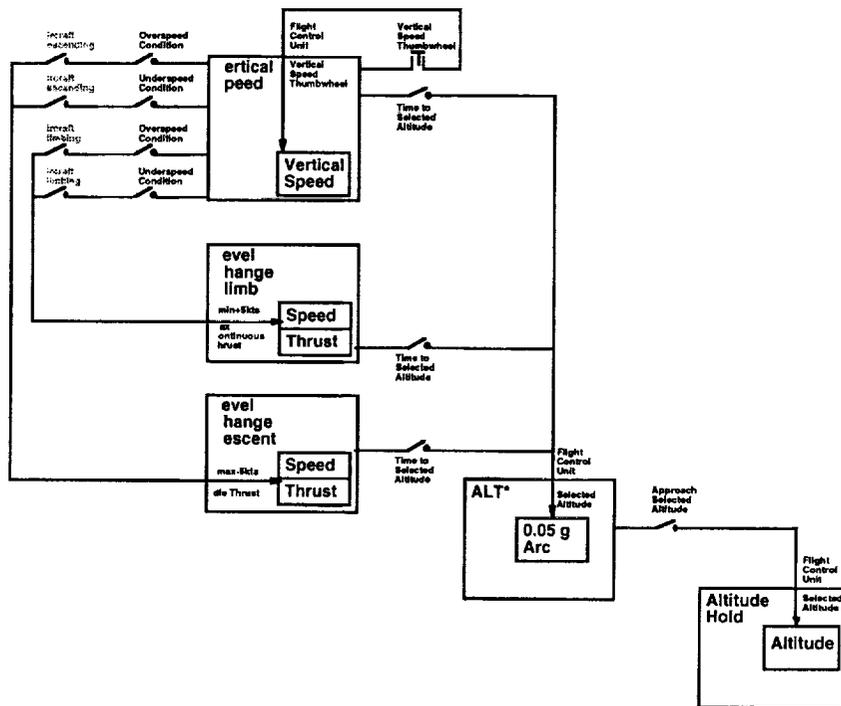
Figure 5.13:   Pilot: Expected A310 Response to Overspeed during Descent

expected that a Descent mode will become active. This behaviour is shown in the bottom section of Figure 5.13. In this diagram, after the mode change occurs due to the overspeed condition, the aircraft continues on its descent at a reduced rate. This behaviour is may be more operationally appropriate, since it allows the aircraft to continue in its trajectory at a reduced rate and does not have the drastic behaviour of the engineering approach. However, this can place the aircraft into an open descent mode while in Level Change, which is an unexpected behaviour since Level Change flies towards the FCU target in nominal operations.

Differing Perspective on Incident Reports

Work is currently underway into examining the different analyses that pilots and engineers use in describing system operation by investigating incident reports (Feary, 1999). This also serves as an indication of the different representations used by each party. Table 5.2 shows a comparison between these two analyses. Pilot descriptions consisted of longer, system-oriented descriptions with an operational perspective. By contrast, engineers presented shorter descriptions from an analytical system-specific perspective. From the structural standpoint, the engineers' report was a bullet list of individual elements as compared to the pilots' cohesive narrative.

Table 5.2:   Pilot versus Engineer Perspective on Incidents (Feary, 1999)

Pilot Response	Engineer's Response

Analysis of Narrative 310133
 	
This incident seems to be nothing specific to the MD-11. It deals with cockpit distractions, not making required callouts, and so forth. Perhaps there was some confusion with an altitude in the FMS versus and altitude sent in the GCP, but the distractions are the overriding factor here.
 	
To deal with the loss of the third set of eyes, automation has given us the capability to monitor the automatic flight. Essentially, we have two computer programmers monitoring the pilot, who is trained to follow behavior limited by very strict rules.
 	
The obvious method that the MD-11 was designed for in this situation was to program the crossing restriction in the FMS and allow PROF to make things happen. Since the copilot took PROF out of the loop, he changed the rules. Vertical speed will even override the GCP altitude if it is done during the level off. This mode requires constant attention of both pilots, and if there are cockpit distractions, it may not be an appropriate mode to use.
	Brief Description: Aircraft descends to 10,000 ft when FMS Flightplan has 11,000 ft constraint
	
	Summary:
	    Operator failed to keep track of the level of automation engaged.
	    Operator did not follow SOP
	
	What happened:
	    Descending with clearance to waypoint at 11,000 ft. Pilot entered 11, 000 ft constraint into the Flightplan
	    Aircraft descended to GCP alt = 10, 000 ft. Pilots missed 1000 ft callout
	
	Engineering explanation of behavior:
	
	The behavior of the aircraft may be explained in a number of ways
	
	1. PROF was not engaged (pilot pulled alt knob after selecting the altitude). The avionics, correctly, controlled the aircraft to the GCP altitude at 10,000 ft.
	
	2. PROF was engaged, but the aircraft was long (high) on the optimum pat and sequenced the waypoint with the 11,000 ft constraint above 11,000 ft. The aircraft continued the descent and leveled-off at the GCP Alt
	
	3. Failure of avionics—no evidence to corroborate this supposition

Similar research has been done outside of the aviation domain in the environment of photocopier repair (Orr, 1999). While appearing completely unrelated, the structure of the relationship between design engineers and repair workers is similar to that between design engineers and pilots. In each case, the engineers use different representations of the task and automation, and are isolated from the operational aspects of the tasks. In a similar manner to

133

aviation, Orr found that "war stories" were the means of information distribution by repair workers.

Insufficient Documentation

A final note on representations is that in the web survey pilots noted multiple instances where modes were inadequately documented or discussed in the Flight Crew Operators' Manual. Several cases were verified, both specific to particular aircraft and common across multiple aircraft.

In the B757/767, the transition between Takeoff and Altitude Hold at a low altitude can cause a reversion of the speed target to the current value, rather than maintaining the active value in the mode control panel. In addition, controlling around speed in Mach switches to controlling indicated airspeed when the pilot switches from Flight Level Change to Vertical Navigation. Neither of these conditions is outlines in the documentation. In the A320, Localizer Capture reverts to Heading Select whenever the heading knob is moved. In the B777, there is no indication in the manuals why a transition is effected from VNAV-Path to VNAV-Speed.

This is a serious issue, as it undermines the ability of pilots to build a complete mental model of the automation. In addition, it can seriously undermine the pilots trust in the documentation and cause them to revert to creating purely experiential mental models—and utilizing those experiential models at the expense of reading documentation.

5.3   Feedback and Lack of Observability

In the absence of appropriate and observable feedback, the pilot may not be able to anticipate the future behaviour of the automation. One of the issues which has been examined during this research has been an analysis of the "observability" of the automation. Observability is based upon the control definition and measures how well automation state can be measured from available feedback and indicators. From the viewpoint of mode transitions, this measures whether mode changes and behaviour changes are made apparent to the pilot. In some cases, mode changes may be made silently, though the behaviour of the aircraft may change. Alternately, the current active mode may not be apparent to the crew. An example of an incident which appears to

have had poor feedback and lack of observability as a contributory factor is the crash of an A320 at Strasbourg, France.

> On January 20, 1992, an A320 aircraft crashed during a non-precision approach into airport. The aircraft was estimated to be descending at 3300 fpm, a much steeper rate than the approach was designed for. It is speculated that the flight crew's intent was to descend on a flight path angle of 3.3°, which safely approximated the numerous level-off altitudes and short descent legs. Instead the aircraft was placed the aircraft into the wrong descent mode: Vertical Speed instead of Flight Path Angle. In Vertical Speed (VS) mode, the indication for 3300 fpm looks almost identical to a descent in Flight Path Angle (FPA) mode of 3.3°. The crew did not recognize the problem until too late and the much higher descent rate resulted in Controlled Flight Into Terrain (CFIT). Similar incidents have been reported in A320s at both San Diego and Gatwick. (FAA, 1996; Sparaco, 1995)

Autopilot modes and target are selected on the Flight Control Unit (FCU) located in the center console on the glare-shield, between the captain and the flight officer. Knobs are used to selected targets: the pilot sets the desired state value and then pulls on the knob to command the mode governing that state. To toggle Vertical Speed or Flight Path Angle modes, pilots use a recessed push-button. Distinguishing between Vertical Speed mode being active versus Flight Path Angle mode was difficult. The top two figures in Figure 5.14 (Pritchett, 1995) shows the FCU in each mode. The differences between each mode were in the identifier in the middle of the display and the decimal between the digits, or lack thereof.

This incident appears to have been caused by automation with poor observability caused by lack of appropriate feedback. A319s and newer A320s have been fitted with a modified target display which shows four digits for during a V/S mode and only two during FPA mode. Air carriers have also been given the option of retrofitting this new display into older cockpits. The bottom two figures in Figure 5.14 shows the updated displays.

5.3.1   Observability in the Vertical Domain

The feedback in the vertical domain is limited as compared to the horizontal domain. This is due to the lack of a vertical situation display and the increased complexity, as measured by the number of possible targets and modes, of the vertical domain. The moving map display provides excellent feedback in the horizontal domain, but there is no widely used analogous display in the

135

Original FCU in Vertical Speed Mode   Original FCU in Flight Path Angle Mode

Modified FCU in Vertical Speed Mode   Modified FCU in Flight Path Angle Mode
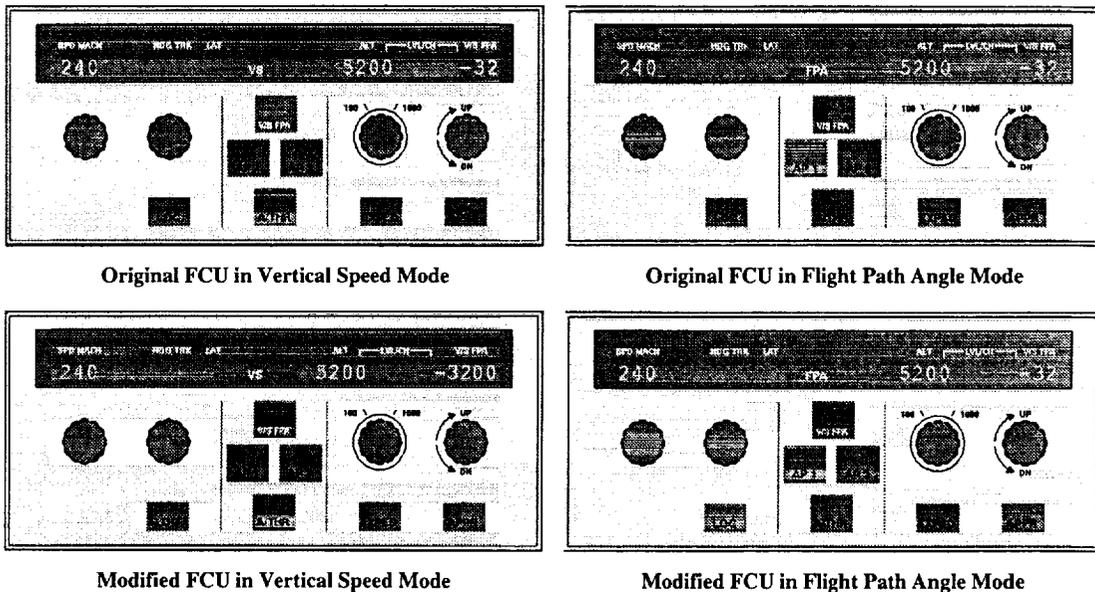
Figure 5.14:   Flight Control Unit Depiction of Active Mode (adapted from Pritchett, 1995)

vertical domain. The graphical feedback for the vertical domain is limited to some elements on the map display showing descent and climb initiation and completions.

The disparity in feedback is inconsistent with the complexity of each of these flight domains. The horizontal flight of the aircraft can be use either a track or a heading as a target and control to them by use of roll. Vertical is controlled by both pitch and thrust, with multiple potential target states: vertical speed, altitude, path, airspeed, thrust, pitch, angle of attack, flight path angle and others. The ASRS review, shown in Figure 1.3, highlights a preponderance of incidents in the vertical domain. In addition, examining the results of the web survey in Table 4.4, Table 4.5, and Table 4.6 shows that the majority of problematic mode transitions identified by pilots to be in the vertical domain. Overall, 70% of the transitions which were identified in the survey were in the vertical segment of flight.

Horizontal Channel Feedback

The primary feedback display for the horizontal channel is the Electronic Horizontal Situation Indicator (Map or Navigation Display) (EHSI) shown in Figure 5.15, which gives a full tactical and strategic view of the current and planned path of the aircraft. The position of the aircraft is shown by a white triangle in the bottom center of the display. Though the colours are not

distinguishable in this document, the aircraft is shown geographically relative to the inactive (blue) and programmed (magenta) waypoints in the vicinity. Programmed waypoints are those which have been entered as endpoints in segments of the FMS path (magenta line). The next active waypoint (LAPEL), the one that is actively being flown to, is shown in white. The current aircraft heading is shown on the compass arc on the top of the display.The second frame shows the same scene a few minutes later, after the descent to LAPEL has begun.
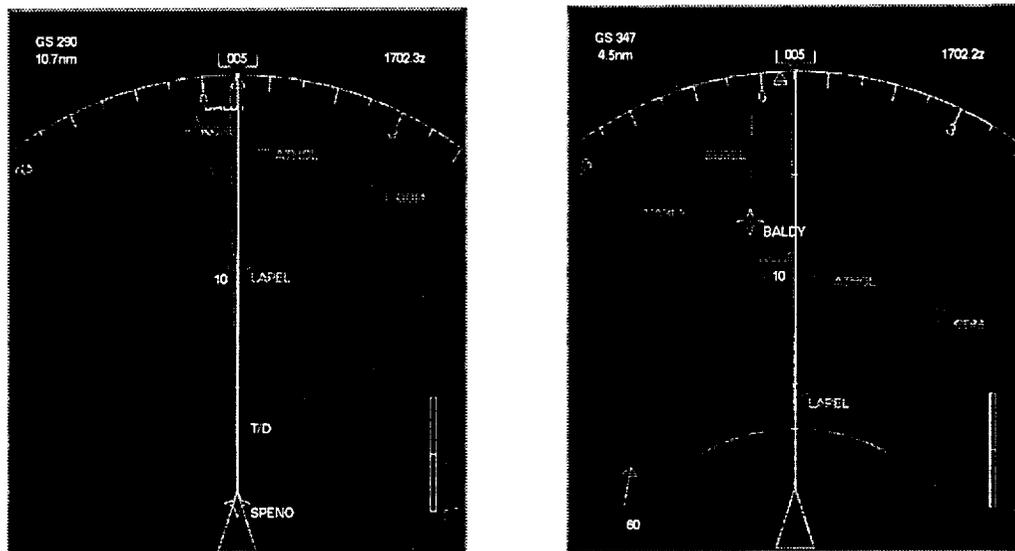
Figure 5.15:   Map Mode Display

By displaying the area in front of the aircraft, this display allows pilots to quickly ascertain how the aircraft has been programmed and anticipate what it is expected to do. In particular, any deviation from the programmed path can quickly be seen by the aircraft symbol flying away from the magnenta line.

Vertical Channel Feedback

The primary source of vertical channel feedback is the textual display in the Flight Mode Annunciator, in the top middle of the Primary Flight Display (PFD) shown in Figure 5.16. In this figure, the aircraft is tracking the ILS glide slope. To the right of the Attitude Determination Indicator, the magenta diamond shows the aircraft deviation above or below the glide slope, indicating, in addition to textually, that the current mode is Glide Slope Tracking. To the right of
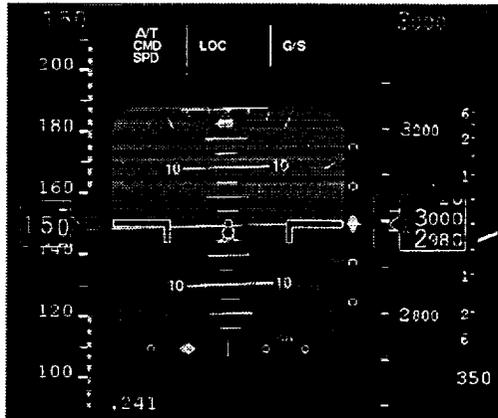
Figure 5.16:   Primary Flight Display

the altitude tape, the vertical path indicator shows the current descent rate of the aircraft, independent of active mode.

A few graphical elements related to the vertical path are displayed also on the Map Mode Display. The first frame in Figure 5.15 shows a Top of Descent marker about 3 nmi in front of the aircraft. During the descent, an Altitude Intercept Arc (as seen in the second frame) indicates where the aircraft is expected to reach its commanded altitude. The programmed VNAV path is implied by the Top of Descent point, but this actual path is only displayed as a series of text based altitude and speed restrictions on the flight management system. The various feedback mechanisms available for the vertical channel do not provide immediate graphical feedback.

The fundamental advantage in observability that the current horizontal channel has over the vertical channel is that the context of the horizontal channel is shown. The position of the aircraft, relative to other waypoint and elements is shown in a clear manner. The vertical channel shows the altitude of the aircraft, and tactically the vertical channel provides feedback for mode changes about to occur by showing markers on the speed and altitude tapes. Figure 5.16 shows a red low speed marker on the speed tape. However, the vertical channel does not show strategic relationships of the aircraft to targets and conditions which can effect the trajectory.

Feedback Mechanism versus Channel Complexity

There is also a fundamental discrepancy in the feedback available in the vertical channel and the horizontal channel. As shown in Table 5.3, the horizontal flight of an aircraft is limited to using a single method to control a single target state: controlling the heading of the aircraft by using the ailerons to for roll authority.

Table 5.3:   Representative Horizontal Modes in the MD-11

Horizontal Modes	Control Allocation
Heading	Roll
Track	Roll
NAV	Roll
ILS Localizer Tracking	Roll

Though the vertical channel is functionally more complex, there is less feedback to the pilots than in the horizontal. The vertical flight of an aircraft requires control of both the speed of the aircraft and its vertical path, since speed and vertical path are interrelated. This leads to the vertical flight of an aircraft having many more potential target states: vertical speed, altitude, preprogrammed vertical flight paths, airspeed, thrust level, aircraft pitch, angle of attack and so on. In addition, each of these targets can be controlled to by a combination of the aircraft pitch and the thrust level. Table 5.4 shows a tally of vertical and speed based flight modes drawn from the MD-11 Cockpit Pilot's Guide.

As an example, in a simple base mode, such as the Altitude Hold mode, the aircraft speed would be controlled by the throttle and the vertical path (in this case, the altitude) would be controlled by the pitch of the aircraft with the elevators. Another base mode, Flight Level Change mode, sets the throttle at a limit value to control the vertical climb rate, and controls the speed of the aircraft with the pitch.

5.3.2   Identifying Feedback Using the Hybrid Automation Representation

In order to effectively monitor the state of aircraft automation, pilots need to be able to predict the future state of the automation using available indicators and feedback. The Hybrid Automation Representation identifies the elements which are required by the pilot to track the

Table 5.4:   Representative Vertical and Speed Modes in the MD-11

Vertical and Speed Modes	Speed Control Allocation	Path Control Allocation
Altitude Hold	Throttle	Elevator
Altitude Capture	Mixed (MIMO transition)	
FMS Altitude Hold	Throttle	Elevator
FMS Profile Descent	Elevator	IDLE Throttle
Vertical Speed (MCP)	Throttle	Elevator
Vertical Speed (FMS)	Throttle	Elevator
Flight Path Angle	Throttle	Elevator
Flight Level Change	Elevator	IDLE or CLIMB Throttle
Glide Slope Tracking	Throttle	Elevator
Flare	Throttle	Elevator
Go Around	Elevator	CLIMB Throttle
Low Speed Protection	Elevator	Throttle
High Speed Protection	Elevator	Throttle

progress of automation state. These elements consist of the feedback to ascertain the state of each of the possible automatic conditions leading to the transition, the notification of a mode transitions and the target values of the new mode.
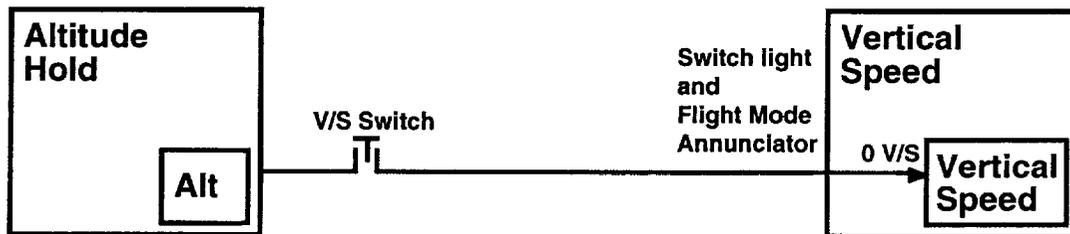
Figure 5.17:   Feedback of Mode Transition Occurring

The feedback to the pilot that a transition has occurred is explicitly identified in the nomenclature as part of the transition between modes. Figure 5.17 shows that the feedback provided during a manual transition from Altitude Hold to Vertical Speed is show by the switch lighting and on the Flight Mode Annunciator.

The second important element of feedback is identifying the criteria upon which automatic transitions will occur. Though this is not identified explicitly, it is necessary to identify when

transitions will occur in order to accurately monitor the automation. An example is that in Figure 5.18, it is necessary for a pilot to be able to identify when an overspeed condition will occur, likely through checking the airspeed on Primary Flight Display. An additional element which needs to be identified is the target values which are specified after a mode change. In Figure 5.17, the new target value for the Vertical Speed mode is a vertical speed of zero feet per minute. In Figure 5.18 the new target values for the Flight Level Change are a speed of Vmax and an idle thrust setting.
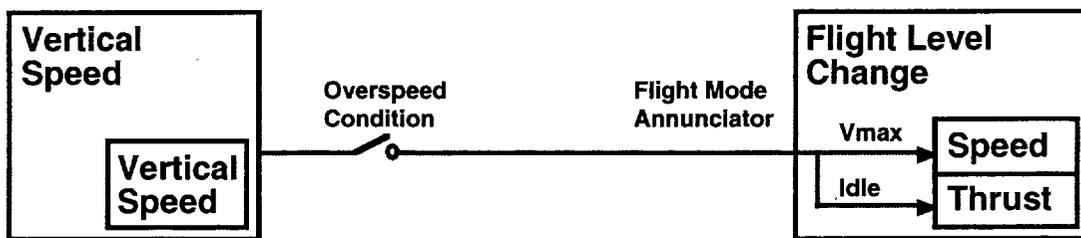
Figure 5.18:   Observability of Conditions and New Target Values

In examining actual autoflight systems, the conditions which need to be monitored are much more diverse, and appear on multiple displays in multiple locations. Figure 5.19 shows the altitude capture subsection of the MD-11 autoflight system. In this mode transition diagram, two conditions are highlighted where feedback is not provided. The transition from Altitude Capture back to Vertical Speed mode is predicated on the Altitude Select Knob being moved by the pilot and the *previous* mode being Vertical Speed. This transition is based on the autoflight system reverting to the previous mode, whether it was Vertical Speed, Flight Path Angle, or Level Change, if the Altitude Select Knob is moved. However, annunciation of the previous mode is not made available to the pilot.

Another example element of feedback which is missing is the state of the Linger Timer (in aircraft including the MD-11), which is not made apparent to the crew even though it can suppress an altitude capture from occurring. The Pitch Wheel is used in the MD-11 to specify the climb/descent rate while in Vertical Speed or Flight Path Angle mode. After the Pitch Wheel is rotated there is a pause of two seconds during which altitude capture cannot occur. This behaviour is mentioned in the MD-11 Cockpit Pilot's Guide, albeit in a positive light:
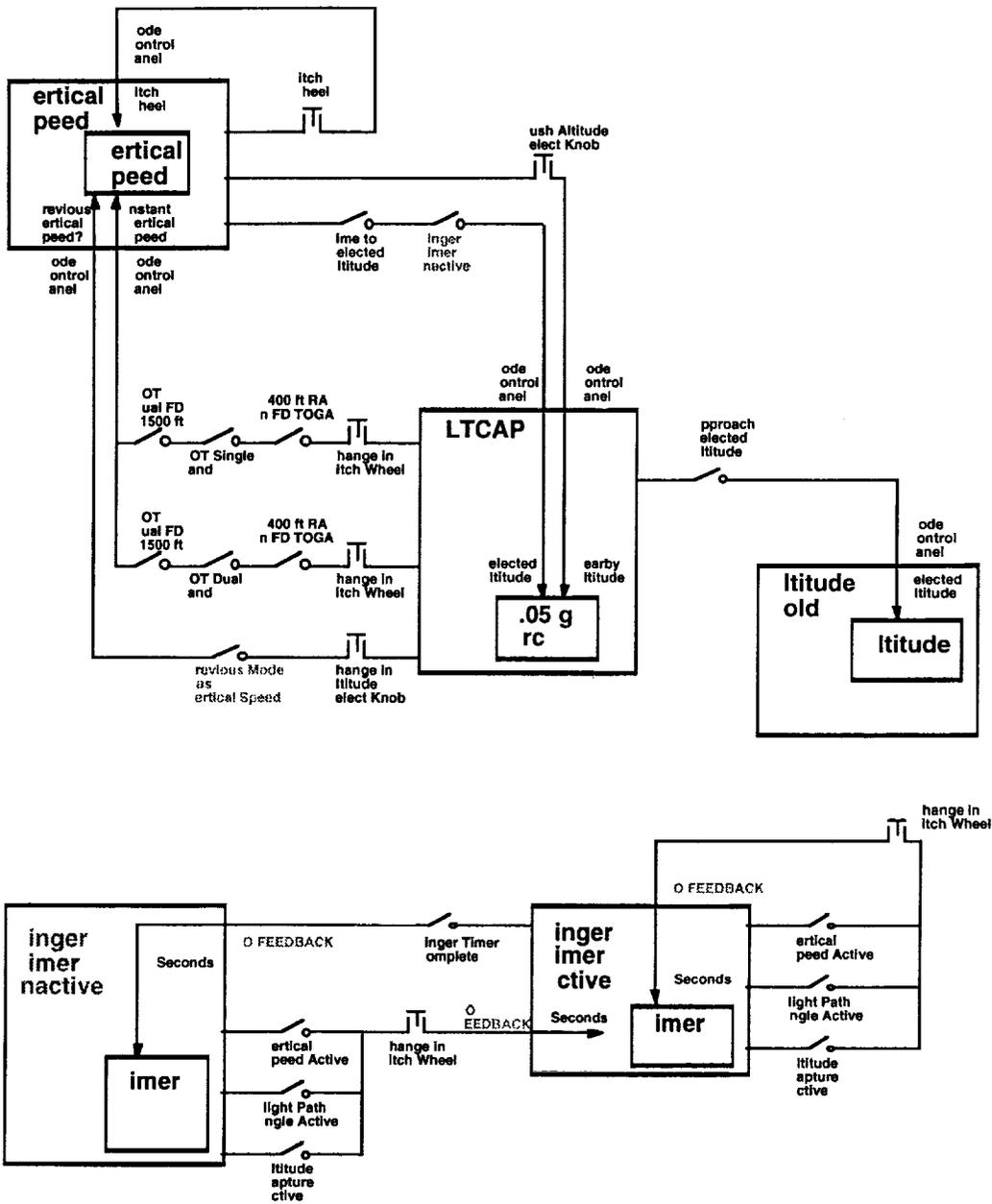
Figure 5.19:   Lack of Observability in MD-11 Altitude Capture

Rotation of the pitch wheel enables the pilot to exit altitude capture for 2 seconds whereupon if altitude capture conditions are satisfied, the aircraft reenters altitude capture. Otherwise vertical speed remains selected. (Honeywell, 1992)

This condition can be seen in Figure 5.19 on the transition between Vertical Speed and Altitude Capture. The Linger Timer behaviour, shown in the bottom diagram, is to engage for 2 seconds on the condition of a change in the Pitch Wheel. While this timer is active, the "Linger Timer Inactive" condition is not met in the Vertical Speed to Altitude Capture transition. As shown in Figure 5.19, there is no feedback as to the engagement, disengagement, and current state of the linger timer. This behaviour of the autoflight system has been implicated as a contributory factor in an incident:

> On July 13, 1996 an MD-11 experienced an in-flight upset near Westerly, Rhode Island. One passenger received serious injuries, and one passenger and two flight attendants received minor injuries. During this incident, the first officer adjusted the pitch thumbwheel seven times as the autopilot was attempting to level the airplane after descending. Boeing DPD engineers informed the Safety Board that, when the autopilot is engaged, movement of the pitch thumbwheel interrupts the autopilot's altitude capture mode. Once the pitch thumbwheel is released, there is a 2-second delay before the autopilot can resume the level-off. Therefore, the American Airlines flight crewmember's repeated use of the pitch thumbwheel during the level-off process prevented the autopilot from capturing the assigned altitude. The Safety Board learned that American Airlines operations and training personnel were not aware of this 2-second delay and that it was not addressed in the manufacturer's operations or training material. (National Transportation Safety Board 1999)
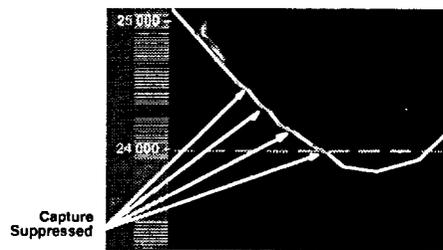
Figure 5.20:   MD-11 Inflight Upset

Figure 5.20 shows this incident graphically. During the descent, the crewmember's repeated using of the thumbwheel suppressed the capture for several 2 second incidents. These are shown graphically as the red segments during the capture maneuver. The capture was suppressed while the aircraft crossed over the target altitude, resulting in an open descent. The captain reacted by using the yoke to manually pitch the aircraft steeply to recapture the assigned altitude. The injuries with sustained during the recapture maneuver.

5.4   Case Study: Altitude Transitions

Many of the examples in this chapter are based on the behaviour of transitions to and from the Altitude Capture Mode from Vertical Speed and Flight Level Change. As discussed earlier, this is due to the transition needing to reconfigure the aircraft in a short amount of time, typically using Multi-Input, Multi-Output control for a smooth, low acceleration ride. As part of this thesis a set of probes was made of the behaviour of these transitions, both through a detailed examination of available literature and through simulator testing. The results showed that the transitions to and from Altitude Capture are inconsistent across aircraft, and in some cases inconsistent within a specific aircraft. The synopses of these results will be presented in this section.

5.4.1   Target Change during Altitude Capture

An issue with Altitude Capture, which may be related to physical or computational limitations, is its response to a change in the target altitude. Depending on the aircraft and the new target altitude, the response may be to level at the initial target altitude, level at the new target altitude, or to change modes to Vertical Speed. This behaviour was not documented in the Flight Crew Operators' Manual and was determined through simulator testing on the B737, B757/767, B777 and the A320.

On the B757/757 and the B777, changing the altitude target while in the Altitude Capture mode caused the aircraft to level at the *original* altitude target. On the B737 and the A320, the altitude target while in the Altitude Capture mode caused the aircraft to switch to a the Vertical Speed or Flight Path Angle mode at the instantaneous Vertical Speed or Flight Path Angle. Figure 5.21 shows these two behaviours. Note that the B737 and the A320 place the aircraft into a situation where it is "flying away" from the altitude target.

Figure 5.21:   Target Change during Altitude Capture

144

In most instances, this behaviour is benign. However, multiple instances were noted by pilots in the survey in which the behaviour was problematic. In particular, while capturing the glideslope from above, the MCP altitude is typically below the aircraft at the lowest altitude to which the air traffic control has granted clearance, often down to the runway field. If the aircraft transitions to Altitude Capture during this capture, the aircraft will begin to level. If the pilot attempts to switch the altitude to comply with an updated clearance, the aircraft may revert to Vertical Speed Mode at the instantaneous vertical speed, resulting in a very slow descent, and a failure to capture the glideslope. This is shown graphically in Figure 5.22.

Figure 5.22:   Reversion to Vertical Speed during Glide Slope Capture

5.4.2   Inconsistent Target Change during Climbs and Descents

The behaviour of the automation is particularly inconsistent when the target altitude is being moved from below the current altitude to above during a descent, or from above the aircraft to below, during a climb. Situations in which the aircraft is placed into a situation where it flies "away" from the target altitude are of concern. It is hypothesized that they can lead to scenarios where pilots expect the aircraft to automatically level at a target altitude (as per nominal operations), but do not. While not a typical directive from the pilot, or from air traffic control, this situation does occur, and places the aircraft into an "unprotected" descent—the aircraft will not level before intersecting the ground.

As discussed for the B737 in Section 5.2.3, aircraft have differing responses to the rate at which the altitude knob is moved, the initial climb/descent mode, and their response to the new target. Summarized below are the results of scenarios run on high fidelity simulators investigating this mode transition.

Vertical Speed Descent

The scenario placed the aircraft at FL330 (33 000 ft) in a Vertical Speed descent at -4000 fpm with the target altitude is set to FL290. As the aircraft passed through FL310, the target altitude was moved to FL350 at various rates.

Table 5.5:   Altitude Target Change during Vertical Speed Maneuver

	Fast Altitude Knob Rotation	Medium Altitude Knob Rotation	Slow Altitude Knob Rotation
B737-500	MCP altitude passes through current altitude and continued descent.	When MCP altitude is near FL310, capture was initiated, and ALT* engaged. As knob was continued to be turned, ALT* reverted to V/S with an instantaneous V/S target and continued descent.	Aircraft captures current altitude and levels off when MCP altitude is near FL310.
B757/767	MCP altitude passes through current altitude and continued descent.	When MCP altitude is near FL310, capture was initiated, and ALT* engaged. Aircraft did not regain V/S mode. Captured altitude few hundred feet below when ALT* initiated	Aircraft captures current altitude and levels off when MCP altitude is near FL310.
B777	MCP altitude passes through current altitude and continued descent.	When MCP altitude is near FL310, capture was initiated, and ALT* engaged. Aircraft did not regain V/S mode. Captured altitude few hundred feet below when ALT* initiated	Aircraft captures current altitude and levels off when MCP altitude is near FL310.
A320	FCU altitude passes through current altitude and continued descent.	FCU altitude passes through current altitude and continued descent.	FCU altitude passes through current altitude and continued descent.

If the altitude knob was moved quickly, as it typically would be, the B737/757/767/777 and the A320 all responded by continuing the descent. If the altitude knob was moved slowly, where slowly corresponded to a slower rate than the Linger Timer, the Boeing aircraft levelled when the target altitude became close to the current altitude. The A320 continued in its descent. Each of these aircraft were placed in a state where they were flying away from the altitude target.

The most interesting case was when the altitude knob was moved at a rate similar to the Linger Timer—at approximately one click every second. In this case, the B737 initiated an altitude capture and then, as the knob was moved again, reverted to Vertical Speed with an instantaneous vertical rate. The B757/777 did not revert to Vertical Speed mode; rather they transitioned to altitude hold few hundred feet below where the capture initiated. The A320 continued to in its descent, but it is suspected that a behaviour similar to the B737 may exist.

146

Flight Level Change Descent

The scenario placed the aircraft at FL330 (33 000 ft) in a Flight Level Change descent at a speed target of 320 kts, indicated with the target altitude is set to FL290. As the aircraft passed through FL310, the target altitude was moved to FL350.

Table 5.6: Altitude Target Change during Flight Level Change Maneuver

	Fast Altitude Knob Rotation	Medium Altitude Knob Rotation	Slow Altitude Knob Rotation
B737-500	Altitude passed through current altitude; and aircraft immediately spooled up engines and to capture the FL350 target	When MCP altitude is near current altitude, the capture initiated, and ALT* engaged. ALT* then reverted to V/S with an instantaneous V/S target resulting in a continued descent.	When MCP altitude was near FL310 (i.e., current altitude), aircraft captured current altitude and levelled off.
B757/767	Altitude passed through current altitude; and aircraft paused, then spooled up engines and to capture the FL350 target	When MCP altitude is near current altitude, the capture initiated, and ALT* engaged. Did not regain V/S mode. Captured altitude few hundred feet below when ALT* initiated.	When MCP altitude was near FL310 (i.e., current altitude), aircraft captured current altitude and levelled off.
B777	Altitude passed through current altitude; and aircraft paused, then spooled up engines and to capture the FL350 target	When MCP altitude is near current altitude, the capture initiated, and ALT HOLD engaged. Did not regain V/S mode. Captured altitude few hundred feet below when ALT HOLD initiated.	When MCP altitude was near FL310 (i.e., current altitude), aircraft captured current altitude and levelled off.
A320	When FCU ALT was at current altitude, aircraft switched from Open Descent to Current V/S (-3200 fpm).	When FCU ALT was at current altitude, aircraft switched from Open Descent to Current V/S (-3200 fpm).	When FCU ALT was at current altitude, aircraft switched from Open Descent to Current V/S (-3200 fpm).

If the altitude knob was moved quickly, the B737 immediately spooled up its engines to capture FL350 target. The B757/777 waited for the duration of the Linger Timer and then initiated a climb. The A320 switched to a Vertical Speed descent with the instantaneous vertical rate when the target altitude moved to above the current altitude. However, it should be noted that earlier generation A320s transition to a Flight Level Change type of mode to capture the new altitude target. This was changed because unintentional actions on the altitude knob could thus cause significant change in pitch and power. Therefore it was decided always to revert to the Vertical Speed leaving the aircraft following its current flight path, which is much less disturbing to flight crew and passengers.

If the altitude knob was moved slowly (relative to the Linger Timer), the Boeing aircraft switched to Altitude Capture and Hold when the target altitude corresponded to the current

altitude. The A320 switched to a Vertical Speed descent with the instantaneous vertical rate when the target altitude moved to above the current altitude.

Once again, the most interesting case was when the altitude knob was moved at a rate similar to the Linger Timer. In this case, the B737 initiated an altitude capture and then, as the knob was moved again, reverted to Vertical Speed with an instantaneous vertical rate. The B757/777 did not revert to Vertical Speed mode; rather they transitioned to Altitude Hold a few hundred feet below where the capture initiated. The A320 continued to in its descent, but it is suspected that a behaviour similar to the B737 may exist.

Behaviour while Altitude Target in Dynamic

The final test was to determine the behaviour of the system while the altitude target was in motion, during the process of entering a new target into the autoflight system. For this scenario, the aircraft is descending from FL330 in Level Change, speed target of 320kts, and the altitude is set at FL290. When the aircraft approached FL310, the altitude knob was moved to above FL350 and, without lingering at any altitude, was kept in constant motion.

Table 5.7:   Behavior while Altitude Knob is in Motion.

	Fast Altitude Knob Rotation
B737-500	Engines immediately started to spool up, without waiting for a stable altitude target to appear
B757/767	While altitude knob was in motion, descent continued. When altitude target stabilized, aircraft added power to climb.
B777	While altitude knob was in motion, descent continued. When altitude target stabilized, aircraft added power to climb.
A320	While altitude knob was in motion, descent continued. When altitude target stabilized, aircraft added power to climb.

The odd behaviour in this case was from the B737, which did not wait for a stable altitude to appear before initiating the climb. Further testing showed that the B737 would "chase" the altitude target while in the Level Change mode.

5.5   Chapter Summary

This chapter examined three areas in autoflight systems analysis: limitations in physical systems, the representation of the automation to the pilot, and feedback from the automation.

Each of these topics, where appropriate, was analyzed through the used of the Hybrid Automation Representation. A more detailed study was presented of altitude change mode transitions, which appear to be the cause of a significant number of problems. For this study, high fidelity simulators were used to investigate behaviours that were not documented in the Flight Crew Operators' Manuals. Several inconsistent, and potentially surprising, behaviours were found.

Chapter 6

Approaches to Mitigating Automation Complexity

This chapter will examine approaches to managing and mitigating the complexity of existing automation systems. Pre-emptive approaches have three classes, corresponding to the time and cost of implementation. The first is to apply more effective training with knowledge of the structure of the automation as well as known aircraft problems (Weiner, 1999). The second is to critically examine the feedback provided by the automation and enhance it to allow the system to become more observable, especially in light of hidden conditional elements and targets. Ultimately, however, the process by which these systems are designed must be examined. A process is presented which is designed to create systems which may not require the same amount of simplification by pilots.

6.1 Training and Procedure Modifications

Modifying training or instituting new procedures is by far the fastest manner in which to modify the representations, and hence behaviour, of pilots. Several approaches are already in active use within the industry, including publicly detailing the issues to make problems readily accessible to line pilots, and modifying procedures in order to make up for automation design errors or deficiencies. Changes to the process by which pilots are trained are also underway (Weiner, 1999).

6.1.1 Public Detailing of Issues

If the behaviour of automation is found to be confusing or insufficiently documented, especially in the aftermath of an accident or incident, flight crews of affected aircraft fleets are notified. The notification process may be initiated by the manufacturer or by airlines directly, and often consists of additions or modifications to the flight crew operating manuals.

An example is the 1994 incident in Orly, France, where the aircraft to pitched up to acquire an altitude target late in an approach. The automation was designed to attempt to capture the current programmed altitude in the event of a speed violation. In this incident, the crew had programmed

151

the target altitude above them, at the Missed Approach Altitude, during final approach (as per operational guidelines). When the flaps were inadvertently lowered at too high a speed, the AutoFlight System detected an overspeed and pitched the aircraft up in an attempt to capture the Missed Approach Altitude (Sparaco, 1994). In response to this incident, Airbus issued an advisory bulletin warning pilots to follow posted flap limit speeds carefully (Sparaco, 1994b). A more recent situation is the NTSB recommendation to include details regarding the vertical speed behaviour of the MD-11 more explicitly in airlines' flight crew operating manuals (NTSB, 1999). Unfortunately, both of these examples show the weakness of this approach: the updating process often occurs after an accident or incident has already occurred.

6.1.2 Modification of Procedures

Another near term, and relatively inexpensive, solution to automation problems is to modify the procedures pilots use when interacting with the automation. In aviation, a procedure is a codified behaviour consisting of a set of tasks to be performed by the pilot. Procedures are designed to provide structure in the complex operating environment of the aircraft cockpit, and allow responses designed to be optimal during both nominal and non-nominal operations. As an example, the procedure to change an altitude would consist of the pilot not flying (PNF) acknowledging the ATC request for an altitude change, dialing in the new altitude and pointing to the altitude window. The pilot flying (PF) has to verify the new altitude target verbally before initiating the altitude change (Midkiff, 1998).

Procedures are used to generate a known aircraft response by traversing a specific set of transitions, and avoiding known anomalies. The MD80 mode control logic has an error where adjusting the speed while in a Altitude Hold mode could cause the aircraft to revert to a Vertical Speed mode. The Vertical Speed target would be the instantaneous speed when the reversion occurred, causing a slow drift from the target altitude. To deal with this problem, pilots reset the Altitude Hold mode after making a speed change. Immediately pressing the ALT button on the MCP will place the system into ALT HOLD if it has inadvertently transitioned to V/S. Rather than monitoring the system to see whether it experience the anomaly, this behaviour simplifies the pilot response into a procedure which results in the correct behavior.

A detailed example is the response of a major American airline to the Cali, Colombia accident described below.

On December 20, 1995, a Boeing B757 crashed into a mountain near Cali, Columbia. This accident was caused by a chain of relatively minor incidents. The pilots in the aircraft were on a normal approach to the airport when they were cleared "Direct to Romeo," which was a reference to the ROZO waypoint on the approach path. The Flight Management System (FMS) was then used to select this waypoint from a list of nearby waypoints. The FMS attempts to simplify looking up waypoints by showing all common named fixes. In this case, the aircraft was cleared to Romeo, or more accurately, the waypoint on the approach which had a name started with an "R": ROZO. The pilots are hypothesized to have entered an "R" and selected the top waypoint on the list and that the waypoint which was selected was actually located near Bogotá, Columbia. The aircraft entered a turn to acquire this new waypoint. When the pilots realized that the aircraft was no longer heading towards the airport, they initiated a turn back towards the airport, but had already lost too much altitude and were laterally displaced an unsuitable distance from the intended path. The aircraft failed to clear a mountain and was destroyed. (Aviation Safety Net, 2000)

In response to this event, pilots flying the Boeing B757/767 were sent a bulletin (Boeing 757/767 Operating Manual Bulletin No. 757/767-19) which described the procedure to use during route modification.

If a route modification involves a navaid or waypoint with duplicate names:

Both pilots must verify that the latitude/longitude of the desired waypoint on the SELECT DESIRED WPT page is correct. Then, *both* pilots must verify that the course/distance to the selected waypoint on the LEGS page are reasonable before using the waypoint for navigation.

During approaches, verification of navaid waypoints by comparing latitude/ longitude may be impractical. Manual tuning and aural identifying of the primary navaid by the pilot using raw data is essential to verify that the intended navaid is selected.

NOTE: Latitude and Longitude for navaid (if available) are only shown on en-route charts, area charts, SIDs, STARs, and profile descents where the navaid forms an airway or route. (Emphasis in original)

Well-designed procedures which take into account automation limitations and inconsistencies can avoid known problems and render automation more consistent and easier for pilots to

monitor. Procedural changes can work around automation issues by requiring pilots to interact with a non-problematic subset of the automation, or to avoid sequences of actions which can lead to incidents. As an example (further detailed in the appendices), mid-generation MD-80 aircraft had an automation error where modifying the vertical rate during an altitude change could result in the aircraft flying through the MCP altitude. The behaviour was avoided if pilots re-armed the MCP altitude each time a vertical rate change was made. The procedural fix was to augment the altitude change procedure to include a re-arming task.

Procedures specify a particular "path" through the aircraft automation. By requiring that pilots are only to use particular paths, the complexity of the automation may be reduced. There is simply less active involvement with sections of the automation which are not part of proceduralized usage. In this sense, procedures are analogous to the complexity management approach used by pilots of reducing the possible behaviours of the system. In the extreme of this behaviour, some airlines have removed large sections of aircraft automation behaviour from their pilots' available repertoire.

6.1.3 Modification of Training Process

The immediate changes made to pilot behaviour through procedural changes and the dissemination of known issues ultimately need to be represented in the initial training provided to pilots for new aircraft (Weiner 1999). An obvious necessity is to provide trainees with the information available to pilots in the field, namely the known problems with the automation and the current procedures which compensate. There is also research underway on an advanced Vertical Navigation (VNAV) trainer (Chappell 1997, Crowther 1994). This tool is designed to show pilots the underlying complexity of the VNAV system and the implications of a particular set of mode choices.

However, some of the issues appearing in automated aircraft may have to do with the distinction between training pilots for operation versus understanding. This distinction appears on Reason's (1990) knowledge hierarchy as the distinction between rule-based interaction and knowledge-based interaction. The former, which is typified by the use of standardized procedures, may be sufficient for nominal usage where pilots may also develop robust

154

representations for these procedures. Unfortunately, this information not lend itself to interpolation and extrapolation to non-nominal conditions.

6.1.4 Utilizing the Hybrid Automation Representation for Targeted Training

The Hybrid Automation Representation may be useful in aiding the development of pilot operational models. As an example, the model may be useful in the identification of infrequently used modes or mode transitions. Using such a model in conjunction with operationally derived data from flights may allow the concentration of training on segments of automation with insufficient usage for experiential reinforcement. In some sense, this is the basis of the majority of simulator training, where time is spent practicing how to react to rare events, such as engine failures, control failure, and hydraulic problems. One recommendation is to extend this training to emphasize interaction with automation states with which limited operational interaction occurs and have critical consequences. With access to pilot-specific data from data recorders, it is conceivable that this could be done on an individual level. Section 5.2.2 has data from revenue earning flights that can be used to identify modes which are rarely used. The Mode Transition Matrix allows the examination of these rare transitions.

6.2 Feedback

A concern in modern automation is that sufficient information may not be available to the pilot to accurately track the state of the automation. Using the parlance of control engineers, the system is not "observable." Accurately characterizing necessary automation feedback can support modifications that make the system more tractable. However, since such changes to aircraft are much more rare, and expensive, than changes to training or procedures, they are likely to only be done in extreme situations. An example of such a situation is the retrofit to the A320 to call more explicit attention to the distinction between being in Flight Path Angle mode and Vertical Speed mode as discussed in Section 5.3. This change was made after an accident which was contributed to by confusion between these two modes (Hughes, 1995).

Feedback can also be used to serve to call attention to transitions or conditions which are operationally rare. If a rare mode has been entered (such as an envelope protection mode), or a rare condition has caused an unexpected transition, this can be made clear to the pilot. There is a

155

trade-off between how often a particular transition is made and how familiar a pilot may be with it. By selectively drawing attention to events which are "rare," it may be possible to compensate for an experiential liability (Tognazzini, 1992).

Feedback can also serve the more fundamental purpose of changing the nature of the system as experienced and interacted with by the pilot. In a manner similar to procedure design, a simpler system can be created by reducing the interaction to a smaller set of more capable behaviours. Alternately, a system can be made easier to monitor by presenting a pilot which more useful feedback about the state of the aircraft. Based on the vertical domain dominating problems with mode transitions in both the ASRS review and the web survey, an Electronic Vertical Situation Display was prototyped and evaluated (Vakil, 1996).

6.2.1   Electronic Vertical Situation Display

The Electronic Vertical Situation Display, shown in Figure 6.1, is analogous to the moving map display, but depicting the vertical progress of the aircraft. The display has four distinct areas. At the top of the display is the mode display window, showing the current and anticipated modes, control allocations and target states. At the left is a scalable altitude tape. The bottom window can either display the path distance (if in LNAV mode), or the range directly ahead of the aircraft. Finally, the main window shows the aircraft vertically in relation to the upcoming waypoints and mode transition points.

The current mode of the automation needs to be identified along with any of the specific attributes of the mode such as target values and control allocation. In Figure 6.1, the current mode is identified in the top window in green text, directly above the aircraft symbol. In this example, the aircraft is in VNAV Path Descent (VPATH). An example of a transition criterion is the dashed magenta line at 15 000 ft, which is the altitude dialed into the Mode Control Panel.

Anticipated modes consist of the future modes into which the automation expects the aircraft to transition. On the EVSD, the anticipated mode is shown in the top window above the point where it is predicted to be engaged. The anticipated targets and control allocations are depicted in a manner similar to the current mode. In Figure 6.1, the system is predicting a speed violation and a mode transition to the VNAV Speed Mode (VSPD). In this mode, the display shows that the
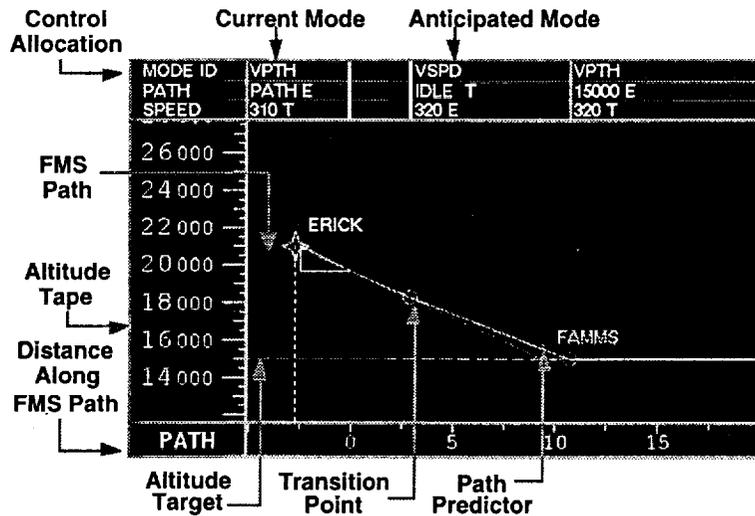
Figure 6.1: Electronic Vertical Situation Display

vertical path will be controlled by the throttles to Idle, and the speed will be controlled by elevators to 320 kts. Note that both the target states and the control allocation change when the new mode is engaged. Another mode change is anticipated once the aircraft reaches the MCP altitude of 14 000 ft, approximately 12 nmi ahead of the aircraft. The aircraft will switch to VNAV Path Mode (VPATH). Once the automation switches to this mode, the altitude becomes a target, as shown in the box underneath the VPATH text.

This display has a green "Path Predictor" line which shows the future vertical state of the aircraft using a linear extrapolation based on the current automation state. This line shows the behaviour of the aircraft in the context of the impending airspace and how it may differ from what is expected by the pilot. The feedback provided is much more useful that what currently exists in aircraft in helping to track conformances with commands.

Evaluation

The approach to EVSD evaluation was to have subject pilots act as Pilot Not Flying (PNF) and observe a set of scenarios running on a part task simulator. Subjects were drawn from a pool of current "glass cockpit" airline pilots. While subjects observed the scenarios, the researcher acted as Pilot Flying (PF), interacting with the simulated aircraft automation and responding to prerecorded Air Traffic Control directives. If during the course of the scenario the subject felt that

157

some sort of *mode event* occurred, they pressed a button on the side stick controller and articulated their concern. This audio data was later analyzed for timing and content. A mode event was described to the subjects as an uncommanded mode transition occurring, an error had been made in interfacing with the FMS, or an unsafe or nonprocedural operation had taken place.

Summary of Results

The Electronic Vertical Situation Display was found to significantly improve mode awareness understanding and the detection of mode awareness problems in both subjective and objective measures of subject response. The full survey used to derive subjective results is available in the appendices. Objective results were particularly strong when the anticipation functions of the EVSD could be used to foresee an event before it actually occurred, as shown in Figure 6.2. In this scenario, pilot communicated with ATC when they felt the aircraft would not be able to make the crossing restriction at MLT. The results show that with the EVSD, pilots were able to make this assessment sooner (lower values are better).
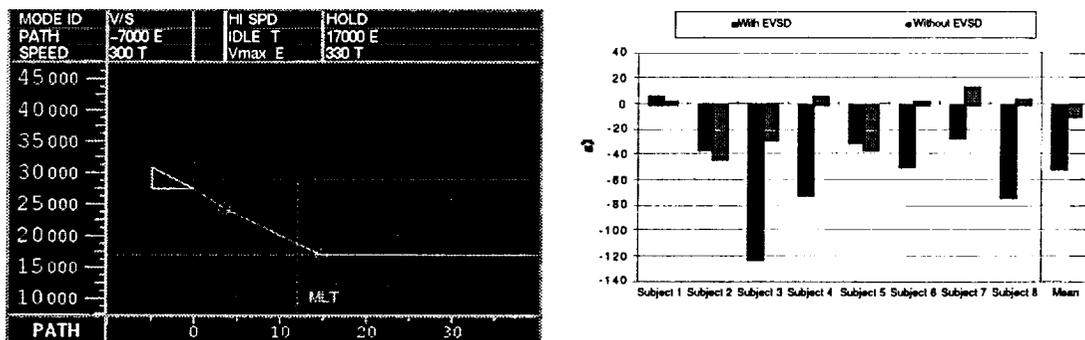
Figure 6.2:   Overspeed Envelope Protection during Altitude Change

Amalgamated ratings of pilot understanding of mode awareness problems over the full set of scenarios increased when the EVSD was available. Figure 6.3 shows the results of probing pilot understanding. These results were found to be statistically significant at the 90% level using a Wilcoxon analysis.

In addition, subjects were much more specific when reporting problems to Air Traffic Control. For example, rather than simply reporting that they were unable to make a crossing restriction, subjects would also report how far past of the waypoint the altitude would be acquired.

158
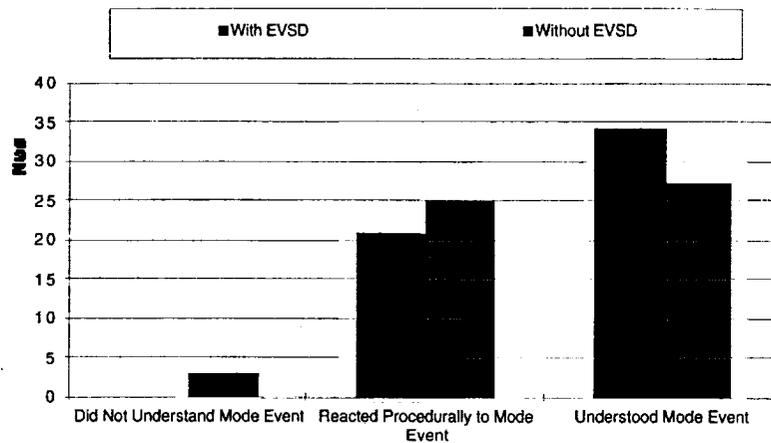
Figure 6.3:   Amalgamated Pilot Understanding Histogram

Several subjects also mentioned the additional utility of having a vertical image of the aircraft's programmed flight.

Subjective results were also positive. Pilots were asked to rate the value of the EVSD on a scale from *Very Valuable* to *Very Detrimental*. The results of this questionnaire are shown in Figure 6.4. Subjects were volunteers for this experiment, so results may be biased by a predisposition to new technology in the cockpit.
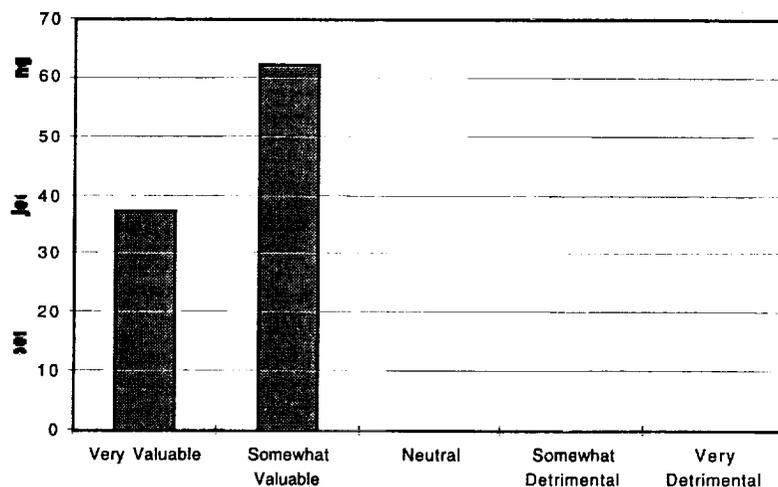
Figure 6.4:   Subjective Questionnaire: How Valuable was the EVSD

159

A comparison between the EVSD and the current AFS is shown in Figure 6.5. These results indicate that the subjects found the EVSD useful in a wide range of tasks. However, this is inconsistent with the objective measures, which suggest that the EVSD tended to be less useful in instances where another instrument provided the same information, especially when the task involved tactical types of inner loop monitoring, as compared to strategic monitoring situations.
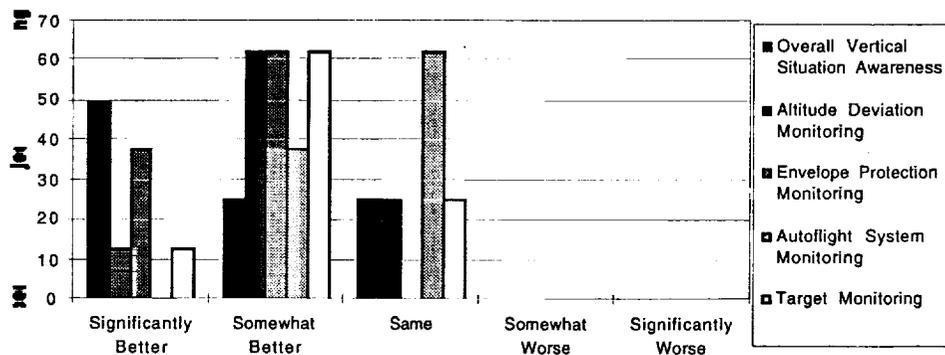
Figure 6.5:   Subjective Questionnaire: Comparison between the EVSD and Current Vertical Feedback Mechanisms

Finally, subjects were asked to rate the usefulness of specific elements of the EVSD. Figure 6.6 shows that subjects were not concerned with the control allocation, or the redundant target information provided in the top bar of the EVSD. The interaction of the Green (Aircraft Path) Line, the VNAV Path information and the graphical target states were cited as being useful.
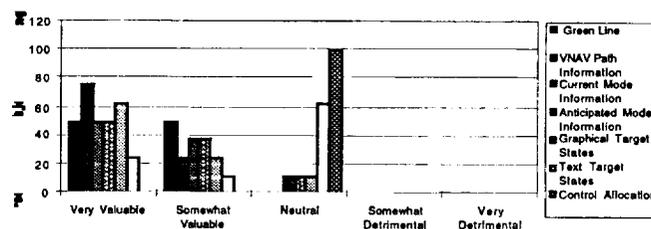
Figure 6.6:   Subjective Questionnaire: Value of Specific EVSD Elements

6.3   Operator Directed Process

The most comprehensive solution is to develop operator-consistent automation that is less vulnerable to problems. The costs and development times involved make this unlikely to occur as retrofits to existing aircraft. More likely, this solution will only be undertaken when new

functionality in the cockpit requires redesign in future aircraft. Work has been underway to determine if guidance can be provided *a priori* to designers to allow the creation of less vulnerable systems. However, in the domain of flight automation, it appears that a process-oriented solution is a more effective approach. This section will discuss the reasons for recommending a process modification and will discuss a specific example, the "Operator Directed Process."

This thesis has made an argument for the necessity of a consistent global model of the automation to be made available to the pilot. Such a model is necessary in order to develop training material and procedures with which to build a well-populated mental representation of the system. This mental representation is required in order to allow the automation's conformance with commanded goals to be monitored. The creation of this global model is the critical first step in the process of creating tractable systems with controlled complexity. In order to cause this change, the process by which these systems are designed must reflect the requirement that the system be represented in a form usable by pilots.

The reason that the trained pilot is a critical element in design is that the task of flying incorporates multiple levels of understanding, from manual skills up to deep knowledge, which are manifested and derived from the flight environment. This environment is usually not available or fully apparent to designers while systems are being developed. As such, the representations of the systems may differ significantly between pilots and engineers, especially when the complexity management techniques used by pilots come to bear. Differences in abstractions and simplifications may undermine the ability of pilots to effectively monitor the automation for conformance. This context and operational dependency distinguishes the field of aerospace from many others. As such, the operational input of pilots is critical in system design.

6.3.1 Process-oriented Solutions

When this research was started, the goal was to create a set of succinct guidelines for designers to follow. Systems which were designed in a manner consistent with these guidelines were to be less vulnerable to human interaction issues. In practice, many of these guidelines are already known, and available to designers: consistency, simplicity, transparency, and other basic

161

elements of human-computer interfaces (Schneiderman 1987, Tognazzini 1992, Card 1983). In aerospace, however, these terms must be considered from the viewpoint of the trained pilot, rather than the viewpoint of a designer or naive user.

The existing development processes for flight automation were developed in an era where computing power was at a premium and the capabilities of computing systems limited. The shift to a new development process is justified based on the flexibility, capability and complexity afforded by modern processors and software systems. One of the goals of the Operator Directed Process is to constrain the complexity of these systems to a level which human operators can internalize and understand while maintaining the necessary functionality. It should be noted that the ODP may suggest limiting functionality for some systems if the required system is so complex that it proves intractable to the pilot.

What is necessary is a mechanism with which to capture the most important system elements, as found in the "engineer's" representation of the system in a form which is suitable for the operator. This is a difficult task since is requires capturing a fundamentally complex task in a simpler form. The Operator Directed Process is one proposed mechanism to put considerations of the human pilot into the development cycle.

6.3.2   Operator Directed Process

Many of the problems appearing in modern aircraft appear to be related to a mismatch between engineering and pilot models. The issues of a lack of consistency, especially among rare modes, and lack of observability can result in inappropriate operational behaviour. One solution is to explicitly consider the human operators early in the design process to prevent these mismatches. This also assists in capturing the limitations that the operator may place on the autoflight automation and to constrain system complexity early in the design process. The fundamental principle is to increase system usability through the constraint of complexity by articulating an operationally appropriate model of automation for use by operators. This may also be extended to assist in certification process.

In order to support the development and certification of complex automation systems that consider flight crew operational understanding, the use of an Operator Directed Process (ODP) is
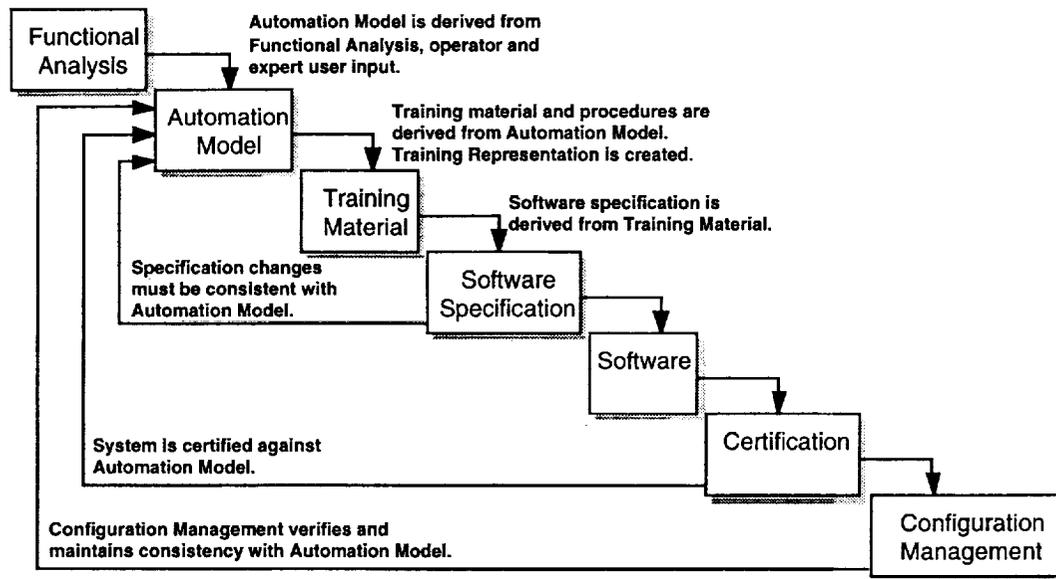
Figure 6.7:   Operator Directed Process (Waterfall Model)

proposed. The Operator Directed Process is shown schematically in Figure 6.7. The major difference in this process is that the training material is the source of the system specification rather than vice versa. Developing training material early forces consideration fundamental issues in human-machine interaction early in the process. This contrasts with existing development cycles that use training material to *document* system design. The intent is to develop a less error-prone and more understandable system by requiring consistency between the training material, procedural usage, and the software, and by limiting the complexity of the system through the articulation of a model for the operator. This enables the explicit consideration of the human operator early in the development process.

In Figure 6.7, the ODP is shown to follow the "waterfall model" used in classic software engineering. The waterfall model flows information and design considerations "downstream" to be dealt with by the next stage. The major stages of this process are needs analysis and specification, design, implementation, testing, and maintenance and upgrades. This is used as an explanatory diagram in order to show dependencies. In practice, it is closer to Boehm's (1981) "spiral" model which consists of a series of repeating stages of iteration, where updates are made to an operational prototype of the final system. An iterative version of the ODP is shown in Figure 6.8.

163

Functional
Analysis

Functional Analysis is
performed to identify
necessary tasks within
operational environmental

Automation Model for
Pilots is derived from
Functional Analysis

Automation
Model for
Pilot

Automation Model is
augmented into Designer
Model and Software
Specification

Software is Incrementally
Instantiated

ncremental Instantiation

Software
Specification

Design
Model

Software

Training
Material and
Documentation

Human in the loop: Prototype

Human in the loop: Interim System

Human In the loop: Actual System

Pilot — Software — Physical
System

Certification is against
Automation Model,
Software, and Training
Material

Certification

Explicit rationale capture allows
consistency to be maintained with
Automation Model

Configuration
Management

Figure 6.8:   Operator Directed Process (Iterative Model)

The Operator Directed Process is based on earlier ideas, including User Centered Design (Norman, 1988), Knowledge-based Interface Design (Shneiderman, 1987), and others (Card, 1983). The process is particularly appropriate to aircraft systems because of the skilled set of operators (i.e. pilots) who may have a differing characterization of tasks than designers. Another major factor is the manner in which procedures influence the task of flying by imposing external structure. For example, in the case of SIDs (Standard Instrument Departures), STARs (Standard Terminal Arrival Routes), and automated approaches, the automation is tied to the structure of the procedure, and the task fundamentally requires the use of the automated system. With the additional target flexibility provided by RNAV capabilities, more reliance on automation for

standard procedures is expected. By contrast, much of the research that has been done into how humans interact with computers has used case studies where operator skill is tied to the use of the computer, rather than the larger task (Card, 1983; Schneiderman, 1987), and where fewer operational impositions exist.

6.3.3 Functional Analysis

The first stage of this process is to determine the functionality that the automation system requires. This analysis needs to be based on the existing environment in which the automation must function and the anticipated operational and procedural usage of the automation. Several other researchers have published work to guide this process (Boy 1998, Vicente 1999).

6.3.4 Automation Model

The key element of Operator Directed Process is the creation of an Automation Model suitable for the pilot. It is derived based on the functional analysis and input from current design engineers, operators, and expert users. This is a representation of the automation which can be articulated and used operationally by the pilot and is a necessary construct for effective monitoring. The purpose of creating this model early in the process is to use it to limit the complexity of the automation, either by limiting the behaviours and functionality of the system, or by consistently abstracting the system at a higher level. This model is intended to be a high level description of the system which captures the philosophical and design goals which lead to specific design criteria at more detailed levels.

The primary goal of the automation model is that it must be capable of describing and explaining all the behaviours of the system *that matter*, and all of the derived operational procedures. The term *that matter* defines the operational dependency—if it is necessary to explain a behaviour in order to utilize the system in an operational environment (including emergency situations), it must be captured in the automation model. In this context, an appropriate model is likely to be one that is rooted in operational domain and acknowledges the background of the user. The representational form of this model is dependent on the automation it is attempting to describe. A number of possible modeling bases are presented in Table 6.1 (see also Rouse, 1986).

Table 6.1:   Possible Automation Model Representations

Control Block Diagrams	Control block diagrams are useful for continuous systems where they can accurate represent the continuous behaviour of a mode. Typically they are used by system designers.
Procedural Constructs	Procedures are used extensively in Pilot Guides and provide a well-defined procedure to accurately instruct the automation. However, they can become confusing: "Through the FCU, an immediate climb/descent is initiated by selecting the desired altitude in the ALT SEL window and either pulling the set knob or pressing the LVL/CH P/B to engage the LVL CHANGE mode. Pressing the LVL/CH P/B also disengages PROFILE, however, if PROFILE is engaged, pulling the set knob does not disengage it, rather it initiates an immediate climb/descent to the altitude selected on the FCU. The exceptions are..."
Finite State Representations and Variants	Use and extend Finite State Machine notation, terminology and analysis techniques to gain insight into underlying modal structure of complex systems. Asaf Degani (1994) used state charts to represent modal systems and was able to model certain mode transition errors. The Hybrid Automation Representation uses a Mode Transition Matrices to accomplish similar ends.
Analogical Descriptions	Many systems are described by an analogy to a previously understood description. An example might be that "This is controlled just like a B727 autoflight system". Graphical user interfaces in modern computers use a desktop metaphor. Spreadsheets embrace and extend the ledger book paradigm.
Anthropomorphic Descriptions	Automation can be designed to emulate a human agent. If successful, the operator can interact with the automation with very little training. However, this approach is limited by language accuracy, completeness, and ambiguity.
Linguistic Descriptions	Used by Riley (1995, 1997) to build system functionality with a consistent language to describe Air Traffic Control Directives. This is a wrapper around the existing automation and its issues. There is some concern that functionality and specificity may be lost with these high level descriptions.
Petri Nets	Useful for capturing all permutations of interactions and for capturing details of reactive systems.
Explanatory Descriptions	Purely explanatory descriptions are used extensively in Pilot Guides to provide a template for usage. For example: "For demonstration purposes, assume level at FL330, a climb to FL370 is desired at PPOS. A manual climb to FL370, utilitizing the PROG page to input data to the FMS is affected in the following manner: Write 370 or FL370 in the SP Press LSK Selected 370 in the ALT SEL window of the FCU"

An additional advantage of an explicit automation model is that it may serve as a stage in development where "rationale capture" can occur. It has been argued that the development processes which are currently used do not have a means to document the rationale behind the design decisions (Leveson, 1998). By capturing this information, the underlying premises of the system can be documented explicitly, aiding in maintaining consistency during subsequent modifications and extensions to the system.

6.3.5 Incremental Instantiation

The waterfall model of system design consists of a linear set of steps which are followed to create a product. The waterfall model flows information and design considerations "downstream" to be dealt with by the next stage. The major stages of this process are needs analysis and specification, design, implementation, testing, and maintenance and upgrades. Similarly, for software, these typically consist of the creation of functional analysis, followed by a software specification, system instantiation and finally the development of documentation. Boehm (1983) has shown that this development approach is inappropriate, and can be proven to be incorrect. This is due to the unknowns in the development process, which requires the design of systems in the absence of complete understanding of the problem to be solved, or its solution.

By contrast, the Operator Directed Process utilizes Boehm's (1983) "spiral" model. This consists of a series of repeating stages of iteration, where updates are made to an operational prototype of the final system. In Figure 6.8, the sections are delineated by gray boxes to indicate that these encompass the necessarily iterative stages of design and require human-in-the-loop testing. It is recognized that in order to effectively design, document and evaluate early revisions of a system, it may be necessary to create and evaluate prototypes in a manner consistent with the spiral model. The reverse arrows shown in Figure 6.8 show the manner in which "downstream" events can impact earlier stages and result in another iterative cycle. Determining when to iterate is dependent on the size of the system. Simple systems may be able to be validated by inspection, whereas more sophisticated systems may require full simulations in order to determine their effectiveness.

By explicitly requiring the creation of the Automation Model early in the design process, the model can be examined, prototyped, and evaluated by human factors experts early in the design cycle. By specifying the Automation Model it is also possible to gain some objective measures of the new automation during the iterative stages. The model defines a specification for interaction between the pilot and the automation which can be evaluated using part task or inexpensive simulators. The automation can also be examined by training personnel to assess difficulties which may arise during instruction. Using this early feedback, decisions to change the system behaviour can be made while they are much less expensive (Tognazzini, 1992).

The other important difference is that the flow of requirements is always from the automation model to both the training representation and software specification. This is intended to prevent software implementation problems from becoming documentation details in the reverse flow. By generating the specifications and software from training material, an explicit constraint is developed which can serve to guide system development. In the event that the specification cannot be made to reflect the training material accurately for technical reasons, the model for the operator must be amended. This change must be evaluated in the context of, and reflected in the training material and procedural design. At this point, changes to the software specification must be rederived.

Design Model

In order to be formulated in a manner which can constrain system complexity for the human operator, the Automation Model is likely to be insufficiently complete to specify the entire system. The Automation Model may be an inappropriate model, or in an unsuitable form for design. Research is underway (Feary 1999) which examines the different representations used by engineers and by pilots. It appears clear that a comprehensive model, such as the Hybrid Automation Representation or related models by other researchers, fall into the previous category. By contrast, pilots tend to have a more anthropomorphic view of the automation as an additional crew member.

As such, the Automation Model is necessarily a subset of Design Model, which augments the Automation Model with necessary implementation details. Typically, the Design Model will become the basis of the full software specification and the interface control document which are used to further augment system details. When creating the Design model, those elements of system behaviour which require clarification are considered in the context of Automation Model. If implementation details results in technical limitations, the solutions must be evaluated in the context of, and may require modifications to, the Automation Model. As an example, any modification which reduces the consistency of the Automation Model will immediately increase the complexity and size of the model and negatively impact the training material. Having access to the Automation Model and training material allows the evaluation of the modification in the context of both the human operator in addition to the engineering and software rework.

168

6.3.6 Training Material

One of the concerns is that any complex engineering model may not be an appropriate representation for pilots. To overcome this issue, the ODP process derives training material based on the automation model. This derivation assures that the proposed automation system can be presented in a form amenable to training. The training material description of the system can then be presented to pilots for feedback.

In designing this process, few limitations have been placed on the form or content of the training material. Rather than attempting to prescriptively specify the form, structure, or nature of the training material, the goal is to explicitly require the consideration of the specifics of knowledge transfer to the pilot. Domain-specific training experts are likely to have an understanding of the appropriate material and how it should be presented. For some applications the presentation of a structural model of the system may be sufficient training. For others, a detailed explanation of how the system is to be used procedurally in various operational scenarios may be more appropriate (Sherry 1999, Leveson 1998). It is likely that cockpit automation is in the latter grouping. As such, the development of training material will also include the development of procedures for both nominal and non-nominal scenarios.

Training Defines System

A consideration during the design of training material is to realize that the training material proscribes how to use automation and forces particular types of interaction (Orr, 1996). Training is a device which is constructed to convey information to the operator. The choice of what to include and what to exclude from this device can seriously impact the nature of interaction with automation. In Orr's work with photocopier technicians "directive documentation" was supplied: a service manual which is designed to instruct the technician during the development of diagnosis and repair. Directive documentation is an outgrowth of the scientific management tradition of rationalizing the work process (Orr, 1996) by reducing the job to a set of instructions which can be performed with minimal knowledge. In addition, the documentation designer's projection of what tasks they technicians are expected to perform is severely constrained. The first constraint is in the information made available by the engineers to those designing the documentation and the second is the policies implicit in the company about which tasks are appropriate to be done by

169

technicians. The response of technicians is to attempt to understand the rationale behind the documentation to gain more insight into the system. By doing so, experienced technicians are capable of solving the problems that were unanticipated during the documentation design.

Extending this concern to the flight environment, the training material defines the system which the pilots interact through the choices of information which are included and those which are excluded. These decisions are made prior to operational usage, and may marginalize the information useful during operation, due to documentation designers not having the appropriate operational insights. In addition, pilots need to be capable of solving unanticipated problems although the capability to explore and understand aviation automation is constrained by its safety critical nature.

Procedure Design

Procedures are used to define a rigorous pattern of interaction with the automation designed to maximize flight safety and efficiency. The fact that procedures define much of the nominal operational usage of the aircraft automation makes them a critical segment during automation development. In a highly proceduralized environment, such as aviation or medicine, the design of training material and the design of procedures to maximize the utility of the system must be done simultaneously. From a simplified standpoint, the procedures define how the automation will be used, especially nominally, during operations operationally. As such, the procedures define a segment of the necessary training material.

In some sense, the relationship between procedures and training is analogous to the relationship between a forward and reverse model. These terms, coined by Norman (1988), refer to differing ways of approaching automation. A forward model is the one typically used by engineers: "If I'm in this state, what does the system do?" By contrast, a reverse model is required by pilots during operation: "I want to do this, how do I get in the state to do it?" Forward models are used in most training material, as they can lay the basis for generating answers to reverse model queries. However, in most systems there are multiple approaches to completing a task. Procedures are designed to specify the appropriate approach based on multiple criteria. As such, procedures generate and populate the reverse model and answer the question of "How do I do

this" in a more specific way: "The appropriate way to do this is..." This duality between procedures and training makes their simultaneous development critical.

Care must be taken during procedure design because the experiential interaction will augment and modify the mental representation of the pilot. This modification will take place through the mechanisms described in Section 5.2, where pilots representations are simplified to fit an experientially-based model. Since nominal operational procedures will be used regularly, their associated mental representation will be highly accurate—or at least highly populated. If the automation behavior during these procedures is inconsistent with more rarely used modes, confusion may arise in non-nominal operations.

The final concern is that procedural design is typically considered an operational concern rather than an design engineering concern. In order to maximize the usage of automation and its understanding, operational insights must be used in the design of procedures. Without this insight, the automation, while fully capable, may be mismatched to the procedures which are necessary in flight. A concern is that the majority of procedures are currently designed by the end customer of aircraft: the airlines. At this stage of development, the design of automation is complete and is incapable of being modified in order to integrate tightly with procedural philosophies and guidelines. However, the growth of computer power may result in flight automation which can be modified on an airline by airline basis in order to allow tight coupling between automation, training, and the design of procedures.

6.3.7 Certification

The current aircraft certification processes were originally designed for the mechanical and electrical aspects of aircraft airframes. This approach has been successful, as shown by the reduction in airframe-related incidents in Figure 1.1 and Figure 1.4. Unfortunately, it does not appear that the approach is as effective in the fields of software design or human factors, likely due to their implicit complexity. The human factors aspects of certification have been recognized as being inadequate:

Current standards for type certification and operations have not kept pace with changes in technology and increased knowledge about human performance. For example, flightcrew workload is the major human performance consideration in existing Part 25 regulations; other factors should be evaluated as well, including the potential for designs to induce human error and reduce flightcrew situation awareness. (FAA, 1996)

Currently, certification authorities do not have the means or criteria available to require aircraft designers to create systems which address human factors issues. With the exception (noted above) of workload issues, certification authorities do not have the means to conduct an evaluation of human factors issues early in design. This has resulted in the evaluation of aircraft flight decks being conducted during flight tests when a design is nearly finalized at the end of the development cycle. At this stage, changes are both expensive and difficult to make.

After design is completed, flight testing is also able to consider human factors issue. However, if problems are found at this stage, it is again too expensive to change the automation, and procedures are often designed to compensate. By imposing a process-oriented solution, it may be possible to minimize the use of procedures in fixing design vulnerabilities.

Current Software Certification Practice

The widespread use of digital avionics has resulted in the need to certify software. This is also a difficult problem, both due to the inherent complexity involved in the creation of large software systems and the difficulty in attempting to quantify its accuracy. The current means to certify these larger systems has been to impose a specific process and set of documentation during development. The document entitled "Software Considerations in Airborne Systems and Equipment Certification" (DO-178B) (RTCA, 1992) specifies a traceable process of software development designed to prevent errors in software. The basic approach is to carefully specify the functions required of the software and to document the stages of translation from high-level specification down to object code. Regular code reviews are also incorporated into the process with the premise that errors in the software will not survive the scrutiny of multiple reviewers.

Note, however, that the review is not designed to be done by the certification authority directly. Instead, the certification authority verifies that the process has been put in active and effective usage and relies on the process to generate accurate and error-free software. In a similar

172

manner, it is suggested that the Operator Directed Process could be used as a process-based mechanism to drive the explicit consideration of human factors issues early in system design.

A major departure of the ODP from existing development process involves the manner in which certification is undertaken. Rather than solely certifying that the software conforms to its specifications, as is done in DO-178B, in the Operator Directed process the final system is compared to the original model and training material created for operators. By doing so, the levels of interpretation and translation which have been traversed in order to design the system can be determined to be appropriate.

Additionally certifying to a known automation model may also have advantages in identifying the exceptional cases. Since the automation model is constrained to a simple form, exceptional behaviours and modes are likely to be highlighted. Examples include non-nominal or rarely used modes or automation failure modes. Examining these cases may allow certification officials to focus on the more problematic issues, similar to the methodology described in Section 5.2 of this thesis. Similarly, the training material can be examined for consistency with the system instantiation.

6.3.8 Configuration Management

The concerns outlined above are focused in type certification of aircraft and components. This refers to the certification of initial equipment from the primary manufacturer. A secondary concern is that changes made to the system need to be approved as "Supplementary Type Certificate" (STC). Any individual or company can apply to modify an existing type-certified airplane through the STC process, but may not be aware of the design decisions made by the original manufacturer. The "philosophy" of the flight deck, the operating assumptions, and other consistencies designed into the system are not currently documented as part of the certification process, and so cannot be considered during the STC process. As such, it is possible for approval of a flight deck modification which is not consistent with the original manufacturer's design. This lack of "rationale capture" is a concern in current aircraft and certification processes. The basis for design decisions is not documented during development, nor is it required by certification. This lack of documentation makes it difficult for inconsistencies to be discovered and evaluated

by regulatory agencies, and for the underlying basis for design to be used when upgrades and changes are made to these systems. If the Automation Model can be captured during initial design and made explicit to parties who modify aircraft, it may be possible to maintain more consistent systems through the life cycle of the systems (Littman, 1987).

6.3.9 Experimental Evaluation of ODP

While a controlled comparison of the ODP to a conventional development process was not undertaken, an opportunity arose to take advantage of the process early in design. Usability can be measured through multiple means, including rapid automation training and adoption rates. In order evaluate the efficacy of the ODP in improving device performance, a planned software development project was identified upon which to test the process. This project was chosen based on availability and access to the planned and ongoing development effort. Full results of this experiment, shown in Appendix G, demonstrated rapid acceptance and minimal training in the new system and high user satisfaction. While this does not constitute a controlled experiment comparing the ODP with a conventional development process, the results are still compelling.

Additionally, several lessons were learned during this experiment. The first was that the automation model was the critical element in the process. Training material was not as critical as expected, at least with a small system, and documentation was left until the final stages of design. It was also found that a designer model, which was an augmentation to the automation model was required in order to deal with each possible situation and behaviour the system was capable of undertaking. The automation model or training material proved to be insufficient for implementation or even specification.

The waterfall model of development, shown in Figure 6.7 was found to model the development process much more poorly than the iterative model shown in Figure 6.8. The spiral iterative model lent itself to both incremental software development and incremental evaluation. Each of these were required to allow regular human in the loop testing during design and development. This process was necessary at each incremental system update during development and so is explicitly identified in the process shown in Figure 6.8.

174

6.4 Chapter Summary

This chapter has considered approaches that have been, and can be, used to manage the complexity of aircraft automation systems. This breaks down into three approaches. The first is to change training and procedures in order to modify pilot behaviour in situations which are known to be problematic. These changes can be effected very quickly and inexpensively, and can be based on the approaches already used by pilots to control complexity. Several procedural changes were presented, and some approaches for training changes were considered.

The second approach was to examine and augment the feedback and interface in the cockpit. This can allow the system to become more observable and tractable, thereby reducing the likelihood of errors caused by confusion as to the behaviour of the automation. An evaluation of mode awareness problems resulted in the design and evaluation of the Electronic Vertical Situation Display. The results of the evaluation were that the EVSD had a significant impact in mode awareness problems where the vertical feedback required augmentation. Obviously the cost and retraining required by the addition of new displays is significantly higher than for procedural modifications.

The third approach was to consider a new development process for automation with which humans interact. The Operator Directed Process was put forward as an approach which allowed the explicit consideration of the human operator early in the development process. This consideration of human operators can serve to limit system complexity by acknowledging human capabilities. It is hypothesized that this could lead to improvements in safety through more accurate certification, and increased usability through reduced training time, improved performance and predictability.

Chapter 7

Conclusions

There are three primary conclusions to this work. The first is that is has been shown that systems are growing more complex and are doing so in the absence of a consistent global model. Furthermore, as these systems grow, humans may become the limiting factor as the operators become less able to deal with the burgeoning complexity. The third is that humans' limitations should be acknowledged early in the design, through mechanisms such as the Operator Directed Process, to guide development. However, it remains to be seen if an approach of this sort will be adopted.

The development of aircraft automation is a testament to the engineering and designers capability of making highly complex systems. This thesis has considered the development of aircraft autoflight systems evolution from multiple perspectives. This evolutionary growth of these systems has been documented along multiple axis: number of displays, number of modes, size of software and others. In most situations, this complexity is only an issue during design and maintenance rather than during operation. However, if the software is to be interacted with in a life- and time-critical situation, such as in aerospace, a mechanism needs to be put in place in order to constrain the complexity of the resultant system. Without such a mechanism, the software will grow in complexity to the point where it becomes a liability for operational use.

The approaches that can be used to manage complexity are all based, at some level, on making assumptions based on a consistent set of behaviours. Unfortunately, it does not appear that there is a consistent global model of automation to exploit in order to allow reductions in system complexity. The lack of such a model may result in the complexity management techniques affecting a loss of understanding of the system. As such, it is felt that the unconstrained growth of these systems may be contributing to the autoflight system safety concerns that are emerging in modern aircraft.

As a related note, those issues which are appearing in aircraft may be the leading edge of problems in other environments. In some sense, aircraft are among the *best* places for these

automation issues to appear, since the industry has access to the resources to deal with the problems through training and system redesign. The heavy regulation in aerospace also allows the imposition of new procedures to work around problem. Contrast this with the automobile industry where the training of operators (i.e., drivers) is negligible. In addition, the regulation of the behaviour of the individual driver is largely non-existent. The only mechanism which can be used by automobile manufacturers is to publicize recall notifications and wait for owners to voluntarily have the systems fixed. Unless some of these automation issues are dealt with in scenarios where significant resources can be brought to bear, they are unlikely to be fixed in situations where resources are more constrained. It is important to consider these issues now, before a wider population becomes subjected to these problems of complexity.

One of the goals of this thesis was to examine the complexity of autoflight systems and, in particular, of transitions between modes. The fact that no consistent global model of autoflight systems was found resulted in the creation of the Hybrid Automation Representation (HAR). The HAR serves to segment the system in a manner that appears to be somewhat consistent with the viewpoints of both the pilots and the engineers. However, the underlying concern, the lack of a consistent global model, still exists—the necessity of creating the HAR underscores this issue. In addition, the HAR is an inappropriate representation for pilots. It is based on the existing automation architecture and becomes ungainly when attempting to describe complex interactions.

Within the framework of the Hybrid Automation Representation, it was shown that cyclomatic complexity could be extended to apply to autoflight mode transitions. This allowed the analysis of transitions using cyclomatic complexity. A survey of pilots showed that there was a relationship between the cyclomatic complexity and those mode transitions which they felt to be most complex. While not exhaustive or conclusive, the results from the survey shows two relevant relationships. The first is that mode transitions which were identified by pilots had a higher cyclomatic complexity than the average of a large set of modes. In addition, pairwise comparison results demonstrated that pilots found that mode transitions with higher cyclomatic complexity were more complicated that those with lower cyclomatic complexity.

Based on these results, designs which have higher cyclomatic complexity—more transitions and sections—are hypothesized to be more difficult to monitor and have behaviour which is less

predictable. This implies that pilots may need to be the constraint on the size and complexity of autoflight systems. One of the future areas that may be explored as a continuation of this work is to determine which elements of system design are most critical for pilots' understanding and ability to monitor. If the ability to monitor decreases precipitously after a certain number of modes, there may be hard limit on the number of modes which can be used in autoflight systems. If the number of transitions, or number of a specific type, is found to be a limiting factor, designers can use that information to develop more tractable systems.

This is not meant to imply the cyclomatic complexity or mode count can or should be used as the sole means to characterize the complexity of systems. Other measures may also exist which are more appropriate. Rather they are tools which can be used to call attention to transitions which may cause confusion. There are certainly other characteristics, including feedback, mental models, and training, which can serve to alleviate or exacerbate the perceived complexity of autoflight systems. In fact, there is a critical need to continue research to investigate other elements which contribute to the perceived complexity of systems.

As a leading indicator for supervisory automation, it appears that aerospace is currently nearing a point where the system will become too complex for humans to effectively monitor and utilize. This needs to be dealt with before the next generation of aircraft is designed. It is anticipated that considering the human early in the development cycle will become increasingly critical as autoflight systems gain in complexity and autonomy. The Operator Directed Process has been explored in this work as one approach to allow early consideration of the operator. This process showed promise in a preliminary utilization. Unfortunately, approaches which consider the human early in the process have not been widely adopted. It remains to be seen if the problems which have been appearing will spur this adoption, or whether it may be required through a regulatory body.

References

Ammar, R. A., Booth, T. (1988). Software Optimization Using User Models, *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 18, No. 4. pp 552-560.

American Airlines (1997), DC-9 Operating Manual.

American Airlines (1994), Boeing 727 Operations Manual.

American Airlines (1996), Boeing 757/767 Operating Manual Bulletin No. 757/767-19, Date 10/31/1996.

American Airlines (1997), A300-600 Operations Manual.

Aviation Safety Net (2000), http://www.aviation-safety.net/database/1995/951220-1.htm

Boehm, Barry W. (1981), Software engineering economics. Englewood Cliffs, N.J.: Prentice-Hall.

Boeing Data (1998), *Hull Loss Accident Rates 1959-1997*. Boeing Commercial Aircraft Company, Seattle, Washington.

Boeing (1985), *Boeing 737-200 Operations Manual*.

Boeing (1985), *Boeing 747-200 Operations Manual*.

Boeing (1986), *Boeing 757 Operations Manual*.

Boeing (1989), *Boeing 737-300/-400/-500 Operations Manual*.

Boeing (1993), *Boeing 767-200 Operations Manual*.

Boeing (1997), *Boeing 777-200 Operations Manual*.

Boy, Guy A. (1998), *Cognitive Function Analysis (Contemporary Studies in Cognitive Science and Technology, V. 2)*. Ablex Pub Corp.

Brooks, Frederick P. (1975), *The mythical man-month: essays on software engineering*. Reading, Mass.: Addison-Wesley Pub. Co.

Callantine, T. J., and Mitchell, C. M. (1994), A methodology and architecture for understanding how operators select and use modes of automation in complex systems. *Proceedings of the 1994 IEEE Conference on Systems, Man, and Cybernetics*, San Antonio, TX, 1751-1756.

Card, S., Newell A., Moran, T. (1983), *The psychology of human-computer interaction*. Hillsdale, N.J.: L. Erlbaum Associates.

Chappell, A. R. & Mitchell, C. M. (1997), Intelligent tutoring systems for 'trained novices': Building expertise in mode management. *Proceedings of the 1997 International Symposium on Aviation Psychology*, Columbus, OH.

Corwin, W. (1995), *Investigation of Operational Problems Associated with Flight Mode Annunciation*. NASA Grant NAG1-1581.

Crowther, E.G., Chappell, A.R., Mitchell, C.M. (1994), VNAV Tutor: System Knowledge Training for Improving Pilots' Mode Awareness. *Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics*.

Curran, J. (1992), *Trends in Advanced Avionics*. Iowa State University Press.

Chu, R., Mitchell, C., Jones, P. (1995), Using the Operator Function Model and OFMspert as a Basis for an Intelligent Tutoring System: Towards a Tutor/Aid Paradigm for Operators of Supervisory Control Systems, *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 25, No. 7, pp 1054-1075.

Daniel, W. (1990), *Applied Nonparametric Statistics.*, PWS-KENT Publishing.

Degani, A. (1994), *Modeling Human-Machine Errors: on Modes, Error and Patterns of Interaction*. Doctoral Thesis, Atlanta, GA: Georgia Institute of Technology.

Degani, A., Weiner, E. (1997), Procedures in Complex Systems. Modes in Human-Automation Interaction: *IEEE Transactions on Systems, Man and Cybernetics*. Vol 27, No 3.

Degani, A., Kirlik, A. (1995), Modes in Human-Automation Interaction: Initial Observations About a Modeling Approach. *IEEE International Conference on Systems, Man and Cybernetics*. Vancouver, Canada.

Degani, A., Shafto, M., Kirlik, A. (1999), Modes in Human-Machine Systems: Constructs, Representation and Classification. *The International Journal of Aviation Psychology* 9, no. 2: 125-138.

Denning, S. (1980), On Learning how to Predict. *Proceedings of the IEEE*, vol. 68 no. 9: 1099-1103.

Domheim, M. (1995), Dramatic Incidents Highlight Mode Problems in Cockpits, *Aviation Week & Space Technology*. January 30, 1995.

Domheim, M. (1995), Modem Cockpit Complexity Challenges Pilot Interfaces, *Aviation Week & Space Technology*. January 30, 1995.

Eldredge, D. et al (1992), *A Review and Discussion of Flight Management System Incidents Reported to the Aviation Safety Reporting System* (DOT/FAA Report DOT/FAA/RD-92/2).

Endsley, M. (1995), Towards a Theory of Situational Awareness in Dynamics Systems, *Human Factors 37*, no. 1: 32-64.

Endsley, M. (1995), Measurement of Situational Awareness in Dynamic Systems, *Human Factors 37*, no. 1: 65-84.

Fadden, Braune, and Wiedemann, 1988. *Spatial Displays as a Means to Increase Pilot Situational Awareness*, Boeing Commercial Aircraft Company, Seattle, Washington.

Feary, Barshi, Sherry, Alkin (1999), "What is it doing that?" Perspectives on the FMS. *The Tenth Biennial International Symposium on Aviation Psychology*. Columbus, USA.

Feary, M. et al. (1998), *Aiding Vertical Guidance*. NASA/TM-1998-112217.

Federal Aviation Administration (1996), FAA Report on the Interfaces Between Flightcrews and Modem Flight Deck Systems.

Federal Aviation Administration (1998), Federal Aviation Regulations.

Fishbein, S. (1995), *Flight Management Systems: The Evolution of Avionics and Navigation Technology*, Praeger.

Funk, K. et al (1999), Flight Deck Automation Issues. *The International Journal of Aviation Psychology* 9, no. 2: 109-124.

Gentner D., Stevens, A. L., (1983). *Mental Models*, S. Hillsdale, NJ: Lawrence Erlbaum Associates.

Godbole, D., Lygeros, J., Shastry, S., Hierarchical Hybrid Control: a Case Study.

Green, T. R. G. (1982), Pictures of programs and other processes, or how to do things with lines. *Behaviour and Information Technology.* Vol 1, No 1, pp 3-36.

Hansman, R.J., Kuchar, J., Pritchett, A., Clarke, J.-P., Vakil, S., and Barhydt, R. (1997), "Integrated Human Centered Systems Approach to the Development of Advanced Air Traffic Management Systems," *Eurocontrol ATM '97 R&D Seminar*, Saclay, France.

Harel, D. (1987), Statecharts: A Visual formalism for Complex Systems. *Science of Computer Programming*, 8, 231-274.

Honeywell (1992), *MD-11 Cockpit Pilot's Guide.*

Honeywell (1994), *Boeing 747-400 Flight Management System Pilot's Guide.*

Honeywell (1992), *A320 Flight Management System Pilot's Guide.*

Hughes, D. (1995), Accidents Direct Focus on Cockpit Automation, *Aviation Week & Space Technology.* January 30, 1995.

Hughes, D. (1995), Incidents Reveal Mode Confusion, *Aviation Week & Space Technology.* January 30, 1995.

Hutchins, E. (1996), *Immediate Mode Management Interface Final Report.* NASA Ames Research Center.

Jager, M. A., Tenney, Y. J., Pew, R. W. (1995), *Situation Awareness and the Cognitive Management of Complex Systems.* Human Factors, 1995, 37 (1), pp 84-105.

Javaux, D. (1997), Measuring Cognitive Complexity in Glass-Cockpits. A Generic Framework and its Application to Autopilots and their Modes. In R. Jenson & L. Rakovan (eds) *Proceedings of the 9th Symposium on Aviation Psychology*, Columbus, Ohio.

Javaux, D. (1997) The Cognitive Complexity of Pilot-Mode Interaction. A Possible Explanation of Sarter and Woods's classical results. In G. Boy & C. Graeber (eds), *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics*, May 27-29, Montreal, Canada.

Javaux, D. and De Keyser, V. (1998), The Cognitive Complexity of Pilot-Mode Interaction. A Possible Explanation of Sarter and Woods's classical results. In G. Boy & C. Graeber (eds), *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics*, May 27-29, Montreal, Canada.

Javaux, D. (1999b), Personal Communication.

Johnson-Laird, P.N. (1983), *Mental models.* Cambridge, MA: Harvard University Press.

Johnson-Laird P. N. (1988), *The Computer and the Mind: an Introduction to Cognitive Science.* Cambridge, MA:Harvard University Press.

Lehman, M. et al. (1980). Programs, Life Cycles, and Laws of Software Evolution. *Proceedings of the IEEE,* Vol. 68, No. 9.

Lehman, M. et al. (1996). *Metrics and Laws of Software Evolution—The Nineties View.*

Lenorovitz, J. (1990), Indian A320 Crash Probe Data Show Crew Improperly Configured Aircraft., *Aviation Week & Space Technology.* January 30, 1995.

Leveson, N.G., Palmer, E. (1997), Designing Automation to Reduce Operator Errors. *Proceedings of Systems, Man, and Cybernetics Conference.*

Leveson, N.G. (1997), Computers and Trust: How Far Can and Should We Rely on Them. *Minta Martin Lecture,* MIT.

Leveson, N., (1998), Intent Specifications: An Approach to Building Human-Centered Specifications. *4th IEEE International Conference on Requirements Engineering.* Colorado Springs, USA.

Leveson, N., Palmer, E. (1998), Designing Automation to Reduce Automation Error.

Littman, D. et al. (1987), Mental Models and Software Maintenance. *The Journal of Systems and Software.* Vol 7, pp 314-355.

Mark, W. (1986), *User Centered System Design.* Ed. Norman D., Draper, S. Hillsdale, NJ: Lawrence Erlbaum Associates.

McCabe, T. (1976), A Complexity Measure, *IEEE Transactions on Software Engineering* 2, no. 4: 308-320.

McRuer, Duane T. (1973), *Aircraft dynamics and automatic control.* Princeton, N.J., Princeton University Press,

Mecham, M. (1994), Autopilot Go-Around Key to CAL Crash. *Aviation Week & Space Technology.* May 9, 1994.

Menendez, J. (1997), *Building Software Factories in the Aerospace Industry.* Master's Thesis. MIT, Cambridge, MA.

Midkiff, A. (1998), Personal Communication.

Ministry of Planning, Housing, Transportation and Space (1992), *Preliminary Report Relative to the Accident which Occurred on Jan. 20, 1992 near Mon Sainte-Odile (BAS-Rhine) to the Airbus A320.*

Modugno, F., Leveson, N., Reese, J. (1997), Integrated Safety Analysis of Requirements Specifications. *3rd IEEE International Conference on Requirements Engineering.* Annapolis, USA.

Molloy, R., Parasuraman, R. (1996). Monitoring an Automated System for a Single Failure: Vigilance and Task Complexity Effects. *Human Factors,* 38(2), pp 311-322.

Morris, P. E. (1987), *Modeling Cognition.* Chichester; New York: Wiley.

National Transportation Safety Board (1999), *Safety Recommendations A-99-39 to A-99-34.*

Norman, D. (1989), *The Design of Everyday Things.* New York, NY: Basic Books.

Orr, Julian E. (1996), *Talking about Machines: an Ethnography of a Modern Job*. Ithaca, N.Y.: ILR Press.

Ostgaard, M.A. (1981), Multifunction Cockpit Display Considerations. *AGARD Guidance And Control Panel Symposium*, Stuttgart, Germany.

Palmer, E. A., Hutchins, E. L., Ritter, R. D., and vanCleemput, I. (1993, October), Altitude deviations: Breakdowns of an error tolerant system. *NASA Technical Memorandum 108788*. Moffett Field, CA: NASA Ames Research Center.

Palmer, E. (1996), "Oops, it Didn't Arm" - a Case Study of Two Automation Surprises. In R. Jenson & L. Rakovan (eds) *Proceedings of the 9th Symposium on Aviation Psychology*, Columbus, Ohio.

Parasuraman, R. (1997), Humans and Automation: Use, Misuse, Disuse, Abuse. *Humans Factors*, 39(2), pp 230-253.

Parker and West (eds), (1973), *Bioastronautics Data Book (2nd Edition)*. NASA SP-3006.

Peters, L. (1980), Software Representation and Techniques. *Proceedings of the IEEE*, Vol. 68, No. 9.

Phillips, E. (1994), NTSB Links Detection-Delay Feature with DC-9 Crash, *Aviation Week & Space Technology*. October 10, 1994.

Phillips E. (1995), NTSB: Mode Confusion Poses Safety Threat, *Aviation Week & Space Technology*. January 30, 1995.

Polson, P. (1995), *Investigation of Operational Problems Associated with Flight Mode Annunciation* (FMA), NASA Grant NAG1-1581.

Port, O. and Armstrong, L. (1998), Your car may be smarter than you. *Business Week*. June 29, 1998.

Pritchett, A., Hansman, R.J., and Johnson, E. (1995), Use of Testable Responses for Performance-Based Measurement of Situation Awareness, *International Conference on Experimental Analysis and Measurement of Situation Awareness*.

Rasmussen, J. (1986), *Information Processing and Human-Machine Interaction: an Approach to Cognitive Engineering*. New York: North-Holland.

Reason, J. (1990), *Human error*. Cambridge, England. New York: Cambridge University Press.

Richards, W. (1997), Personal Communication in Cognitive Artifacts and Architectures class.

Riley, V. (1995), Presentation at the Massachusetts Institute of Technology.

Riley, V., DeMers, B., Misiak, C., (1997), The Cockpit Control Language.

Rouse, W. B., Morris, N. M., (1986), On Looking Into the Black Box: Prospects and Limits in the Search for Mental Models. *Psychological Bulletin*, Vol. 100, No. 3, pp 349-363.

Rouse, W. B., et al (1992), The Role of Mental Models in Team Performance in Complex Systems. *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 22, No. 6 pp 1296-1308

RTCA Document DO-178B (1992), *DO-178B Software Considerations in Airborne Systems and Equipment Certification*.

Sarter, N. and Woods D. (1992), Pilot Interaction with Cockpit Automation: Operational Experiences with the Flight Management System, *The International Journal of Aviation Psychology 2*, no. 4: 303-321.

Sarter, N. and Woods D. (1994), Pilot Interaction with Cockpit Automation II: An Experimental Study of Pilots' Model and Awareness of the Flight Management System, *The International Journal of Aviation Psychology 1*, no. 1: 1-28.

Sarter, N. and Woods D. (1995), How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control, *Human Factors 37*, no. 1: 5-19.

Sarter, N. and Woods D. (1995), Strong, Silent and 'Out-of-the-Loop': Properties of Advanced (Cockpit) Automation and their Impact on Human-Automation Interaction, Cognitive Systems Engineering Laboratory. *CESL Report 95-TR-01*.

Sarter, N. and Woods D. (1997), Team Play with a Powerful and Independent Agent: Operational Experiences and Automation Surprises on the Airbus A-320. *Human Factors 39*, no. 4: 553-569.

Sharit, J. (1987), The Use of Measures of Entropy in Evaluating Human Supervisory Control of a Manufacturing System, *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. SMC-17, No. 5. pp 815-820.

Sheridan, T. (1992), *Telerobotics, Automation and Human Supervisory Control*, MIT Press.

Sherry, L., Polson, P. (1995), A New Conceptual Model for Avionics Annunciation, *Institute of Cognitive Science Technical Report 95-08*.

Sherry, L., Polson, P. (1999), Shared Models of Flight Management System Vertical Guidance, *The International Journal of Aviation Psychology 9*, no. 2: 139-153.

Shneiderman, B. (1987), *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, Mass.: Addison-Wesley.

Sparaco, P. (1994), Human Factors Cited in French A320 Crash, *Aviation Week & Space Technology*. January 3, 1994.

Sparaco, P. (1994), A330 Crash to Spur Changes at Airbus, *Aviation Week & Space Technology*. August 8, 1994.

Sparaco, P. (1994), A310 Inquiry Target FCU, *Aviation Week & Space Technology*. October 10, 1994.

Sparaco, P. (1994b), French Issue Airbus Alert, *Aviation Week & Space Technology*. October 24, 1994.

Tenney, Y. et al (1995), *Pilot Opinions on High Level Flight Deck Automation Issues: Toward the Development of a Design Philosophy*, Contractor Report 4669, NASA Langley Research Center.

Tognazzini, Bruce. (1992), *Tog on Interface*. Addison-Wesley.

UAE (1998), *United Arab Emirates A330 Flight Crew Operating Manual*.

US Airways (1998), *A319/320/321 Flight Crew Operations Manual*.

Vande Vegte, J. (1990), *Feedback control systems*, Englewood Cliffs, N.J.: Prentice-Hall.

Vakil, S., Hansman, R.J., and Midkiff, A. (1995), "Feedback Mechanisms to Improve Mode Awareness in Advanced Autoflight Systems," *Eighth International Symposium on Aviation Psychology.*

Vakil, Hansman, Midkiff, Vaneck, (1995b), Mode Awareness in Advanced AutoFlight Systems. *Proceedings of the 6th IFAC Symposium.*

Vakil, S., Hansman, R.J., and Midkiff, A. (1995c), "Impact of Vertical Situation Information on Vertical Mode Awareness in Advanced AutoFlight Systems," 14th AIAA/IEEE Digital Avionics Systems Conference.

Vakil, S., Hansman, R.J., and Midkiff, A. (1996), "Preliminary Results of the Experimental Evaluation of a Prototype Electronic Vertical Situation Display," 8th European Aviation Safety Seminar, Amsterdam, The Netherlands.

Vakil, S., Hansman, R.J. (1996b) "Development and Evaluation of an Electronic Vertical Situation Display" *MIT Aeronautical Systems Laboratory Report*, ASL-96-2.

Vakil, S., and Hansman, R.J. (1997), "Predictability as a Metric of Automation Complexity," *Human Factors & Ergonomics Society 41st Annual Meeting.*

Vakil, S., Hansman, R.J., (1998) "Understanding Automation Structure by Analyzing Mode Transition Matrices," *17th Digital Avionics Systems Conference.* Seattle, USA.

Vicente, Kim J. (1999), *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work.* Lawrence Erlbaum Assoc.

Wanke, C. (1995), *Jet Transport Autoflight System Design: MCP Mode Control Logic and Command Window Setting Specification.*

Weener, E. (1998), Integrated Safety Systems Design and Future Developments of Large Airplanes. *Keynote Presentation at the 2nd Workshop on Human Error, Safety, and System Development.*

Weiner, E. and Nagel, D. ed. (1988), *Human Factors in Aviation*, Academic Press.

Weiner, E., (1993), Intervention Strategies for the Management of Human Error. *NASA Contractor Report 4547.*

Weiner, E., Chute, R., Moses, J. (1999), Transition to Glass: Pilot Training for High Technology Transport Aircraft. *NASA Contractor Report*, NASA/CR-1999-208784.

Wertz, J., Larson, W., (1991), *Space Mission Analysis and Design.* Dordrecht, Netherlands; Boston: Kluwer Academic.

Wilson, N. R., Rutherford, A. (1989), Mental Models: Theory and Application in Human Factors. *Human Factors* 31(6), pp 617-634.

Yeh, R., Zave, P. (1980). Specifying Software Requirements. *Proceedings of the IEEE,* Vol 68, No. 9, pp 1077-1085.

Ziegler, B., *How to Fly Windshear.* Airbus Industrie.

Appendix A

Known Accidents / Incidents and Automation Concerns

Table A.1 is a partial list of incidents and accidents which are suspected to have automation as a contributory factor. Accidents are defined as occurrences associated with the operation of aircraft that result in: a person being fatally or seriously injured; the aircraft sustaining damage or structural failure that adversely affects the structural strength performance, or flight characteristics of the aircraft and would normally require major repair or replacement of the affected component; the aircraft becoming missing or completely inaccessible. Incidents are occurrences, other than accidents, associated with the operation of aircraft that affect or could affect the safety of operation (FAA, 1996).

While the table lists the multiple aircraft types which are affected by this concern, the majority of incidents are in more recent aircraft with advanced cockpit automation. Multiple airframe and, though less apparent avionics manufacturers appear in the set of accidents. The Airbus A320 and the Boeing B757/767 are among the latest generation of aircraft and figure prominently in Table A.1. Incidents involving automation also appear to be more numerous in recent years. This can be partially attributed to the growing population of modern aircraft in service while older aircraft become retired.

It has also been conjectured that the causal factors underlying the human errors have changed. An example of previously observed error is a stall accidents, such as those on the B707, suspected to have been caused in part by poor handling characteristics. Recent incidents appear to have the automation contributing to the error. As an example, consider the A320 accident at Bangalore where the inappropriate use of an automation segment led to a loss of airspeed and ultimately the crash.

Table A.1: Examples of Incidents and Accidents (adapted from the FAA Report on The Interfaces Between Flightcrews and Modern Flight Deck Systems)

Date	Location	Aircraft Type	Operator	Description
7/31/73	Boston	DC-9-31	Delta Air Lines	Flightcrew was preoccupied with questionable information presented by the flight director. Fatal crash.
2/28/84	New York	DC-10-30	Scandinavian Airlines	Malfunctioning autothrottle; airplane overran runway, minor injuries
2/19/85	San Francisco	B747SP	China Airlines	Autopilot masked approaching onset of loss of control after loss of power on one engine. Airplane went into unusual attitude high speed dive, but was successfully recovered.
6/26/88	Habsheim	A320	Air France	Low, slow flyover at air show. Possible overconfidence in the envelope protection features of the A320. Fatal crash.
7/3/88	Gatwick	A320	unknown	Intended for 3 degree flight path; inadvertently in vertical speed mode, almost landed 3 miles short.
6/8/89	Boston	B767	unknown	Airplane overshot the localizer; confusion led to vertical speed mode commanding an 1 800 fpm rate of descent. Go-around from about 500 feet.
2/14/90	Bangalore	A320	Indian Airlines	Inappropriate use of open descent mode. Fatal crash.
6/90	San Diego	A320	unknown	Pilot mistakenly set vertical speed of 3 000 fpm instead of 3.0° flight path. Error was caught after serious altitude deviation.
2/11/91	Moscow	A310	Interflug	Pilot intervention in autopilot coupled go-around resulted in airplane badly out of trim and several extreme pitch oscillations before regaining control.
1/20/92	Strasbourg	A320	Air Inter	Flightcrew inadvertently selected 3 300 fpm descent rate instead of 3.3° flight path. Fatal crash.
9/13/93	Tahiti	B747-400	Air France	The flightcrew lost directional control of the airplane as the speed decreased and the airplane went off the right side of the runway due to autothrottle confusion
9/14/93	Warsaw	A320	Lufthansa	After touchdown, the air/ground logic delayed deployment of ground spoilers and reversers. Airplane overran runway. Two fatalities.
4/26/94	Nagoya	A300-600	China Airlines	Flightcrew inadvertently activated the go-around which led to a stall. Fatal crash.
6/21/94	Manchester	B757-200	Britannia	Altitude capture mode activated shortly after takeoff, airspeed dropped toward V_2 before flightcrew pitched the nose down to recover.
6/30/94	Toulouse	A330	Airbus	Unexpected mode transition to altitude acquire mode during a simulated engine failure. Fatal crash.
9/24/94	Paris - Orly	A310-300	Tarom	Overshoot of flap placard speed during approach caused a mode transition to flight level change. Airplane stalled, but was recovered.
12/20/95	Cali	B757-200	American Airlines	Confusing database information led to incorrect waypoint target with a collision course with a mountain. Fatal crash.

Appendix B

Predictability and Measures of Modal Structures

One of the key concerns for the design of future aircraft automation systems and retrofits/modifications to existing systems is to establish a metric to evaluate designs. Previous research has focused on identification of the elements of automation (mode structure, consistency, command languages and others) which may lead to faulty human-automation interactions. These approaches require the complex system to have underlying structure in an available and communicable form. The incremental development of automation in aircraft has created a system with limited structure. This section will present a more easily testable "end-to-end" metric which can be used independent of structural knowledge. The concept of predictability is presented as a candidate metric of the complexity of automation and is defined as a measure of how well an operator can anticipate what the system will do at some point in the future. In essence, this is a measure of the complement of how often a system will "surprise" an operator by acting in an unanticipated manner.

B.1 Existing Measures

Various metrics were drawn from the human factors community, the psychology community and the software design community to evaluate of complex automated systems. A great deal of research has centered on the operator's internal representation (mental model) of the automation (Morris 1987, Johnson-Laird 1988). Unfortunately, this representation tends to be dynamic, implicit and uncertain, leading to difficulty in extracting the model in a systematic and reproducible manner for examination. More recent work has examined the automation to understand how mode structure and consistency affect complexity (Sarter 1994). These techniques examine the number of inconsistencies in the underlying structure of the automation to measure system complexity. Lexical analysis of command languages (Riley 1995) has been completed on various complex systems. The required number of nouns, verbs, modifiers and lexical combination mechanisms could be used as a metric of system complexity. Rouse has used hierarchical decomposition to measure the size of layers and number of branches in well structured systems (Rouse, 1986). These metrics could be used as an a priori measure of system

191

complexity. Finally, graphical measures of software complexity based on McCabe's measure have been used to determine the number of non-reducible flowchart elements (McCabe, 1976).

There are two problems common to the metrics discussed above. The first is that many of these metrics measure system complexity independent of the operator interface. Since this interface can have a detrimental or a beneficial effect on operator performance, and, hence, on perceived complexity, it must somehow be included in a candidate metric. The second problem is that many of the metrics depend on knowledge of the underlying model of the automation, and on the model being well structured. In the particular case of aircraft automation, a global model is never presented to the operator through the training material, so any candidate metric of its complexity must be insensitive to this absence.

B.2 Predictability

In contrast predictability may provide a more easily testable "end-to-end" metric which can be used independent of the knowledge of the underlying structure. The concept of predictability is defined as a measure of how well an operator can anticipate what the system will do at some point in the future. In essence, this is a measure of the complement of how often a system will "surprise" an operator by acting in an unanticipated manner. This metric is completely independent of an underlying structure to the automation, and may be applicable when this structure does not exist or is not available.

There are two distinct components postulated to contribute to the predictability of a system – one related to the operator and one to the automation itself. The former is determined by how well the operator can predict the output of the automation given the observable information. This component is directly related to the training, competency and experience of the operator.

The latter, automation behaviour, determines how predictable the system would be with an operator capable of perfectly interpreting the sensors and acting on them in an optimal manner. In one sense, this is a measure of the intrinsic observability of the underlying system when it is filtered through the interface of the automation, an idea related to Norman's "Gulf of Evaluation" (Norman 1989). Behaviour of the automation which is non-observable, or ambiguous, may reduce predictability even with a perfect operator.

In addition, it is hoped that predictability could be used in the capacity of both an *a priori* and *a posteriori* design evaluation. In an *a priori* evaluation, predictability will enable designers to determine the predictability of paper designs and assess the impact of design decisions on specific elements of the automation. Once the design has been implemented, predictability will provide a means of measurement of the new or existing system to determine problematic regions and to provide insight for operator training.

B.2.1 Experimental Evaluation of Predictability

In order to evaluate this metric, a small, manageable experiment was performed comparing the predictability of two standard types of calculators. The first type was a "Four Function" (FF) calculator. These are the standard and ubiquitous calculators that have the four simple operators (+, -, x, /), and an equals (=) key, and accept calculations in the "in-fix" form: "3 + 5 =." Unfortunately, these calculators also have some "implicit" functionality and are designed to assume arguments in situations where the user does not explicitly state them, thereby ideally anticipating and correcting for user error. This behaviour is hypothesized to reduce predictability. The second type of calculator was a stack based "Reverse Polish Notation" (RPN) calculator, used primarily in the engineering world. These require more training, but have a very consistent set of responses to calculations in the form "3 Enter 5 +." This consistency is gained by requiring the user to format the calculation in a very specific manner.

The experimental design consisted of having subjects predict the calculator response to an exhaustive set of four keystroke sequences and evaluating their prediction accuracy based on the number of correct responses predicted. Subjects consisted of graduate students in science and engineering. To reduce the size of the experiment, a reduced set of calculator keys was used. Extraneous keys, such as Clear Entry and the decimal marker were not introduced. To maintain similar levels of calculation complexity, RPN calculators were limited to only two levels of stack. In addition, the calculator was assumed to be cleared between sets of keystrokes. Subjects were only tested on the calculator type with which they were most familiar, and were not permitted to use a calculator to determine the response. Examples of keystroke sequences are listed in Table B.1. Note that many of these keystrokes are nonsensical and would not appear in regular

calculator usage. By exhaustively testing the "keystroke-space" of each type of calculator abnormal and rare situations were tested in addition to those with which users are familiar.

Table B.1:   Example Keystroke Sequences

Four Function	RPN
= = = 8	4 X E E
X = 3 -	1 E E 6
5 + 3 -	2 3 6 E
X - 8 /	- 5 X 3

B.2.2   Experimental Results

A total of 22 subjects were tested, 11 RPN users and 11 Four Function users. Even with this small subject set, the mean correct response rate was found to be higher for RPN users (89.8%) than for Four Function users (80.7%) at a 95% confidence level. To ascertain whether mathematical aptitude was a factor in the difference, subject math SAT scores were analyzed. No significant differences were found.
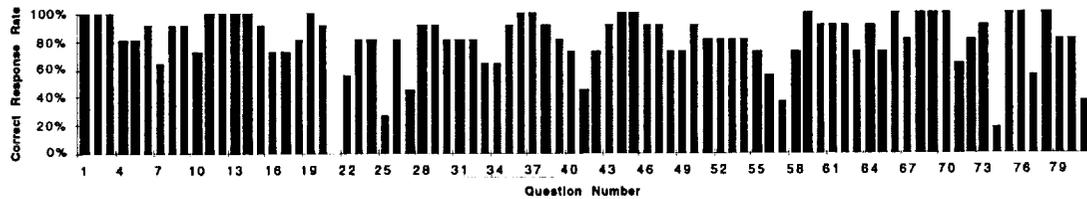
Figure B.1:   Four Function Calculator Response Profile
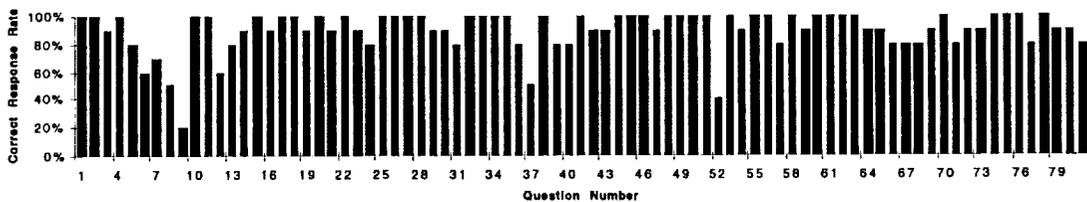
Figure B.2:   RPN Calculator Response Profile

More interesting than the simple accuracy of subject responses was the element of regularity in errors that were made. Specific keystroke sequences caused difficulty for a majority of

subjects. Figure B.1 and Figure B.2show the correct response rate of subjects to each of the keystroke sequences posed. Note that certain sequences, namely 21, 57 and 74 in the Four Function set and 9, 37 and 52 in the RPN set have very few correct responses. The keystroke sequences in each of these cases are detailed in Table B.2, along with the correct answers. While these answers are not obvious from the standpoint of using a calculator, they appear more obvious in the context of the underlying structure of the calculator.

Table B.2:   Error Prone Keystroke Sequences

Four Function	RPN
- 6 / = [-1]	6 E - E [6]
- 3 X - [-3]	4 X E E [4]
4 X = = [64]	1 X E / [1]

In analyzing these errors, three separate causes were tentatively identified: *implicit arguments*, *operator overloading* and *error management*.

Implicit arguments have already been identified as one of the elements of a Four Function calculator that lead to lower predictability. Sequences such as "3 0 + =," where the calculator would assume a second argument of "0" for a response of "30" caused common errors. Similar situations where a first argument was neglected, "/ 2 0 =," where a zero (or sometimes a one) was assumed to be the first argument causing a response of "0" were also problematic.

Operator overloading consisted of situations where additional operators were added at odd times: "9 X - =." In this situation, the subtract overrides the multiplication, and causes the calculator to assume that "9 - 0 =" was the intended calculation, leading to a result of "9."

Error management was the single largest error on the RPN calculators. Users were often not clear what an error-inducing keystroke would do to the rest of operands already in the calculator. In practice, the keystroke following an induced error would clear the error and be executed. No "Clear Error" mechanism was necessary. Keystrokes such as "6 E - E," which caused an error part way through would have the error cleared by the later keystrokes.

In summary, RPN calculators were shown to be statistically more predictable than Four Function calculators. Attributes that lead to loss of predictability included implicit functionality

and poorly defined behaviour: operator overload on Four Function, and error management on RPN.

B.2.3 Mode Transition Diagram Representations

The preceding results discuss the use of predictability in an a posteriori evaluation of the selected calculators. In addition, a mode transition diagram analysis was done of the calculators in an attempt to capture their underlying structure and functionality. In these diagrams, individual modes are named A-D. Associated with each mode is the state of the calculator. Transition criteria consist of the keystrokes which are typed into the keypad. These are shown are shown on the arrows between the modes, with particularly problematic transitions called out in grey.

Figure B.3 shows the mode transition diagram empirically derived for the Four Function calculator. The Four Function calculator requires three elements in order to do a calculation: two arguments and an operator. Starting in Mode A, the user is expected to enter the first argument (Arg1). This argument is modified in Mode B until an Operator keystroke is pressed, leading to Mode C. At this point, the calculator expects another number to define Arg2. Once Arg2 has been entered and modified (Mode D), user presses the "=" key to finish the calculation, placing the result in Arg1 and leaving Arg2 untouched.

Figure B.3:   Four Function Calculator Mode Transition Diagram

As shown in Figure B.4, the RPN calculator is similar in requiring two arguments and an operator, but the ordering is different. Starting in Mode A, the user is expected to enter Arg1, and continue to modify it in Mode B. When complete, the user presses the "Enter" key and continues to Mode C. In a similar manner, the user enters and modifies Arg2 in Mode C and D. The Operator is only entered after both arguments have been completed, and sends the system back to Mode C, with the answer to the calculation placed into Arg1 and Arg2 is deleted.
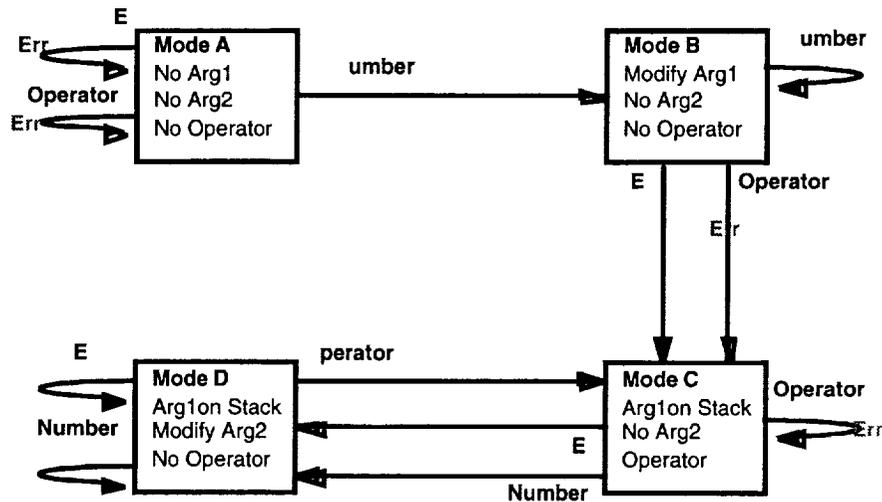
Figure B.4:   RPN Calculator Mode Transition Diagram

Note that the Four Function calculator diagram shown in Figure B.3 does not identify the events that occur on transitions, but only the keystrokes which lead to particular transitions. The transitions exhibiting implicit functionality show up as situations in which the calculator does not have enough information from the user to make a calculation, and so assumes an implicit value of zero (or sometimes one). Operator overloading occurs in the Mode C, where a user can overwrite the current operator by simply entering a new one. As an example, entering "5 + - 3 =" results in a response of "2" rather than "8."

The RPN calculator mode diagram has a significantly more linear structure, with specific entries expected to move to the next mode. Errors occur when an unexpected input is received, such as a number when an operator was expected. These transitions are problematic because the

197

response of the calculator to this error varies between treating the keystroke as an "Enter" (Modes B to C), or by not transitioning at all (Modes A and C).

B.2.4 Implications for Flight Automation

In order to evaluate complex systems (such as aircraft Flight Management Systems) a metric which is capable of evaluation independent of the knowledge of the underlying structure is needed. The calculator predictability experiment has shown that this metric may have promise in fulfilling this need. In addition, with appropriate extensions, automation elements which can lead to poor predictability situations may be able to be identified before a system is fully implemented.

Even more interesting is that the areas of difficulty observed with the calculator evaluation have direct analogies in existing Flight Management Systems (FMS). In particular, multiple flight systems in modern aircraft, including the Vertical Navigation (VNAV) system and the Mode Control Panel (MCP) have a large number of implicit behaviours and arguments.

The VNAV system on many aircraft hides the criteria it uses for engagement of overspeed and underspeed envelope protection modes from the operator, and generates these criteria based on the current flight conditions. These criteria are seldom detailed in operators manuals. VNAV also makes implicit, and often difficult to understand decisions when selecting the flight path.

The MCP has implicit limits when determining path capture criteria in transitioning from a Vertical Speed mode to Altitude Hold. In many modern aircraft, this maneuver is designed to maintain a 0.05G loading, to minimize passenger discomfort. In addition to being a non-intuitive choice of criteria from an operational standpoint, the performance of this maneuver often utilizes a mixture of simultaneous elevator and throttle control, further complicating the situation. The MCP also has problems similar to the VNAV system relating to envelope protection modes. In the 1994 incident in Orly, France, the MCP caused the aircraft to pitch up to acquire an implicit, and incorrect, altitude target late in an approach. The MCP was designed to attempt to capture the current programmed altitude in the event of a speed violation. In this incident, the crew had programmed the target altitude above them, at the Missed Approach Altitude, during final approach (as per operational guidelines). When the flaps were inadvertently lowered too early, the

198

AutoFlight System detected an overspeed and pitched the aircraft up in an attempt to capture the Missed Approach Altitude (Sparaco, 1994).

Error management is another weakness in current FMS. Most systems are ambiguous as to which mode to reacquire after an envelope protection mode has been disengaged. A specific example is that in order to maintain smooth flight near approach, once the Glide Slope has been acquired by the aircraft AutoLand System many aircraft systems will maintain the most recently acquired vertical speed in the event of the loss of signal. This is to prevent the aircraft from large attitude corrections at low altitudes caused by an intermittent signal immediately after acquisition. Unfortunately, in the event of an actual loss of signal, perhaps due to transmitter failure, the aircraft will continue its descent with minimal additional feedback to the pilot.

B.2.5 Extensions of Predictability

In its current form predictability may be a very useful metric as a conceptual tool and for testing of discrete event systems. However, the metric suffers from some limitations that will have to be overcome before it can be used on more complex systems: no consequence or context dependency, difficult to extend to continuous system, and difficulty in dealing with complex systems.

The first limitation describes the problem that predictability does not attempt to measure how important it is to accurately anticipate future events. Predicting where an aircraft will start a descent is much more important at 200 ft than at 20 000 ft, as the consequences are significantly different. The context independency of predictability makes it insensitive to the importance of accuracy. This problem can be dealt with by using some sort of weighting criteria where an experimenter creates the context dependency. Unfortunately, this will render the final analysis directly dependent on the weighting and so is highly dependent on the experimenter. One way to reduce this subjectivity would be to associate the weightings with existing operational or procedural system elements.

The limitation of continuous space systems is more fundamental. One solution may be to discretize the continuous state space in a manner consistent with the functional requirements of

the task being evaluated. Alternately, varying discretizations may be tested before the final evaluation.

Complex systems tend to break down under transition diagram analyses. A better may be to treat the system as having discrete operator-identified modes. These modes would be treated as very large, complex portions of the underlying modal structure. Since each operator-identified mode is much larger and has wide ranging implications, accurate prediction of the important implications of a operator-identified mode may be more important than simply predicting the future individual modes. The calculator example discussed in this paper has the majority of the information about its future behaviour stored in the current state, whereas an aircraft FMS has that information stored partially in the current mode and partially in the state of the environment. The experimental technique would have to be extended to capture the subjects' ability to predict both the future operator-identified modes and the associated implications.

Appendix C

Web-based Pilot Automation Complexity Survey

A survey was conducted to gain insight into those modes which pilots identify as complex in current aircraft autoflight systems. The purpose of the survey was to gain insight into those mode transitions which pilots self-identified as complex in order to examine characteristics common across those modes. The survey was placed on the Internet and accessible through the World Wide Web allowing pilots from around the world, and representing a varied cross-section of aircraft types, to take part. Notification of contacts within the industry as well as pilot automation mailing lists (notably the "Blue Coat Digest") provided access to a large quantity of subjects.

C.1   Demographic Results

A total of 105 pilots took part in the survey, with 93 valid results were generated from multiple aircraft types, as listed in Table C.1. Four military pilots and 89 commercial pilots responded. Table C.2 shows that the majority of the results were from modern "glass-cockpit" and transitional aircraft and from aircraft. Five female pilots and 84 male pilots responded; four responses did not fill out the gender field. The average age of respondents was 43.

Table C.1:   Breakdown of Responses by Aircraft Type (total n=93)

Aircraft Type	Number of Responses
Boeing B727	1
Boeing B737-100/-200	3
Boeing B737-300/-400/-500	6
Boeing B737-600/-700/-800	4
Boeing B747-400	6
Boeing B757/B767	17
Boeing B777	2
Airbus A300	2
Airbus A310	1
Airbus A320/330/340	12
Boeing MD-11	2
Other	37

Data regarding the flight hours is shown in Table C.2, and identifies the majority of respondents being experienced aviators.

Table C.2:   Flight Hours of Respondents (total n=93)

	Total Flight Hours	Hours in Current Type	Hours in 1999	Hours in 1998
Average	10 250	2 039	584	629
Maximum	27 500	10 000	2500	1 850
Minimum	150	26	0	50
Standard Deviation	5 750	2 064	310	248

47 respondents identified themselves as having the rank of captain, 33 were first officers. A small number (four) identified themselves as "Pilot In Command" (PIC). Additionally, eleven respondents identified themselves as instructors. Note that these numbers do not total the number of respondents, since individuals could have multiple positions.

C.2   Modes Identified by Respondents

Many mode transitions were identified by multiple pilots in this survey. Table C.3 show all of the modes which were identified by respondents. Note that the majority of the identified modes—70%—are in the vertical domain.

Table C.3:   Transitions Identified by Respondents

Mode	Number of Time Identified
Vertical Speed	33
Altitude Capture	31
FLCH	29
Altitude Hold	24
Heading Hold	22
LNAV	22
Approach	20
VNAV Path	16
Go Around	13
VNAV Path Descent	12
VNAV	10
ANY	7
Glideslope Capture	7

Table C.3:   Transitions Identified by Respondents

Mode	Number of Time Identified
VNAV Descent	5
Glideslope Track	4
VNAV Speed	4
VNAV Speed Descent	4
Localizer Track	3
Holding Fix	2
Localizer Capture	2
VNAV Climb	2
Autopilot OFF	1
Autopilot ON	1
CWS Climb	1
Runway	1
Speed	1
VOR Track	1

Another manner in which to examine the data is to identify which modes appear most often as the starting mode in a transition. This is an indication of which modes are most difficult to leave, but is biased towards those modes which are used most commonly. As shown in Table C.4, vertical modes dominated the starting modes.

Table C.4:   Starting Modes Identified by Respondents

Starting Mode	Number of Times Identified
FLCH	20
Vertical Speed	18
Altitude Hold	17
Heading Hold	17
VNAV Path	13
Approach	11
LNAV	7
VNAV	7
Altitude Capture	6
Go Around	5
VNAV Path Descent	3
ANY	2
VNAV Climb	2

Table C.4:   Starting Modes Identified by Respondents

Starting Mode	Number of Times Identified
VNAV Descent	2
Autopilot OFF	1
CWS Climb	1
Glideslope Capture	1
Glideslope Track	1
Holding Fix	1
Localizer Capture	1
Runway	1
Speed	1
VNAV Speed Descent	1

Conversely, Table C.5 shows those modes in which transitions end. Not surprisingly, Altitude Capture is identified the most often, since it is the mode through which one typically exists Flight Level Change or Vertical Speed before transitions to Altitude Hold.

Table C.5:   Ending Modes Identified by Respondents

Ending Mode	Number of Times Identified
Altitude Capture	25
LNAV	15
Vertical Speed	15
Approach	9
FLCH	9
VNAV Path Descent	9
Go Around	8
Altitude Hold	7
Glideslope Capture	6
ANY	5
Heading Hold	5
VNAV Speed	4
Glideslope Track	3
Localizer Track	3
VNAV	3
VNAV Descent	3
VNAV Path	3

Table C.5:   Ending Modes Identified by Respondents

Ending Mode	Number of Times Identified
VNAV Speed Descent	3
Autopilot ON	1
Holding Fix	1
Localizer Capture	1
VOR Track	1

Finally, Table C.6 shows which transitions appeared with the greatest frequency in the survey. Each of these were identified explicitly by pilots, or derived from their narrative.

Table C.6:   Most Frequently Identified Transitions

Transition	Number	Transition	Number
FLCH -> Altitude Capture	12	FLCH -> FLCH	1
Heading Hold -> LNAV	10	FLCH -> Glideslope Capture	1
Vertical Speed -> Altitude Capture	7	FLCH -> VNAV	1
Altitude Capture -> Vertical Speed	6	FLCH -> VNAV Path	1
Approach -> Go Around	6	Glideslope Capture -> Altitude Capture	1
VNAV Path -> VNAV Path Descent	5	Glideslope Track -> Go Around	1
VNAV Path -> VNAV Speed	4	Go Around -> ANY	1
Altitude Hold -> Approach	3	Go Around -> FLCH	1
Altitude Hold -> FLCH	3	Go Around -> Go Around	1
Altitude Hold -> Vertical Speed	3	Heading Hold -> Heading Hold	1
Approach -> ANY	3	Heading Hold -> Localizer Capture	1
Heading Hold -> Approach	3	Heading Hold -> Localizer Track	1
LNAV -> Heading Hold	3	Heading Hold -> VOR Track	1
Vertical Speed -> FLCH	3	Holding Fix -> LNAV	1
VNAV Path Descent -> VNAV Speed Descent	3	LNAV -> LNAV	1
Altitude Hold -> Glideslope Capture	2	LNAV -> Localizer Track	1
Altitude Hold -> Glideslope Track	2	Localizer Capture -> Heading Hold	1
FLCH -> Altitude Hold	2	Runway -> LNAV	1
FLCH -> Vertical Speed	2	Speed -> ANY	1
Go Around -> LNAV	2	Vertical Speed -> Glideslope Capture	1
LNAV -> Approach	2	Vertical Speed -> Glideslope Track	1
Vertical Speed -> Altitude Hold	2	Vertical Speed -> VNAV Descent	1
Vertical Speed -> Vertical Speed	2	Vertical Speed -> VNAV Path Descent	1
VNAV -> Altitude Capture	2	VNAV -> FLCH	1
VNAV -> Altitude Hold	2	VNAV -> Localizer Track	1

Table C.6:   Most Frequently Identified Transitions

Transition	Number	Transition	Number
Altitude Hold -> Holding Fix	1	VNAV -> VNAV	1
Altitude Hold -> VNAV	1	VNAV Climb -> Altitude Hold	1
Altitude Hold -> VNAV Descent	1	VNAV Climb -> VNAV Path	1
Altitude Hold -> VNAV Path Descent	1	VNAV Descent -> Vertical Speed	1
ANY -> Altitude Capture	1	VNAV Descent -> VNAV Path	1
ANY -> VNAV Path Descent	1	VNAV Path -> Altitude Capture	1
Approach -> Approach	1	VNAV Path -> Glideslope Capture	1
Approach -> Glideslope Capture	1	VNAV Path -> Vertical Speed	1
Autopilot OFF -> Autopilot ON	1	VNAV Path -> VNAV Descent	1
CWS Climb -> Altitude Capture	1	VNAV Speed Descent -> VNAV Path Descent	1

C.2.1   Transition Analysis

The fully detailed analyses on each pilot identified transitions is shown in Table C.7, Table C.8 and Table C.9. Table C.10 and Table C.11 show the "typical" transitions culled from the B737-300 and B757 manuals for comparison.

Table C.7: B737-3/4/5/6/7/800 Analyzed Transitions

Transition	Report	Cyclomatic complexity	Conditions	Branches	Terminal States	FCOM reference	Description
VNAV Path to VNAV Speed (via ALT INV)	3	6	3	2	1	4.10.9	1. Aircraft departs planned descent profile 2. All FMC Alt constraints removed 3. HDG SEL engaged
FLCH to VNAV Path (speed reverts to FMS)	9	4	2	0	2	4.10.5	1. Press FLCH button, new speed is entered 2. Change speed target manually
Heading Select to Approach	9	4	2	0	2	lookup in -400	1. Correct capture of LOC when in range 2. Incorrect capture based on >90deg from approach vector
VNAV Path to V/S	11	6	3	0	3	????	1. Press V/S button 2. Change Alt in MCP 3. Select "Capture" on FMC Descent Page (suspect subject is confusing V/S and VNAV Path)
V/S to FLCH (overspeed)	13	4	2	0	2	4.20.21	1. Press FLCH button 2. Overspeed envelope protection
ALT ACQ to V/S	21	4	2	0	2	4.10.7	1. Press V/S button 2. While in ALT CAP, move Alt knob > 100ft
VNAV Path to VNAV Speed (via hdg sel)	21	6	3	2	1	4.10.9	1. Aircraft departs planned descent profile 2. All FMC Alt constraints removed 3. HDG SEL engaged
ALT ACQ to V/S	85	4	3	0	1	4.10.7	1. Press V/S button 2. While in ALT CAP, move Alt knob > 100ft
Go Around to V/S Speed reset to instantaneous value during ALT ACQ	97	6	3	2	1	4.20.17	This is odd. Whenever we transition into ALT ACQ, the speed is captured at the instantaneous value, which can be problematic during a Go Around. 1. ALT ACQ engaged 2. Change speed target manually
V/S to FLCH (overspeed)	97	4	2	0	2	4.20.21	1. Press FLCH button 2. Overspeed envelope protection
VNAV to ALT HOLD / Path	25					4.20.5	An inconsistency: when in VNAV SPD climb, intersecting MCP reverts to ALT HOLD, but in Descent, goes to VNAV Path? (not in manual)

Table C.8: B757/767 Analyzed Transitions

Transition	Report	Cyclomatic complexity	Conditions	Branches	Terminal States	FCOM reference	Description
VNAV to ALT HOLD	43	7	4	0	3	07.10.05	1. Press ALT HOLD switch 2. Intercept MCP target altitude 3. Pass TOD with MCP alt > cruise alt
Speed in Mach to Speed in IAS	43	6	3	2	1	07.10.02	1. Descent through 300 KIAS 2. Push IAS/Mach switch (if in range) 3. Switch to FLCH from VNAV
TO to ALT HOLD (at low alt): ALT CAP resets Speed Target to current value	49	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
VNAV Path to VNAV Speed	51	6	3	1	2	NONE	1. Manual Speed intervention 2. Exceed commanded speed 3. Change crossing restrictions to unachievable value
TO to ALT HOLD (at low alt): ALT CAP resets Speed Target to current value	51	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
A/P off to A/P on (A/T armed to active)	53	10	5	4	1		Engaged when AT modes selected: EPR/SPD/ VNAV/FLCH/GA
VNAV Path to G/S capture	61	5	4	0	1	07.10.05	Capture requires FMC Alt below capture Alt, MCP Alt below capture alt, APP armed, LOC captured
VNAV PATH to ALT CAP: ALT CAP Speed Target resets Speed Target	71	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
Speed in Mach to Speed in IAS	71	6	3	2	1	07.10.02	1. Descent through 300 KIAS 2. Push IAS/Mach switch (if in range) 3. Switch to FLCH from VNAV
CLMB/DES to ALT CAP: ALT CAP Speed Target resets Speed Target	79	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged

Table C.8:   B757/767 Analyzed Transitions

Transition	Report	Cyclomatic complexity	Conditions	Branches	Terminal States	FCOM reference	Description
TO to ALT HOLD (at low alt): ALT CAP resets Speed Target to current value	79	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
VNAV to ALT HOLD	99	7	4	0	3	07.10.05	1. Press ALT HOLD switch 2. Intercept MCP target altitude 3. Pass TOD with MCP alt > cruise alt
G/A to APP	101	5	4	0	1	None	Disconnect A/P, cycle both FD, reconnect A/P, rearm APP
VNAV PATH to ALT CAP: ALT CAP Speed Target resets Speed Target	71	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
Speed in Mach to Speed in IAS	71	6	3	2	1	07.10.02	1. Descent through 300 KIAS 2. Push IAS/Mach switch (if in range) 3. Switch to FLCH from VNAV
CLMB/DES to ALT CAP: ALT CAP Speed Target resets Speed Target	79	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
TO to ALT HOLD (at low alt): ALT CAP resets Speed Target to current value	79	6	3	1	2	07.10.02	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
VNAV to ALT HOLD	99	7	4	0	3	07.10.05	1. Press ALT HOLD switch 2. Intercept MCP target altitude 3. Pass TOD with MCP alt > cruise alt

Table C.9: A320 Analyzed Transitions

Transition	Report	Cyclomatic complexity	Conditions	Branches	Terminal States	FCOM reference	Description
LNAV to Heading Hold	39	6	3	1	2	None	End of stored flight plan, route discontinuity leads to heading hold, CC from B757
ALT* to V/S	45	4	2	0	2	14-6	1. Press V/S button 2. Move Alt knob
V/S to Open Climb	45	5	3	0	2	14-6	1. Press LVL CHG 2. Underspeed condition and FCU alt is above aircraft
ALT* to V/S	77	4	2	0	2	14-6	1. Press V/S button 2. Move Alt knob
ALT* to V/S	87	4	2	0	2	14-6	1. Press V/S button 2. Move Alt knob
LOC CAP to Heading Select	87	14	7	5	2	14-8	RA>400ft ^ (1. Pressing LOC 2. Pressing APPR 3. HDG selected 4. Disengage A/P 5. G/A engaged) 6. Moving Heading knob
ALT* to V/S	89	4	2	0	2	14-6	1. Press V/S button 2. Move Alt knob

Table C.10: B737-300 Typical Transitions

Transition From	To	Cyclomatic complexity	Conditions	Branches	Terminal States	Description
LNAV	LNAV Heading Hold	10	5	4	1	discontinuity, end of later offset, last waypt, waypt b/w runway and extended ctrline, intercept leg/course to proc.
VNAV off	VNAV on	6	3	2	1	>400ft RA, MCP alt > current, press button
VNAV on	VNAV off	2	1	0	1	press button
SPD/VNAV	Idle/VNAV	2	1	0	1	Cross t/d
Idle/VNAV	Throttle Hold/ VNAV	2	1	0	1	headwind
Manual Throttle	Speed	3	2	0	1	Speed Switch AND A/T armed
Manual Throttle	EPR	3	2	0	1	EPR Switch AND A/T armed
FLCH	Alt Hold	5	2	1	2	1. Intercept MCP altitude 2. Press ALT HOLD
LNAV	Heading Select	2	1	0	1	Switch
LNAV	Heading Hold	2	1	0	1	Heading Hold Switch
V/S	Alt Hold	5	2	1	2	1. Intercept MCP altitude 2. Press ALT HOLD
LNAV	Localizer	3	2	0	1	Capture and LOC armed
VNAV	V/S	2	1	0	1	Push switch
VNAV	Alt Hold	9	4	2	3	1. Press ALT HOLD switch 2. Intercept MCP target altitude 3. Pass TOD with MCP alt > cruise alt
VNAV	Speed	2	1	0	1	press button
VNAV	EPR	2	1	0	1	press button
VNAV	G/S	3	2	0	1	Capture and G/S Cap armed
Speed	Speed	6	3	1	2	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed
Go Around Off	Go Around ARM	4	2	1	1	G/S Cap or flap extension

Table C.11:   B757 Typical Transitions

Transition From	To	Cyclomatic complexity	Conditions	Branches	Terminal States	Description
Alt Hold	V/S	4	2	1	1	Press V/S, Rotate knob while at MCP alt
Speed	N1	5	3	1	1	Press N1 while FLCH or VNAV engaged
N1	Speed	5	3	1	1	Press Speed while FLCH or VNAV engaged
VNAV	Alt Hold	9	4	2	3	1. Press ALT HOLD switch 2. Intercept MCP target altitude 3. Pass TOD with MCP alt > cruise alt
VNAV	V/S	6	3	2	1	Press Switch, or extend flaps, or Localizer off
VNAV	G/S	3	2	0	1	Capture Glide Slope and G/S Armed
V/S	FLCH	4	2	1	1	Switch, or Overspeed
LNAV	Hdg Sel	2	1	0	1	switch
LNAV	Heading Select	2	1	0	1	Switch
LNAV	Localizer	3	2	0	1	Capture and LOC armed
G/A Armed	G/A	2	1	0	1	TOGA switch
G/A eng	Off	2	1	0	1	Touchdown (squat switch)
G/A Eng	Alt Acq	2	1	0	1	Stabilize Trim ok for single A/P
Speed	Speed	7	3	2	2	1. Speed reset manually 2. Press FLCH while in FLCH, reset to current speed 3. Speed reset when ALT CAP engaged
Alt Hold	V/S	4	2	1	1	Press V/S, Rotate knob while at MCP alt
Speed	N1	5	3	1	1	Press N1 while FLCH or VNAV engaged
N1	Speed	5	3	1	1	Press Speed while FLCH or VNAV engaged
VNAV	Alt Hold	9	4	2	3	1. Press ALT HOLD switch 2. Intercept MCP target altitude 3. Pass TOD with MCP alt > cruise alt
VNAV	V/S	6	3	2	1	Press Switch, or extend flaps, or Localizer off

Appendix D

Web-based Pilot Automation Complexity Questionnaire

Massachusetts Institute of Technology

MIT ICAT Survey on Complexity in Autoflight Systems

International Center for Air Transportation
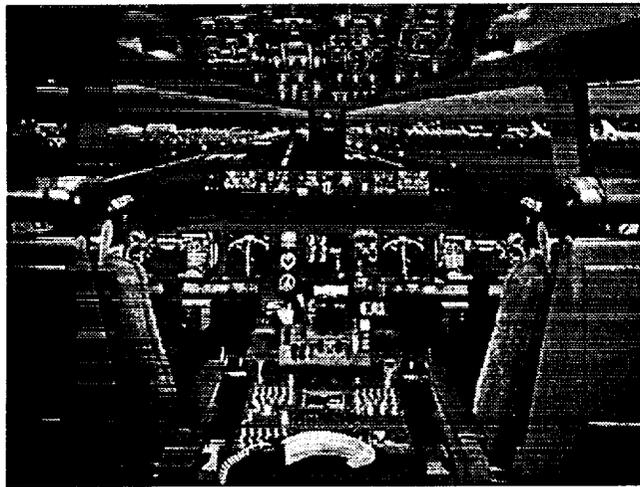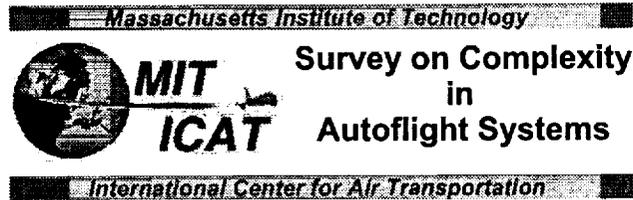
The International Center for Air Transportation (ICAT) at MIT is involved in an effort to understand complexity in commercial aircraft. We are trying to understand those factors which make autoflight systems complex. The input of commerical pilots with advanced autoflight system experience is very important in this process. Your participation in this survey will make a valuable contribution, since we need the input of current glass cockpit pilots to identify the areas of current designs which can be improved upon in future aircraft. It should take approximately 20-30 minutes to complete this survey. Speed is not, however, a goal for the experiment.

Information concerning your aviation background will help us to more accurately assess some of the variables that affect the performance of the pilots. *All information that you provide will remain completely anonymous.*

If you are curious about the researcher behind this work, I am a doctoral student in Aeronautics & Astronautics at MIT, and a student pilot, particularly interested in aviation & aerospace human-interface and safety issues. Results from this experiment will contribute to the work of my thesis, which is supervised by Professor R. John Hansman and supported by NASA Langley Research Center. Please feel free to contact me if you have questions. Thank you, and enjoy the experiment.

Sanjay Vakil
Graduate Research Assistant

International Centre for Air Transportation
MIT Room 35-217,
Cambridge, MA 02139 USA
phone: (617) 253-0993 fax: (617) 253-4196
sanj@mit.edu

Figure D.1:   Web-based Survey, Page 1

Background on Mode Transitions

Based on a review of the Aviation Safety Reporting System (ASRS), we found situations where the autoflight system caught the pilot unaware or had some sort of unexpected behaviour. Many of these situations involved mode transitions. A mode transition occurs any time that the aircraft switches from one mode to another, such as between Vertical Speed mode and Altitude Hold mode.

The next sections will describe the characteristics of mode transitions which are hypothesized to contribute to complexity. These consist of multiple transition types, multiple paths along which a transition can occur, and conditions which must be met before a transition will occur. Each of these is detailed in the next section.

The images on the next several pages are quite large, in order to let you see the details of the primary flight display. Please expand your web browser to its largest dimensions in order to view the images.

[Continue]

Figure D.2:   Web-based Survey, Page 2

Types of Transitions

- **Manual transition:** caused by a pilot pressing a switch.
  Example: Boeing B777 in a Vertical Speed descent. When the HOLD button is pressed, the aircraft immediately switches to the Altitude Hold mode and holds the current altitude.

Vertical Speed Descent   Press HOLD button   Transition to ALT HOLD at current altitude

- **Automatic transition:** occurs when the aircraft switches modes without direct pilot intervention
  Example: Transition to Altitude Hold when an aircraft intercepts the altitude shown in the altitude window. During a Vertical Speed manuever in a B737, the aircraft will transition to Altitude Hold mode when this interception occurs

Vertical Speed Descent   Approach Altitude in Altitude Window   Transition to ALT HOLD at Altitude Window Altitude

- **Armed transition:** occurs when the autoflight system has been authorized or armed to make a transition.
  Example of this is the transition from a Glide Slope Armed mode to a Glide Slope Tracking mode. The autoflight system will not switch directly into the tracking mode unless it was previously armed by the pilot.

Glide Slope is Armed for Capture   Glide Slope Capture Occurs   Glide Slope Mode becomes Active

Figure D.3:   Web-based Survey, Page 3

215

Multiple Paths

Some transitions have multiple ways to occurs: different paths. The transition between Vertical Speed and Altitude Hold can occur along two different paths. Either the HOLD button can be pressed by the pilot, resulting in an immediate leveloff, or the aircraft can approach and leveloff at the altitude in the altitude window. Each path also has different final states:

- The automatic transition captures the value in the altitude window.
- Pressing the HOLD button results in the aircraft leveling off at a different altitude than the one shown in the altitude window
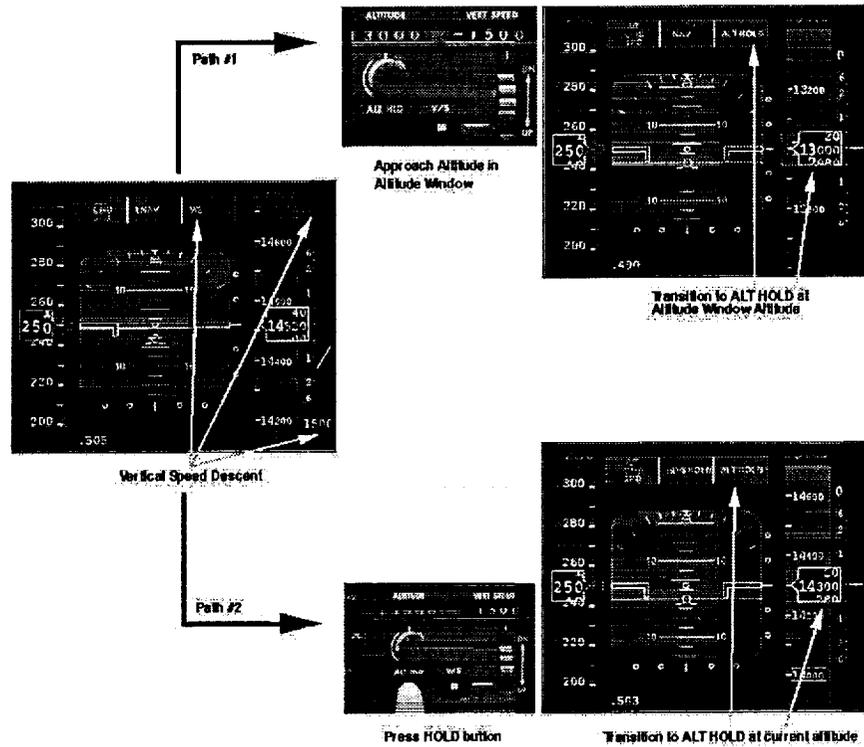
Figure D.4:   Web-based Survey, Page 4

Conditions

For a transition to occur, a set of conditions must be met associated with each path. These conditions are shown in the figure along each path.

- Single Condition Example: Pressing the Heading Select button.

Start in LNAV Mode   Press HDG SEL Button   Transition to HDG SEL Mode
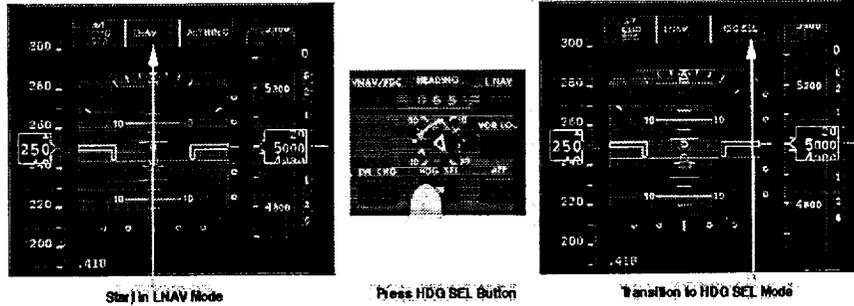
- Multiple Condition Example: Transition from Vertical Speed to the VNAV mode on the B757. Requires the VNAV button to be pressed, the aircraft to be above 400 ft RA, and the altitude in the altitude window to be above the current altitude.

Start in ALT HOLD Mode   Press VNAV Button while • Above 400 ft RA • Altitude Window is Above Aircraft   Transition to VNAV SPD Mode
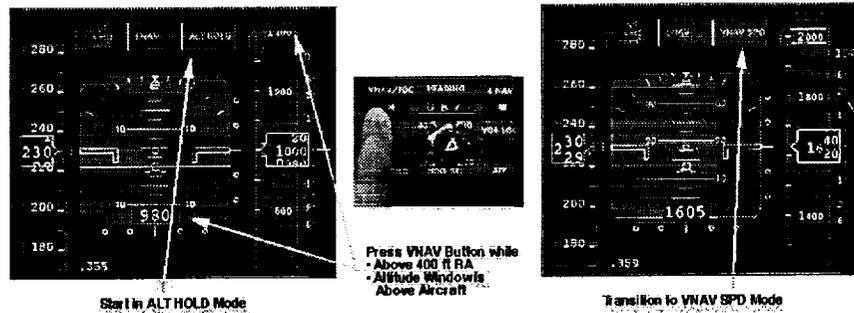
The difference between manual and automatic transitions are the criteria which must be met before the switch occurs. In manual transitions, there must be at least one manual condition, such as a button press. In automatic transitions, all of the conditions are satisfied without manual intervention.

Figure D.5:   Web-based Survey, Page 5

Background Information

Age: [_____]   Sex: Male ○   Female ○

Total Flight Time: [_____]

What aircraft are you currently flying, or have you recently flown?

	Aircraft Type	Flight Hours (Approximate)	Position
Current	[_____]	[_____]	[_____]
Previous	[_____]	[_____]	[_____]

Estimated Flight Hours in 1998: [_____]
Estimated Flight Hours in 1999: [_____]

[Continue]

Figure D.6:   Web-based Survey, Page 6

Most Difficult Mode Transitions

In the aircraft you currently fly, please identify what you consider to be the 3 most difficult mode transitions. Try to answer these questions as if you were instructing another pilot, new to this aircraft, about its difficult modes. Please be sure to list all conditions which must be fulfilled for the transition to occur. Please, include what the aircraft does after the transition: what new heading/speed/descent rate/altitude become active.

Also, there will be buttons at the bottom of each page to allow you to review the information on Transition Types, Multiple Paths, and Conditions at any point. Each button will jump you to the information page an then allow you to continue the survey where you left off.

[Continue]

Figure D.7:   Web-based Survey, Page 7

First Mode Transition

Transition from | Mode A | to: | Mode B |

Type of transition (check all that apply):

Manual ☐   Automatic ☑   Armed ☐

Estimate the number of different paths possible in this transition: [Select estimated number of Paths ▲]

Describe the transition In as much detail as possible.
Think about explaining this to another crew member.
If possible, include the necessary conditions, possible paths, and outcomes for the transition:

```
Enter transition description here.
```

What makes this transition difficult?

```
Enter what makes the transition difficult here.
```

[Review Transition Types]   [Review Multiple Paths]   [Review Conditions]   [Continue]

Figure D.8:   Web-based Survey, Page 8-10

Comparing Mode Transistions

We are interested in how complicated you feel difficult transitions are. In this final section, you will be asked to compare the mode transitions that you listed previously against each other. Please rate the following 10 comparisons between pairs of mode transitions.

Mode A to Mode B is	Much Less Complicated ○	Less Complicated ○	Equally Complicated ○	More Complicated ○	Much More Complicated ○	than Mode C to Mode D

Mode A to Mode B is	Much Less Complicated ○	Less Complicated ○	Equally Complicated ○	More Complicated ○	Much More Complicated ○	than Mode E to Mode F

LVL CHG or FLCH to ALT HOLD is	Much Less Complicated ○	Less Complicated ○	Equally Complicated ○	More Complicated ○	Much More Complicated ○	than Mode A to Mode B

LNAV to Heading Select is	Much Less Complicated ○	Less Complicated ○	Equally Complicated ○	More Complicated ○	Much More Complicated ○	than Mode A to Mode B

Mode C to Mode D is	Much Less Complicated ○	Less Complicated ○	Equally Complicated ○	More Complicated ○	Much More Complicated ○	than Mode E to Mode F

Mode C to Mode D is	Much Less Complicated	Less Complicated	Equally Complicated	More Complicated	Much More Complicated	than LVL CHG or FLCH to ALT HOLD

LNAV to Heading Select is	Much Less Complicated	Less Complicated	Equally Complicated	More Complicated	Much More Complicated	than Mode C to Mode D

Mode E to Mode F is	Much Less Complicated	Less Complicated	Equally Complicated	More Complicated	Much More Complicated	than LNAV to Heading Select

LVL CHG or FLCH to ALT HOLD is	Much Less Complicated	Less Complicated	Equally Complicated	More Complicated	Much More Complicated	than Mode E to Mode F

LNAV to Heading Select is	Much Less Complicated	Less Complicated	Equally Complicated	More Complicated	Much More Complicated	than LVL CHG or FLCH to ALT HOLD

Figure D.9: Web-based Survey, Page 11

220

Thank you for participating in this survey. If you enjoyed taking part, and are interested in taking part in other aviation experiments, please send mail to sanj@mit.edu so that you can be contacted in the future.

Similarly, if you are interested in the results, send mail and an electronic copy of the report will be sent out when it has been completed.

Figure D.10:   Web-based Survey, Page 12

Appendix E

Electronic Vertical Situation Display

A concern in modern automation is that sufficient information may not be available to the pilot to accurately track the state of the automation. Using the terminology of control engineers, the system is not "observable." Accurately characterizing necessary automation feedback can support modifications that make the system more tractable. However, since such changes to aircraft are much more rare, and expensive, than changes to training or procedures, they are likely to only be done in extreme situations. This chapter will discuss the results of a study performed to gauge the impact of a new display, designed to show the vertical situation of the aircraft, in a set of automation-incident related situations.

E.1   Motivation for Vertical Domain Research

The fundamental motivation for this research is to reduce the number of aircraft incidents related to mode awareness problems. Toward this end, incidents in which mode awareness issues are suspected to have been a contributing factor have been examined. The review of the reports contained within the Aviation Safety Reporting System (ASRS) database along with a set of focused interviews with flight crews and an examination of the feedback mechanisms in modern AutoFlight Systems were used to ascertain the areas of interest. The background research has identified the lack of feedback and a high degree of automation complexity in the vertical channel as a possible source of mode awareness problems. Based on this information it was hypothesized that adding feedback to the vertical channel would mitigate some of the reported and suspected mode awareness problems.

E.2   Electronic Vertical Situation Display

Background research highlighted the vertical channel as an area in need of improved feedback. Based on this hypothesis, a set of crew information requirements was developed. This set of requirements was incorporated into an Electronic Vertical Situation Display (EVSD) to mitigate some of the identified problems. The EVSD is envisioned to provide an analog to the Electronic Horizontal Situation Indicator, which is currently available in glass cockpits. The

display is designed to show the programmed vertical path of the aircraft and the modes associated with that path.

The information requirements for the EVSD were created based on operator requirements and on known mode awareness problems. The information requirements have four major components: the current mode, any anticipated modes, transitions into anticipated modes, and the consequences of the current state of aircraft automation. The four requirements also provide answers to the most commonly asked questions asked in glass cockpits: "What is it doing?," "Why did it do that?," and "What will it do next?" (Wiener, 1989).

E.3 EVSD Format

Figure E.1 shows the prototype EVSD. The display has four distinct areas. At the top of the display is the mode display window, showing the current and anticipated modes, control allocations and target states. At the left is a scalable altitude tape. The bottom window can either display the path distance (if in LNAV mode), or the range directly ahead of the aircraft. Finally, the main window shows the aircraft vertically in relation to the upcoming waypoints and mode transition points.
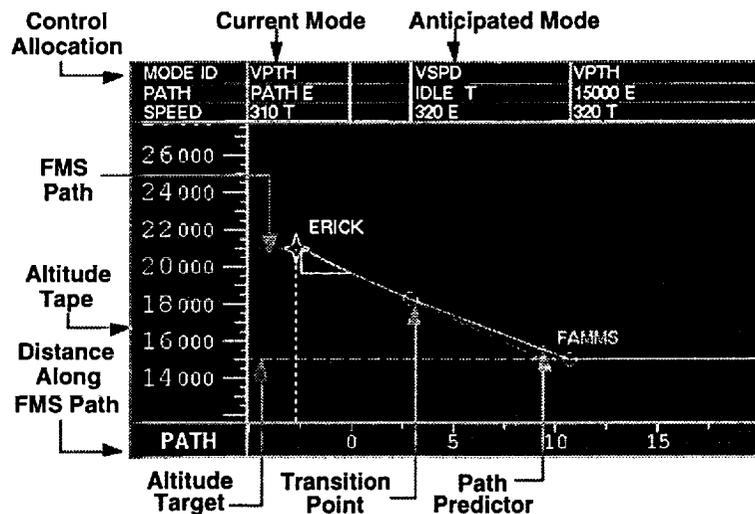
Figure E.1:   EVSD During Descent, Reaching High Speed Limit

224

E.3.1 Current Mode

The current mode of the automation needs to be identified along with any of the specific attributes of the mode such as target state values and control allocation. In Figure E.1, the current mode is identified in the top window in green text, directly above the aircraft symbol. Underneath are the control allocations and the target states for the active mode. In this example, the aircraft is in VNAV Path Descent (VPATH) with the vertical path controlled by elevator (E) and the speed controlled by throttle (T). An example of a transition criterion is the dashed magenta line at 15 000 ft, which is the altitude dialed into the Mode Control Panel.

E.3.2 Anticipated Modes

Anticipated modes consist of the future modes into which the automation expects the aircraft to transition. Based on some extrapolation of the current aircraft and automation state, the system needs to be able to anticipate future mode transitions and the associated future modes. This determination may be straightforward for many preprogrammed macro modes, but it may be inaccurate for uncommanded mode changes. For example, the highly accurate prediction of entering an envelope protection mode downstream is difficult.

On the EVSD, the anticipated mode is shown in the top window above the point where it is predicted to be engaged. The anticipated target state and control allocations are depicted in a manner similar to the current mode. In Figure E.1, the system is predicting a speed violation and a mode transition to the VNAV Speed Mode (VSPD). In this mode, the display shows that the vertical path will be controlled by the throttles to Idle, and the speed will be controlled by elevators to 320 kts. Note that both the target states and the control allocation change when the new mode is engaged.

Another mode change is anticipated once the aircraft reaches the MCP altitude of 14 000 ft, approximately 12 nmi ahead of the aircraft. The aircraft will switch to VNAV Path Mode (VPATH). Once the automation switches to this mode, the altitude becomes a target state, as shown in the box underneath the VPATH text.

225

E.3.3 Mode Transitions and Consequences

Anticipation of the consequences of the current state of the automation is necessary. A predictive profile based on the current automation state should display the locations of anticipated mode transitions and their consequences on the flight path. Determining the location of these transitions requires making assumptions about both the intentions of the pilots and factors external to the aircraft, such as weather and air temperature. Closely related to this is the ability to anticipate mode changes. Since there may not be warning of certain types of mode changes, especially those which are selected by the crew, this may not be possible in all situations.

Mode transition alerting has several elements. The first is the text of the anticipated mode translating across the top window, arriving at the current mode slot when engaged. The second is an Aircraft Path Line, commonly referred to as the "green line" which shows the path that the aircraft will travel based on the current state of the automation. In Figure E.1, the green line is shown deviating from the solid magenta line as the automation anticipates the aircraft being unable to maintain the commanded descent rate. In this case, an elbow in the path also highlights where the transition is calculated to occur. The consequences of a mode change are based on an extrapolation of the current state of the aircraft automation. A predictive profile based on the current automation state is shown to display the locations of anticipated mode transitions and their consequences on the flight path.

The green line also has a set of circles which highlight where the aircraft is going to undergo a mode transition. Green circles depict when the aircraft is going to undergo a programmed mode transition, such as leveling at a commanded altitude as seen in Figure E.1 near the 12 nmi point. These circles align with the green bars in front of programmed modes shown in the upper window. Yellow circles depict uncommanded or automatic mode changes that occur due to speed violations. In Figure E.1 the yellow circle at the bend of the elbow highlights the transition at the 3 nmi point to the VNAV Speed Mode (VSPD). These circles also align with the yellow bars in front of automatic modes in the upper window.

E.3.4 Additional Symbology

A solid magenta line connecting waypoint crossing restrictions is a graphical display of the current vertical path programmed into the FMS. Shown in the main section of the EVSD screen are the waypoints programmed into the FMS. In a manner consistent with the EHSI, programmed waypoints are magenta, and the active waypoint (the one the aircraft is flying to) is white. Non-programmed waypoints are not shown.

Altitude crossings are shown by the altitude of the waypoint symbol. Waypoints without restrictions are placed on the ground, or on the bottom edge of the screen if the ground is not visible. Dashed lines descend to the path scale on the bottom of the display to allow the distance to future waypoints to be easily determined. The horizontal scale is defined by the path distance between each waypoint.
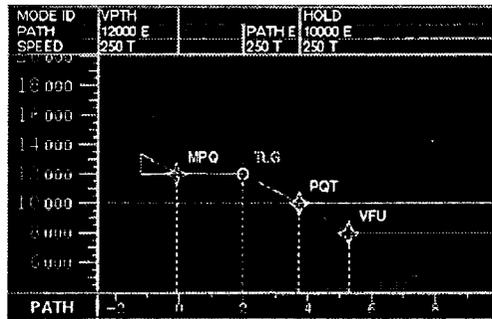
Figure E.2:   EVSD in VNAV Descent

The example shown in the Figure E.2 is a Vertical Navigation macro mode (VNAV) descent. The aircraft is shown on the first segment of a descent profile using the VNAV mode to maintain an altitude of 12 000 ft. Near the TLG waypoint, the aircraft is anticipated to remain in the VNAV macro mode, but to change the target state at the Top of Descent point (a transition criterion) to a path-based descent. The Aircraft Path Line highlights another mode change that will occur near PQT at 10 000 ft as the aircraft intersects the preprogrammed MCP altitude transition criterion and switches out of VNAV and into an Altitude Hold mode.

Figure E.3 shows an aircraft close to approach with the additional symbology required near landing. The dashed white line is an indication of the location of the ILS glide slope. At the end of the glide slope is the runway, shown at the correct altitude and length. The difference between

227
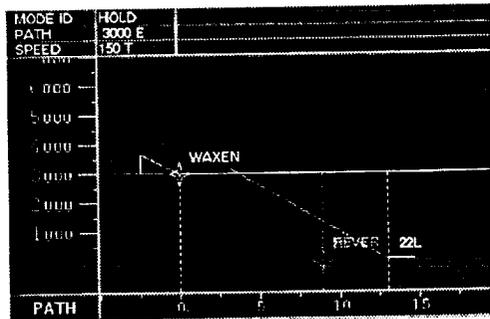
Figure E.3:   EVSD Near Approach

waypoints with and without an altitude restriction is also apparent. WAXEN has an altitude restriction associated with it, and is shown at that altitude of 3000 ft. Since there is no altitude restriction associated with the REVER waypoint, it is placed at ground level. Finally, the very bottom of the main window in the Figure E.2 has terrain information. While this capability was not used in the experimental evaluation, it has been prototyped on the EVSD.

E.4   EVSD Evaluation

The approach to EVSD evaluation was to have subject pilots act as Pilot Not Flying (PNF) and observe a set of scenarios running on a part task simulator. Subjects were drawn from a pool of current "glass cockpit" airline pilots. While subjects observed the scenarios, the researcher acted as Pilot Flying (PF), interacting with the simulated aircraft automation and responding to prerecorded Air Traffic Control directives. Both the subject and the researcher were located in from of identical displays to simulate the set of instruments available to the captain (subject) and first officer (researcher) in an actual cockpit. Both displays were electronically slaved together to show identical information at all phases of the experiment. If during the course of the scenario the subject felt that some sort of *mode event* occurred, they pressed a button on the side stick controller and articulated their concern. This audio data was later analyzed for timing and content. A mode event was described to the subjects as: an uncommanded mode transition occurring, an error had been made in interfacing with the FMS, or an unsafe or nonprocedural operation had taken place.

This evaluation methodology is very similar to the SAGAT situational awareness technique (Endsley, 1995) with some implementation differences. In contrast with SAGAT evaluations,

where the subjects actively interact with the simulator, the EVSD evaluation scenarios were completely passive for the subjects. The only interaction was for the subjects to articulate concerns at any point during the evaluation that they felt necessary. During experimental design it was found that simply monitoring the displays was an unrealistic representation of the workload in actual cockpit operations. To correct this, subjects were given an unrelated side task to monitor.

E.4.1 Measures

Both objective and subjective measures were used during display evaluation. The objective measures were based on the elapsed time between the mode event and the pilot response or, when relevant, the net altitude deviation the aircraft from the intended path. The latter indication was used when the altitude deviation of the aircraft was the relevant factor in the incident. For subjective measures, after each scenario subjects were asked questions which addressed the cause of the mode transition, the consequences of the new mode state and the cues used by the subject to determine that an incorrect automation state was reached.

These questionnaires were rated to determine the level of the subject's understanding of the problem. Each rating corresponded with differing levels of comprehension. In order to determine in a objective manner whether pilots understood the underlying cause of the incident, each scenario had a key element that was deemed to be indicative of understanding. Finally, after the experiment was completed, subjects filled out a survey comparing the EVSD enhanced AutoFlight system to the regular AutoFlight System.

E.4.2 Scenarios and Objective Results

A total of eight subjects participated in this experimental evaluation. The subject ranged in ages from 39 to 53 with a mean age of 47. First officers comprised 62.5% of the subjects; the remaining 37.5% were captains. 37.5% were initially trained as civil pilots, while the remainder were initially trained by the military. Subjects had between 4000 and 15870 hours of civilian flight experience with a mean of 8646 hours. Between 200 and 5000 of these hours were in glass cockpits, with a mean of 2263 hours. Estimated flight hours in 1995 ranged between 250 and 800 with a mean of 533 hours.

The bulk of the scenarios have results based on objective timing information. Two of the scenarios (Target Error by Pilot Flying during Non-precision Approach and the Altitude Capture Failure during Altitude Change) use the net altitude deviation as a more relevant measure, as it is of greater merit. In this context, negative altitude deviations refer to pilots reporting the incident before they reached the point at which an altitude deviation would occur. Similarly, negative times refer to pilots anticipating the incident and reporting it before it would occur, with the absolute value of the time indicating the anticipation time.

Glide Slope Transmitter Failure during Approach

This scenario simulated the loss of the ground based glide slope transmitter during an autoflight system flown Instrument Landing System (ILS) approach. A few seconds after the glide slope is captured, the ground based transmitter fails. This failure results in a loss of signal to the aircraft system, causing a mode reversion to Vertical Speed mode with the current (instantaneous) vertical speed as the target criterion. The vertical speed is such that the aircraft will land short of the runway. Figure E.4 shows that by using the EVSD, subjects noticed and reacted to this mode event 17.2 seconds sooner than without at a confidence level of 80%.
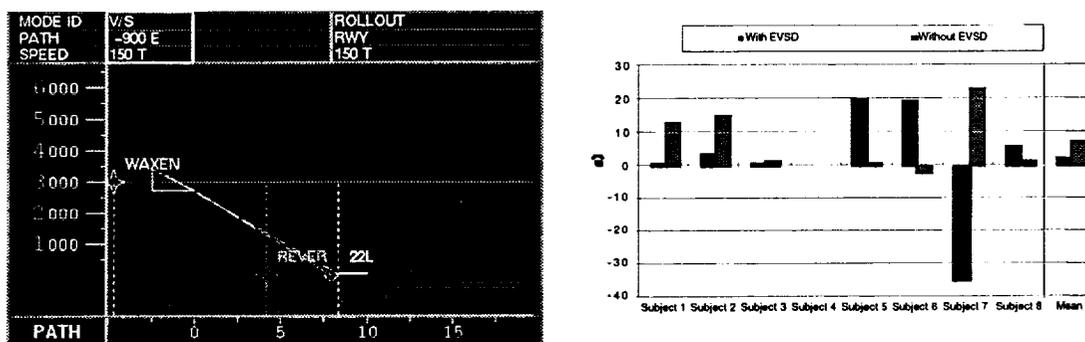
Figure E.4:   Glide Slope Transmitter Failure during Approach

Figure E.4 shows that by using the EVSD, subjects noticed and reacted to this mode event 17.2 seconds sooner than without at a confidence level of 80%. This is interesting in that the only changes on the EVSD during this scenario are the shift in the green line and the flashing and alteration of the text displayed in the mode header

230

Overspeed Envelope Protection during Altitude Change

A late ATC directive leads to a high vertical descent rate, which engages the envelope protection of the aircraft. After initially trying a Flight Level Change mode descent, it becomes clear that a high speed Vertical Speed mode descent will be necessary. During this steep descent in vertical speed mode, the envelope protection limits of the aircraft are surpassed. The aircraft transitions to High Speed Protect mode and AFS control allocation shifts from speed-on-thrust to speed-on-pitch to prevent the aircraft from overspeeding further. Finally, due to insufficient descent rate, the aircraft overflies the altitude crossing.
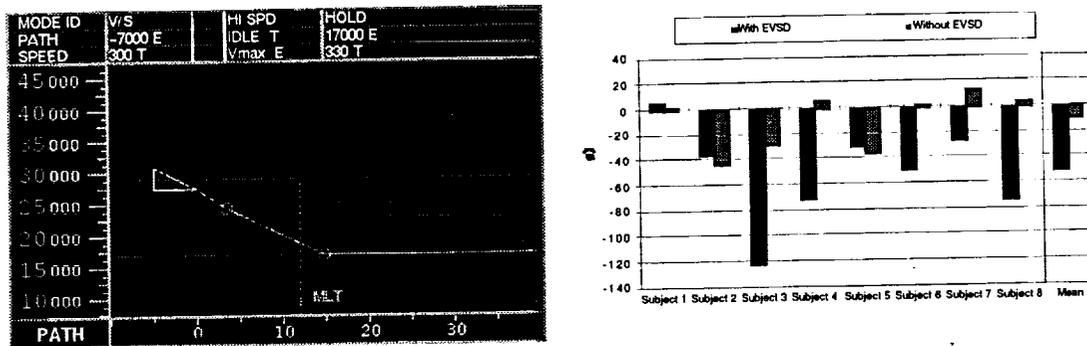
Figure E.5: Overspeed Envelope Protection during Altitude Change

Figure E.5 shows that the anticipation capabilities of the EVSD allowed pilots to predict the overspeed situation 40.3 seconds sooner than they did without the display, at a confidence level of 95%. Negative times refer to the pilots anticipating the incident. It was clear during this experiment that the anticipation capability was directly responsible for the improvement as subject directly referenced the EVSD when calling ATC to report that they would be unable to make the crossing. Several subjects commented on the fact that the information was easily available in the graphical format. In addition, several subjects mentioned additional quantitative information available on the display, such as the distance to the overspeed condition, the altitude at which the overspeed would occur, and the distance by which the altitude restriction would not be met.

Altitude Capture Failure during Altitude Change

During a Flight Level Change Descent, the aircraft fails to level-off at the MCP altitude. Instead of capturing the programmed altitude, the aircraft, the aircraft continues its descent at the commanded vertical speed target as seen in the figure. Since the MCP altitude is now above the aircraft, the vehicle enters an open descent.
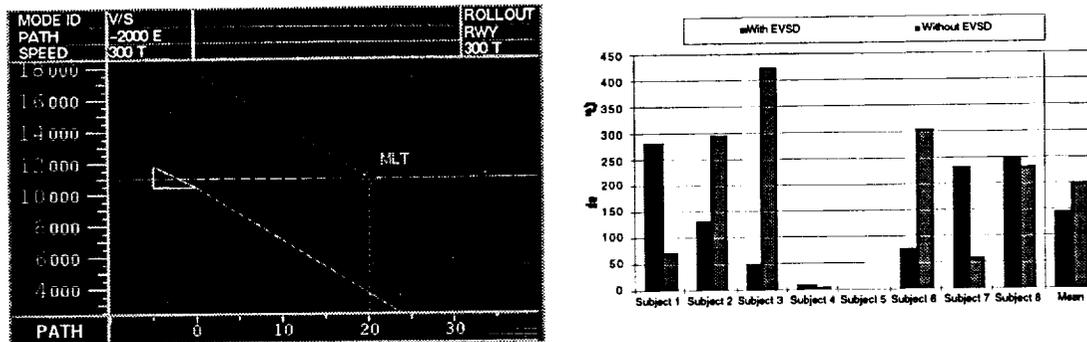
Figure E.6:   Altitude Capture Failure during Altitude Change

As seen in Figure E.6, subjects were observed to deviate from the target altitude by 45.8 feet less with the EVSD than without, but this value had little statistical significance. In this scenario, the EVSD did not appear to help or hinder this type of inner loop monitoring in a significant manner. It was observed that subjects tended to watch the altitude tape on the Primary Flight Display rather than the EVSD. This seems to imply that the EVSD may not be useful in situations where the information necessary to detect a mode event is explicitly available on a standard piece of cockpit automation. Alternately, the fact that pilot are trained to use specific display when dealing with high importance tasks which have dedicated feedback mechanisms may have overridden any use of the EVSD. Due to an anomaly in the simulation, one of the subjects was not tested in this particular scenario.

VNAV Path to VNAV Speed Transition due to High Winds

During a VNAV flight profile, unanticipated wind conditions cause the aircraft to switch to an envelope protection VNAV SPD mode in order to prevent the aircraft from varying too far from the speed target. The automation is designed to switch to VNAV Speed mode that when the speed in VNAV Path mode reaches the critical value of commanded speed ± 10 kts. In VNAV Speed

mode, the vertical path is sacrificed to maintain the speed target, causing the aircraft to overfly the next altitude restriction.
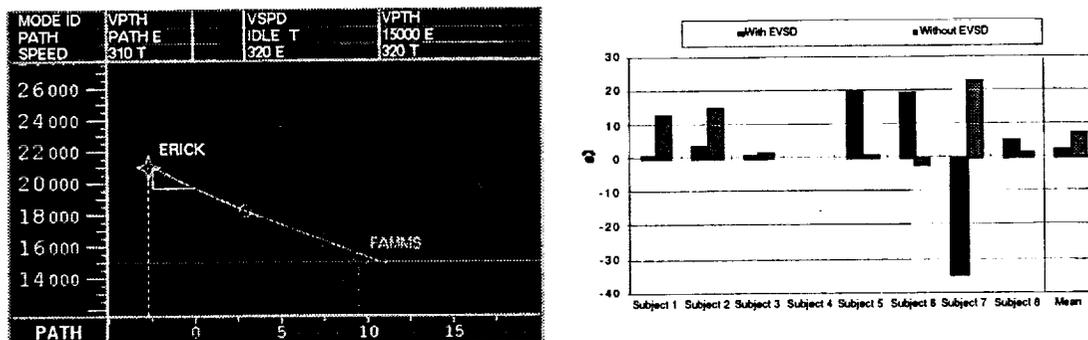
Figure E.7:   VNAV Path to VNAV Speed Transition due to High Winds

Figure E.7 shows that subjects reacted to this mode change 4.7 seconds faster with the EVSD than without. However, this value is not statistically significant with the small number of subjects tested. This result was unexpected since the same anticipation capabilities that were used effectively in two other scenarios were available to the subjects. In fact, only one pilot appeared (Subject 7) to use the anticipation capabilities by directly referencing the EVSD when calling ATC to report that the aircraft would be unable to make the crossing. It is hypothesized that a lack of experience with the VNAV system may have been a contributing factor. Due to a simulator anomaly, the timing results for subject 4 were rendered invalid and so were not included in the analysis.

Target Error by Pilot Flying during Non-Precision Approach

In this scenario, the Pilot Flying misdials the MCP altitude and misses an intermediate waypoint. During the non-precision approach, the researcher acting as Pilot Flying deals with slowing the aircraft by dialing the correct speed targets, and guiding the aircraft through each altitude restriction. During one such descent, an intermediate altitude target between the Final Approach Fix and the Missed Approach Point is inadvertently omitted by the Pilot Flying (researcher) causing an altitude deviation.

Altitude deviation in this scenario was measured by how far below the correct altitude the aircraft was when the subject recognized that a mistake had been made by the Pilot Flying. In this

233

MODE ID	V/S	HOLD
PATH	-1000 E	1900 E
SPEED	190 T	190 T
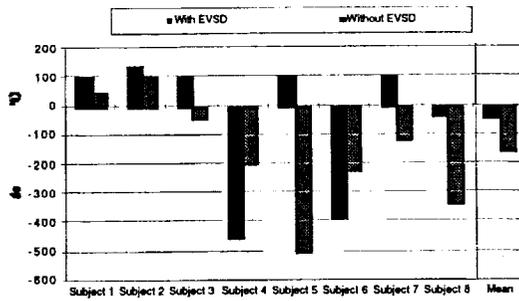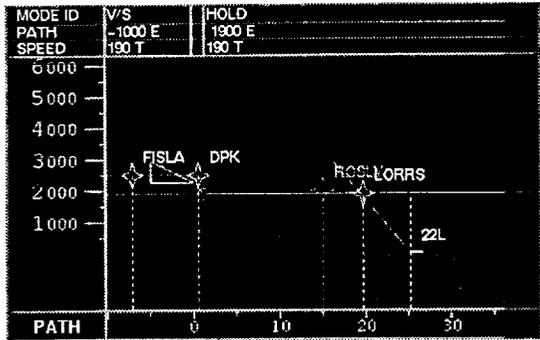
Figure E.8:   Target Error by Pilot Flying during Non-Precision Approach

context, a negative number represents that the subject realized the error before an altitude deviation occurred, a positive number represents an actual altitude deviation.

As seen in Figure E.8, using the EVSD, subjects were found to deviate from the target altitude by 117.1 feet more than without, but this value has little statistical significance in light of the sample size. It was also observed that subjects watched the Pilot Flying dial the incorrect value in the MCP rather than observing the mistake elsewhere. One pilot commented on the fact that the EVSD was not the correct display to use to catch this error, and that good Cockpit Resource Management should prevent this type of mistake from occurring.

Underspeed during Altitude Change due to ATC Directive

Overconstraining ATC directives cause the aircraft to lose speed to the point where the low speed envelope protection function of the aircraft is engaged. After switching the aircraft into Vertical Speed mode, and dialing in the target speed, it becomes clear that the last ATC amendment results in a target that exceeds the climb performance capability of the aircraft, as shown in the figure. The lower climb rate in the envelope protection mode prevents the aircraft from making its crossing restriction, and from maintaining the vertical speed target.

Altitude deviation in this scenario was measured by how far below the correct altitude the aircraft was when the subject recognized that a mistake had been made by the Pilot Flying. In this context, a negative number represents that the subject realized the error before an altitude deviation occurred, a positive number represents an actual altitude deviation.
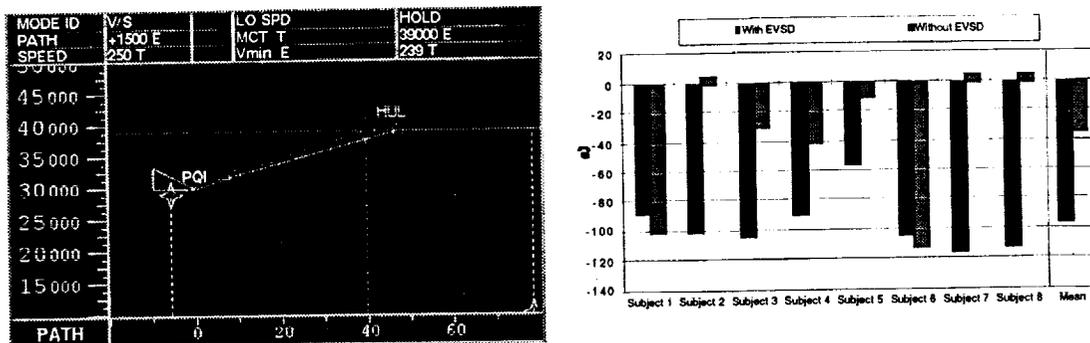
Figure E.9:   Underspeed during Altitude Change due to ATC Directive

Figure E.9 shows that the anticipation functionality of the EVSD allowed pilots to predict the underspeed situation 60.6 seconds sooner with the EVSD than without, at a confidence level of 95%. Negative times refer to the pilots anticipating the incident before it actually occurs. In addition to catching the error, several subjects recorded the distance by which the crossing restriction would be missed and additional related information, such as the altitude at which the envelope protection mode would be engaged, the location at which the low speed mode would be engaged, and so on.

Unexpected Climb during Approach due to Flap Overspeed

In this scenario, the aircraft attempts to climb to the Missed Approach Altitude due to a flap overspeed condition caused by Pilot Flying error. While lowering the flaps a few seconds too early, the PF engages the High Speed mode of the AFS. Since the MCP is dialed to the Missed Approach Altitude (above the current altitude), the aircraft adds suddenly pitches up and adds power in an attempt to capture the higher altitude. This particular problem emulates an event that occurred on an A310 in Orly, France in September 1994.

Figure E.10 shows that the subjects reacted to the sudden climb 0.8 seconds slower with the EVSD than without, but the confidence level of this value is low. This value is also within the noise level of human reaction time. In this scenario subjects responded by viewing the pitch up of the aircraft in the PFD and immediately calling in a go-around condition to ATC. Once this was completed, subjects considered what caused the situation and examined the EVSD when it was available.
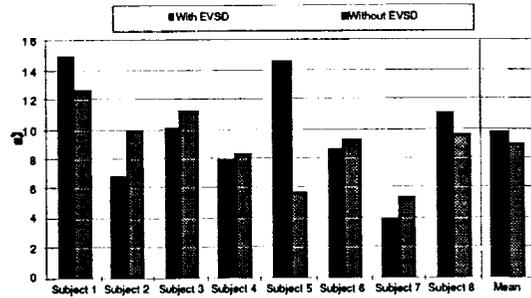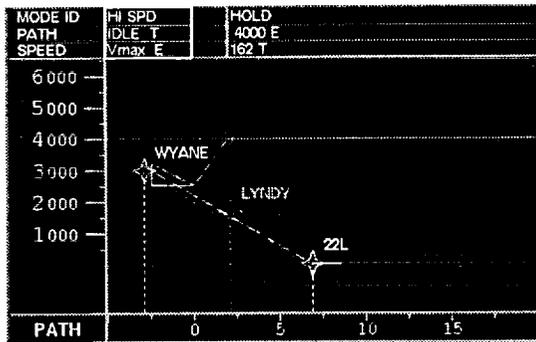
Figure E.10: Unexpected Climb during Approach due to Flap Overspeed

E.4.3 Subjective Results

In addition to the scenario specific results, each subject was asked to fill out a survey to assess the usefulness of the EVSD in various situations and to determine the usefulness of specific elements of the EVSD.

Subjective Value of the EVSD

Subjects were asked to rate the value of the EVSD on a scale from *Very Valuable* to *Very Detrimental*. The results of this questionnaire are shown in Figure E.11. All of the subjects felt that the display was at least somewhat valuable. Subjects were volunteers for this experiment, so that these results may be biased by having subjects which were predisposed to new technology in the cockpit.
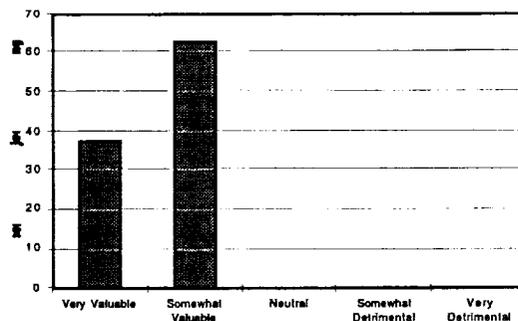
Figure E.11: Subjective Questionnaire: How Valuable was the EVSD

236

Comparison of EVSD with Current AutoFlight System

Subjects were also asked to compare the EVSD to the current AFS for a variety of tasks on a scale from *Significantly Better* to *Significantly Worse* as shown in Figure E.12. These results show that the subjects found the EVSD useful in a wide range of tasks, from inner loop target monitoring to augmenting overall situation awareness. This is not consistent with the objective results, which seem to point to more effective usage in situations where there was a strategic advantage to the information on the EVSD. The display tended to be less useful in instances where another instrument provided the same information, especially when the task involved tactical types of inner loop monitoring.
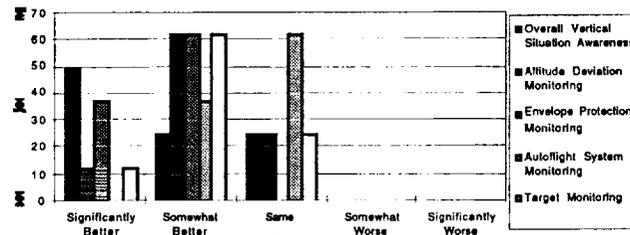
Figure E.12:   Subjective Questionnaire: Comparison between the EVSD and Current Vertical Feedback Mechanisms

Subjective Value of Elements of EVSD

Finally, subjects were asked to rate the usefulness of specific elements of the EVSD on a scale from *Very Valuable* to *Very Detrimental*. Each of these were scales with five levels, with the middle value being neutral. In addition, subjects were asked additional comments on the EVSD and features they felt were missing or required. The results of this questionnaire are outlined on Figure E.13. Once again, for each question, the percentage of subjects responding in that category is shown. Figure E.13 shows that subjects were not concerned with the control allocation, or the redundant target state information provided in the top bar of the EVSD. The interaction of the Green (Aircraft Path) Line, the VNAV Path information and the graphical target states were cited as being even more useful than the individual elements.

Subjects were also given the opportunity to comment on features that they felt were missing on the prototype EVSD. 75% of the subjects were interested in seeing terrain information on the display and another 25% were interested in seeing vertical weather information. It should be
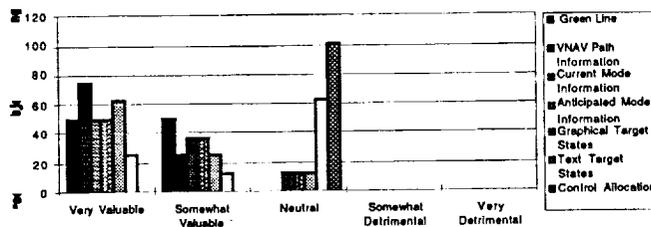
Figure E.13:   Subjective Questionnaire: Value of Specific EVSD Elements

noted that terrain information drawn from a database has been incorporated into the current EVSD design (as seen in Figure E.2), but the feature was not considered mature and stable enough to incorporate into this experimental evaluation.

E.5   Conclusions

Automation mode awareness problems have been reported by operators of many air transport aircraft. An examination of current generation AutoFlight Systems and a review of the ASRS database highlighted a lack of feedback in the vertical channel of aircraft automation. It was hypothesized that many of the incidents involving mode awareness problems could be mitigated by increased feedback in the vertical channel through an Electronic Vertical Situation Display.

An EVSD was prototyped which had four major display features: the current mode, anticipated modes, transitions into anticipated modes, and the consequences of the current state of aircraft automation. To evaluate the utility of the display, an experimental set of test scenarios was developed based on a representative set of known mode awareness problems from the ASRS review. Commercial airline pilots with glass cockpit experience were used as subjects in the experimental evaluation of the EVSD.

The Electronic Vertical Situation Display was found to significantly improve mode awareness understanding and the detection of mode awareness problems in both subjective and objective measures of subject response. Objective results were particularly strong when the anticipation functions of the EVSD could be used to foresee an event before it actually occurred. Amalgamated ratings of pilot understanding of mode awareness problems over the full set of scenarios increased in a statistically significant manner when the EVSD was available. In addition, subjects were much more specific when reporting problems to Air Traffic Control. For

example, rather than simply reporting that they were unable to make a crossing restriction, subjects would also report how far past of the waypoint the altitude would be acquired. Several subjects also mentioned the additional utility of having a vertical image of the aircraft's programmed flight.

The subjective survey results showed all of the subjects finding the display at least *Somewhat Valuable*. Display elements of the EVSD were also rated individually, with no elements being rated as detrimental and certain elements, such as the Aircraft Path Line and the vertical depiction of the VNAV trajectory being rated as *Very Valuable* by at least half the subjects. When subjects were asked to compare the EVSD with the mode feedback available in the cockpit available, they felt that overall situational awareness was improved, as was altitude monitoring and envelope protection monitoring.

Based on the positive results of this preliminary study, further evaluation of the EVSD concept appears warranted. Several issues remain to be addressed before a vertical display can be incorporated into current glass cockpits. For example an EVSD implementation must be designed to be compatible with specific AutoFlight Systems. Specific mode symbologies and names, scaling concerns, and colour conventions must be addressed, along with issues of retrofitting this display to current aircraft. This also entails finding an appropriate location for the EVSD in the valuable real estate of the modern cockpit.

Appendix F

Electronic Vertical Situation Display Questionnaire

F.1   Subjective Questionnaire

1. How valuable was the Electronic Vertical Situation Display:

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

Why?

2. How does the EVSD compare with the current AutoFlight System for:

Overall Vertical Situation Awareness

Significantly Better	Somewhat Better	Same	Somewhat Worse	Significantly Worse

Altitude Deviation Monitoring

Significantly Better	Somewhat Better	Same	Somewhat Worse	Significantly Worse

Envelope Protection Monitoring

Significantly Better	Somewhat Better	Same	Somewhat Worse	Significantly Worse

Autoflight System Monitoring

Significantly Better	Somewhat Better	Same	Somewhat Worse	Significantly Worse

Target Monitoring

Significantly Better	Somewhat Better	Same	Somewhat Worse	Significantly Worse

3. What do you consider the

 BEST feature of the EVSD:

WORST feature of the EVSD:

feature that is MISSING on the EVSD:

4. Please rate the value of each of the following features of the EVSD:

Green Line

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

VNAV path information

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

Current Mode Information

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

Anticipated Mode Information

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

Graphical Target States
(MCP line / G/S line)

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

Text Target States
(in top window)

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

Control Allocation

Very Valuable	Somewhat Valuable	Neutral	Somewhat Detrimental	Very Detrimental

5. Any additional comments or suggestions?

F.2   Briefing for Experiment

The experimental methodology is to fly a series of "mini-scenarios", each consisting of a short (about 5 minute) session of observation of the autoflight system maneuvering via the standard methods (EHSI display, ADI, MCP). Instead of actively controlling the aircraft, the subject, acting as pilot-not-flying will allow the researcher (acting as the First Officer and pilot flying) to react to all ATC commands and maintain tactical control of the aircraft. The subject is encouraged to ask for scale changes on the displays and other passive observations.

If at anytime during the scenario, the subject feels that they should query the Pilot Flying about the behaviour of the aircraft, or of any concerns about the performance of the PF, the subject should articulate this concern via the "Press to Talk" button. Situations in which the button should be used include, but are not limited to:

Inability to meet ATC directive, clearances or procedures.
Any sort of unsafe condition.
Any perceived fault with the aircraft.

Essentially, the subject CANNOT hit the button too often. Any situation which is of concern should be articulated.

In addition, the subject has another task, which is to keep a load centered. The display for this load is in the top left hand side of the screen. The side stick controller should be moved left to right to center the load. An error value based on the duration of the misbalancing and the degree of misbalancing will be recorded.

F.2.1   Training Session

Flight Level Change - climb, descent
       Control Altitude
       Note Targets
       Note Control Allocation
Vertical Speed - climb, descent
       Control Altitude
       Note Targets
       Note Control Allocation
High Speed - climb, descent
       Control Altitude
       Note Targets
       Note Control Allocation
       Note mode change circle
Approach Mode
       Arming
       extrapolation vector

F.2.2   The Simulator:

The Aeronautical System Laboratory's Advanced Part Task Simulator is a medium fidelity simulation of aircraft operations, with particular attention paid to displays and autopilot logic. However, since it does not emulate any single aircraft (it is a mixture of the B757/767/747-400 with a B737 MCP and some display conventions from the MD-11), a few of the characteristics it exhibits must be noted and understood.

1. The VNAV function consists of two modes, VNAV PATH (where the a/c tries to follow a specific path at the target speed) and VNAV SPD (the a/c tries to maintain a target speed during vertical maneuvers). The transition from VNAV PATH to VNAV SPD occurs when due to an unforeseen AFS target conflict, the a/c speed exceeds the target speed by 10 kts. This should be considered a legitimate reason to use the press to talk button.

2. Envelope protection boundaries are shown on the speed tape on the ADI. The red marker indicate Vmin and Vmax. If these boundaries are exceeded, the a/c autopilot will automatically switch to either a LOSPD or HISPD mode to maintain the Vmin or Vmax speed until it captures the target altitude.

3. The Altitude Capture Logic on the simulator flies a .05g arc when close to the target altitude. It sometime overshoots slightly. During this capture, the a/c will be in the ALT HLD mode.

4. The localizer capture/tracking logic is somewhat coarse: it overshoots the localizer several times before tracking down it tightly.

5. Some of the scenarios do *not* have anything worth reporting. These are inserted to keep the task interesting and make sure that the task does not become too easy. If the subject feels that the scenario did not have any problems, he or she should simply state so on the questionnaire.

6. A great deal of effort has been put into the fidelity of the simulation. Correct cues will be available from the MCP, the EVSD (if it is currently visible), the ADI, the CDU and the EHSI. The throttle (when on manual), flap and landing gear are all functionally operational and may be used for additional information.

F.3   EVSD Conventions

magenta	FMS/active waypoint
white	MCP/Glide Slope
green	active/predicted
amber	caution
brown	terrain

Scaling

x-axis: path   distance in nautical miles along FMS path (LNAV mode engaged)
      range   distance in nautical miles straight ahead of a/c (HDG mode engaged)

y-axis: altitude in feet above mean sea level

Windows/Header

MODE ID   name and x-axis location of active or armed mode

VPTH	vnav path
VSPD	vnav speed
HOLD	altitude hold
FLCH	flight level change
V/S	vertical speed
LAND	autoland/glide slope track
HISPD	hi speed envelope protection mode
LOSPD	low speed envelope protection mode

VERTICAL   name and x-axis location of vertical target and means of control

nnnnn	altitude target (e.g. 28000)
+/- nnnn	vertical speed target (e.g. -2500)
PATH	FMS-computed vnav path
G/S	glide slope
FLARE	flare segment of autoland
MCT, CLB, IDLE	thrust limits

Postfix displays the control mechanism used:
      E   control by elevator/pitch
      T   control by thrust

SPEED   name and x-axis location of speed target and means of control
nnn indicated airspeed target (e.g. 250)
.nn mach target (e.g. .80)
Vmin/Vmax min/max flight envelope speeds

Postfix displays the control mechanism used:
      E   control by elevator/pitch
      T   control by thrust

F.4   Background Information

Information concerning your aviation background will help us to more accurately assess some of the variables that affect the performance of the pilots. *All information that you provide will remain completely anonymous.*

Personal Data / Miscellaneous Information

1. Age: _____   Sex: Male ( )   Female ( )

2. How were you *initially* trained to fly?   Civil ( )   Military ( )

3. Civil Experience:

   Total Civil Pilot Flight Time: _____

   Pilot Ratings Held:

   Fixed Wing: ATP ( )   Commercial Pilot ( )   F.E. Written ( )

   Rotary Wing: ATP ( )   Commercial Pilot ( )   Other _____

Electronic (Glass) Cockpit Equipped Transport Aircraft Flying Experience

	Aircraft Type	Flight Hours (Approximate)	Position
1.	_____	_____	_____
2.	_____	_____	_____
3.	_____	_____	_____
4.	_____	_____	_____
5.	_____	_____	_____

Estimated Flight Hours in 1994: _____

F.5 Pilot Understanding Questionnaire

Scenario Number: Pilot Number:

Why did you use the "press to talk" button?

What caused the event? Why did it happen?

What cues did you use to determine that the event occurred?

What would have eventually happened if you had not stopped the scenario?

Appendix G

Evaluation of the Operator Directed Process

Usability can be measured through multiple means, including rapid automation training and adoption rates. In order evaluate the efficacy of the ODP in improving device performance, a planned software development project was identified upon which to test the process. This project was chosen based on availability and access to the planned and ongoing development effort.

G.1 Background

In order to investigate implementing of the Operator Directed Process, an early instantiation of the ODP was used in the development of an automation tool to assist physicians. This tool supported portable electronic documentation, and was first used in a billing scenario. Given the similarities between the nature of the hospital operational environment and the flight environment, this study appears to validate the Operator Directed Process for both fields. In addition, this proved to be an excellent opportunity to field test the process within the limited resources of a small company. The details of the development of this tool, and experimental results from a preliminary pilot study will be presented.

The medical domain has similarities to the flight domain in terms of the relationship between operators and automation. Beyond the obvious life-critical characteristics, they are similar in other respects. Both pilots and physicians follow procedures in order to complete many tasks, each also has a set of highly refined manual skills as well as rule- and knowledge-based understanding, each group is highly trained in a specialized tasks, both augment capabilities through automation, and each group has members which are reluctant to use new technology.

The test product which was developed was an "Electronic Billing Card." It was designed to replace the existing system consisting of paper cards. These cards are the sole repository of billing information generated by physicians in some hospitals. As such, they represent a single point failure in the payment system for physicians. Replacing them with an electronic version could result in large cost savings to both the physicians and the hospital by reducing the number of cards

and charges which are lost, accelerating the process by which charges are reported, and other mechanisms.

Figure G.1: Paper Billing Card

Billing cards are typically available at nurses' stations throughout the hospital. To create a card for a new patient, a doctor has to visit the nurses' station and "stamp up" a card. The process of stamping up places patient demographic information onto the card, as see on the top right section of Figure G.1. Each day that a physician sees a patient, a billing code is placed into the appropriate box on the card. When a physician has completed a card and wishes to turn it in, the entire stack is submitted to the billing office. An example is shown in Figure G.1. This card is for Alex Watkins (patient ID 237-52-16), who was first seen on June 13 by Dr. John Goodbelly. On that date, the physician performed a code A (an "Admission Visit, Initial (Level 3)") with codes F, G, G, and G on subsequent days.

G.2 Functional Analysis

The functional analysis of the current billing process was examined through observation and focused interviews with physicians. This analysis showed a primary use for the billing cards as well as several secondary ones. The distinction between primary and secondary uses and their grouping was corroborated by active physicians. Some additional capabilities, not available through the current system, were also identified through this analysis and served to accelerate adoption into the operational environment.

250

Primary Function of Billing Card

The primary goal of the billing card is to allow the recording of billable information in a structured form that is easily submitted to the next stage in the billing process. This requirement implies that the cards must be portable and highly available for use at any time during a physician's rounds. The availability and portability are an issue, since physicians find it difficult to remember and transcribe billing information after a service has been rendered. By allowing the capture of this information at the "point of care," revenues can be maximized.

Table G.1:   Primary Functions Required of Electronic Billing System

Primary Functions
Allow structured billing
High portability
Wide availability

The format of the cards is designed to minimize the amount of writing by the physician by creating a very structured environment for the capture of information. This structure is imposed by the form required by the United States Health Care Financing Administration (HCFA), which organize billable events into organized groups. On the bottom of the card shown in Figure G.1 are the short hand codes for these individual billable events.

Secondary Functions of Billing Card

In addition to its stated primary function, billing cards are used as a longer term record of billings and interactions with patients. By examining a billing card, in conjunction with clinical charts, a physician can gain a better understanding of their own interaction with a patient. Some physicians also jot specific notes about patients such as lab results, critical elements of the patient history and others pieces of information on the blank reverse surface of the card. The conjunction of these minimal notes and the diagnostic and billing codes on the front of the card create a record of the patient which is always available to physician. This is in contrast to the full clinical record which is stored in a less accessible single location and serves as a repository of the interactions and notes of all physicians with a patient. These aspects of the card's use prevents physicians from handing them in to the billing office on a daily basis, since physicians are unwilling to relinquish the information maintained on each card.

251

Table G.2:   Secondary Functions Required of Electronic Billing System

Secondary Functions
Record of billing and interactions
Secondary notes
Active list of patients

Billing cards are also used as an "active list" of patients. Each card is a physical placeholder for recently seen patients. These cards can be organized in various manner to streamline the path of a physician on their rounds, or to keep track of specific patients.

Supplementary Functions of Billing Card

By moving the billing capture from paper cards to an electronic device, additional functions can be used to assist physicians and to streamline the billing process. The first is that by maintaining information from multiple patients on a single electronic device, it is possible to dynamically sort the list of patients. Many physicians currently maintain their stack of cards in order of admission date, patient name, or patient location. The electronic version can automatically provide this function.

As mentioned earlier, the paper cards are the only repository of billing information. To provide some insurance over losing cards, physicians have taken to photocopying the cards, or using specialized cards with carbon paper to retain receipts of the data before submission to the billing office. Since the electronic device is still a single source of captured billing information, a necessary function is to archive this information. This is even more critical for an electronic device both since it has more failure modes than paper cards and because the granularity of the loss is much coarser. Rather than losing a single card, worth hundreds of dollars, the loss of an electronic device may result in the loss of charges worth thousands of dollars.

Table G.3:   Supplementary Functions Required of Electronic Billing System

Supplementary Functions
Sorted list of patients
Archival of billing information
Maintain updated guidelines

In order to evaluate the value of the final supplementary function, listed in Table G.3, it is necessary to understand the guidelines used to determine the correct billing code. Guidelines, specified by Health Care Financing Administration, enumerate the services which must be provided—and documented on the clinical chart—by a physician in order to warrant charging using specific billing code. These guidelines are very detailed, and complete characterization requires data from multiple documents. In addition, the guidelines change on an annual basis. The importance of accurate billing has recently become particularly important due to efforts by the US Federal government to reduce instances of inaccurate or excessive charges. Mistakes can be accompanied by large fines to the physicians ($10 000 per item) and even larger fines to hospitals and organizations. From 1997-1999, the hospital at the University of Pennsylvania was assessed with a fine of $30 million, Thomas Jefferson University hospital $12 million, and University of Pittsburgh Hospital $17 million.

This financial liability introduced an additional function into the electronic billing card, which was to provide an up-to-date set of guidelines to physicians to minimize incorrect billing. Currently, hospitals and physician groups create shortened versions of these guidelines to allow accurate billing. These guidelines are distributed to physicians via seminars, pamphlets, and "quick reference cards," and end up stored in physicians' memories. A required function of the electronic billing card is to enable this same information to be made available at the point of care directly on the electronic device.

G.3   Automation Model

The fundamental automation model which was designed for the electronic billing card was to create software which emulated an "enhanced" stack of billing cards. The stack of cards is a metaphor which is understood by physicians and allows for easy manipulation of patient billing data. Each individual card is also modelled on the visual structure of the paper cards, though modified in order to fit within the size limitations of the electronic device. This is a very simple model of automation for the device.

The environment for the device also needs to be modelled. As discussed earlier, physicians currently have to visit the nurses' station in order to create new cards and populate them with

demographic information. These stations are located on each floor of the hospital. This process was recognized as an instantiation of an existing procedure used to structure the physician workflow. The electronic billing card was designed to coexist and follow the same procedure: the larger environment in which the electronic billing card system exists consists of synchronization stations located at each nurses' station. In an analogous manner to the current system, physicians visit these stations to populate patient cards with demographic information. This serves to minimize the shift in work flow of the physicians. When physicians synchronize the electronic billing cards to gain demographics, a sophisticated back-end infrastructure completes several additional tasks: submitting new codes to the billing office, updating guidelines as necessary, and archiving the data on the electronic device.

Billing Card Interface

The image on the right of Figure G.2 shows a single electronic card associated with a patient. This screen allows immediate access to the patient's name, location, and diagnosis. Beneath the demographics are boxes in which codes can be entered. The form of the code entry boxes mimic the organizational structure seen on the bottom of the original paper billing card, shown in Figure G.1. Either a code can be selected directly, or it can be "constructed" using the hierarchy of category and level. The enhancements which have been provided to this model fall within the bounds of this metaphor. The first is the ability to sort patients based on differing criteria. The left image in Figure G.2 shows an example of multiple of the list of multiple patients. The header of each column is shown as underlined if it is being used to sort the list of patients.
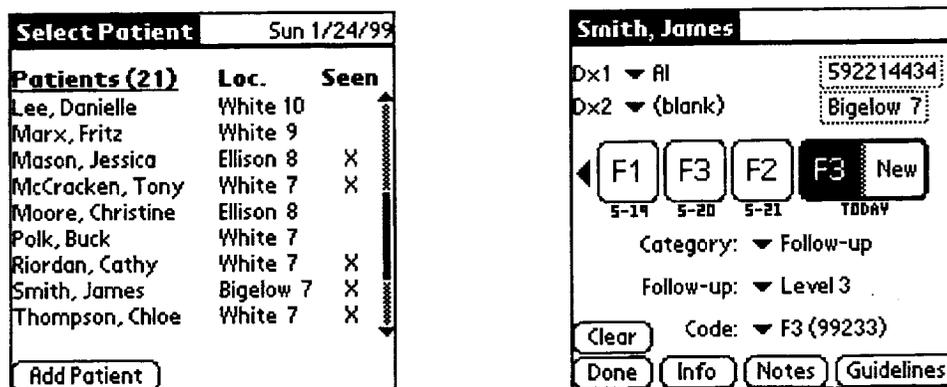
Figure G.2:   List of Patients and Patient Card

254

The bottom of the screen contains buttons to continue to less common information, such as the complete demographics and the notes regarding the patient, shown in Figure G.3. This information was spread across multiple screens and organized based on feedback from physicians to place the most critical information immediately available and to push less used information onto secondary screens.

Figure G.3:   Additional Screens of Data

Billing compliance guidelines made available directly on the device, organized by the billing codes with which they were associated. The stack of cards now has an additional database which contains up-to-date guidelines which are immediately accessible by the physicians. Figure G.4 shows an image of the billing guideline screen.

Figure G.4:   Compliance guidelines

G.4   Training Material and Procedure Design

In this prototypical instantiation of the Operator Directed Process, formal training material was not created before software development. Instead, "storyboards" were created of each screen by a "Subject Matter Expert"—an engineer who worked closely with the physicians to understand the requirements and translate them into a form suitable for use on the organizer. These storyboards, shown in Figure G.5, are a visual layout of the screens and describe the underlying behaviour of each of the elements as viewed by the operator.
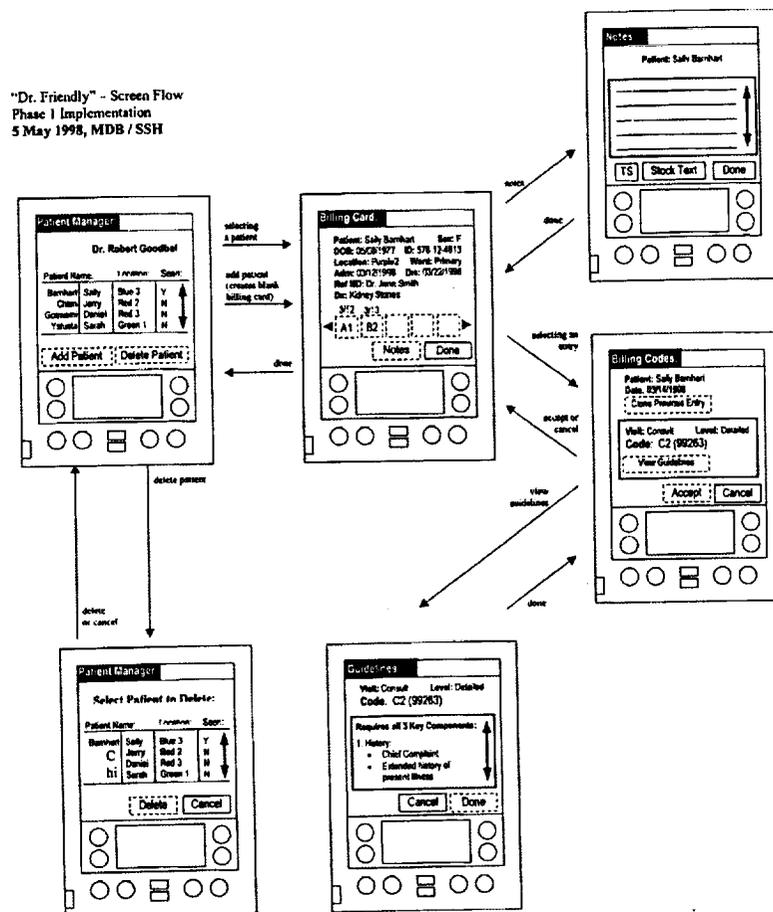
Figure G.5:   Example Storyboard

The storyboards were void of underlying implementation details. Each storyboard described a standard set of interactions, or procedures, which a physician would have with the automation.

Storyboards became a system description suitable for the basis of training and of procedural design. Specific paths through the storyboard represented procedures describing usage. This capacity was not anticipated, or anticipated as being a requirement, before application of the ODP and served to refine the model. The procedures defined were also placed into the training manuals in a series of "How Do I?" questions, shown in Table G.4.

Table G.4:   "How Do I?" Procedural Guidance

Start the Electronic Billing Card?
- press the far right button, accented in gold

Turn the screen backlight feature on/off?
- press and hold the green power button for a few seconds

Sort the list of patients?
- tap the column heading you wish to sort on
- a second tap reverses the order
- a very useful ordering is to first sort by Location, then by Seen
  to create a list of unseen patients sorted by their location

View patients past the bottom of the screen?
- use the hardware scroll buttons to view additional patients
- alternately, use the scrollbar to view additional patients

Enter today's code for a patient?
- select the patient from the Select Patient Screen
- select the appropriate Category and Level
- alternately, select the Code directly
- tap the Done button to return to the Select Patient Screen

Enter or change a diagnosis?
- select the patient from the Select Patient Screen
- select the appropriate diagnosis
- or choose (blank) to have no diagnosis
- tap the Done button to return to the Select Patient Screen

View a Guideline?
- select a patient from the Patient List
- tap Guidelines button
- select the appropriate Guideline
- tap the Done button to return to the Billing Card Screen
- tap the Done button to return to the Select Patient Screen

In addition, training sessions were carried out concurrently with development. At appropriate stages of design the prototypes were shown to physicians and administrators. These physicians were given demonstrations and often proffered immediate feedback. The model that was presented to the physicians during these sessions was identical to the automation model, that of a "smart" stack of billing cards.

During some of these sessions, the operational concern of "multiple codes" was raised by physicians. This capability was later added to the system, and that process is described later in Section G.5.

As part of the preliminary evaluation, a complete set of documentation for the project was developed. During this development, minor errors and inconsistencies were found in the system. Rather than reflecting these errors in the documentation, the system itself was modified. Anecdotally, it appeared that very few physicians had the patience or desire to pore over the small training manual. Instead, using the knowledge gained during the personal training session, they explored the system directly.

G.5 Iterative Design and Software Development

The development of the prototypes was highly iterative. Once a set of specifications has been established, multiple generations of prototypes are created with growing degrees of functionality. Each prototype is evaluated to determine if the new capabilities are fit within the metaphor and are useful to the operator. In the case of the electronic billing card, a total of six iterative design cycles have been completed, and the project is currently in the midst of the seventh.

The first stage of design consisted of sketches of the layout and function of the product on paper. This sketch was used to establish the hierarchy and interaction between various screens and establish the scope of the project. The next generation design was created using a Rapid Application Development (RAD) tool which generated screens and a simple database. Using this tool allowed some early usability testing to be done with physicians and to determine problems with the product. What was noted during this evaluation is that the speed with which the unit switched between screens was used by users as a measure of its responsiveness. At this stage, many of the more complex operations (such as actually entering a code) were not available.

At this point a more robust development environment, since the limitations of the RAD tool had been reached. A full prototype was created based on a storyboard document which described each screen and its behaviour in detail. This prototype had the majority of the capabilities required to fully demonstrate the billing card product in isolation. The next several revisions modified the underlying architecture and databases of the billing card system in order to allow the

product to be more flexible, configurable, and maintainable, but did not change its overall form or user experience.

Identification of Additional Functionality Requirement: Multiple Codes

The exception to this monotonic refinement was the creation of a new feature which was found to be required by the physicians: "multiple codes," the ability to bill for more than one code per day. On the paper card, this was accomplished by writing the letter code very small, so that multiple codes could be entered in the same box. It was decided that this would be an inappropriate solution for the electronic device, since the size of readable text is limited by the resolution and size of the screen.

A second design and development effort ensued in which a second round of storyboards were created on paper by the Subject Matter Expert. It was found that the evaluation of the implementation was dependent on accurately recreating its dynamic behaviour. As such, physician evaluation was done after the storyboards were prototyped on the device. The final solution was to retain the large format boxes and link boxes together if they were associated with the same day. In Figure G.2, the "Today" code box has one code, and is can have another entered in the box labelled "New." This feature required a re-evaluation and extension to the underlying automation model and a modification to the training material in order to be added to the system.

Resolution of Edge Conditions

One additional anecdotal point should also be made. During the development of the system it was found that "edge" conditions were regularly found by programmers, where the behaviour of the system was not fully defined by the storyboard documentation or training material. As an example, consider the situation where a physician is selects a code box to modify, then uses the left/right arrows to view another code, thereby sliding the active box off screen. Should the physician be able to modify a non-visible box? Should the box reappear if modification is attempted? Should the physician be prevented from moving the active box off screen at all? These edge conditions were submitted to the Subject Matter Expert for resolution, and if necessary, back to the physicians. By doing so, non-nominal behaviours at the edges of the system were consistent with nominal behaviour.

Update to System During Study

During the operational evaluation, it was also found that physicians required the additional function of being able to remove patients from the devices. This capability had, in fact, been specified in early storyboard models of the systems, but was left out of the prototype due to time constraints and in recognition of the duration of the study. During the study, this capability was added to the prototype and immediately distributed to physicians to allow it to be evaluated. The modification to the system was recognized as a consistent extension to the model of the system and adopted by physicians.

G.6 Evaluation

A preliminary evaluation of the billing card system was undertaken at Massachusetts General Hospital with the goal of determine physician acceptance. Other electronic tools which have been introduced into the medical community have been stymied by a lack of physician adoption. During this evaluation, 11 physicians were chosen by the hospital administration to use the electronic billing card. The physicians ranged from computer experts to those who had not used electronic devices at all and spanned a total of 8 medical disciplines. The duration of the study was two weeks.

Objective measures were taken of physician usage of the device. These measures attempted to understand which screen were used the most often, where physicians had difficulties in manipulating controls, and to gain insight into overall usage patterns. Physicians were also asked to complete a questionnaire to measure their acceptance of the product and to determine which aspects required more effort. Physicians were compensated for their time by being given the handheld organizer. To allow the development of trust in the system, physicians were allowed and encouraged to maintain both paper and electronic copies of charges.

G.6.1 Objective Results

As part of the study, physician interaction with the electronic billing card was automatically monitored. Interaction events consisted of viewing a particular screen on the device, using a stylus to tap the screen, viewing guidelines, and so on, associated with this particular application. Other programs being used on the device were not recorded. A given session with the device

would result in dozens, if not hundreds of events. Each event was individually tagged for identification and was stored with both the time of the interaction (one second granularity) and any data associated with the interaction. The goal of this data collection was to measure the regularity of physician interaction and the rapidity of device adoption.

Figure G.6 shows the results of the automatic monitoring, grouped into daily segments during the fifteen day trial. Examining the data with one second accuracy was not insightful for overall usage patterns, but was helpful in identifying detailed physician interaction problems. The data shows that physicians' interaction pattern were heterogeneous, with some interacting on a daily basis, and other rarely. The data does not show a gradual adoption curve, but rather immediate usage. Instead, physicians appear to interact in a "bursty" manner. There are also long durations during which some physicians did not interact with the device. Specifically, days 2, 3, 9, and 10 fell on the weekend, resulting in limited physician interaction. The use of the device on successive days does not appear to have been influenced by this period of inactivity.

G.6.2 Survey Results

The survey was divided into three major sections. The goal of the first section was to gauge physicians' acceptance and high-level impressions of the product and the handheld computer platform. The next section made an explicit comparison between the billing card system and the existing paper-based system. The final section mentioned specific attributes of the system and asked about their importance to the billing task. The full survey is available in the appendices.

Acceptance Responses

The first set of questions was designed to elicit high-level responses from the participating physicians.

One of the major concerns with introducing new applications into the medical environment is the amount of training which will be required. Physicians are reluctant to waste even a small fraction of their time learning new non-clinical technologies. The billing card system was designed to be extremely easy to use through use of the Operator Directed Process. That ease of use was reflected in a single ten-minute training session being unanimously considered adequate.

261
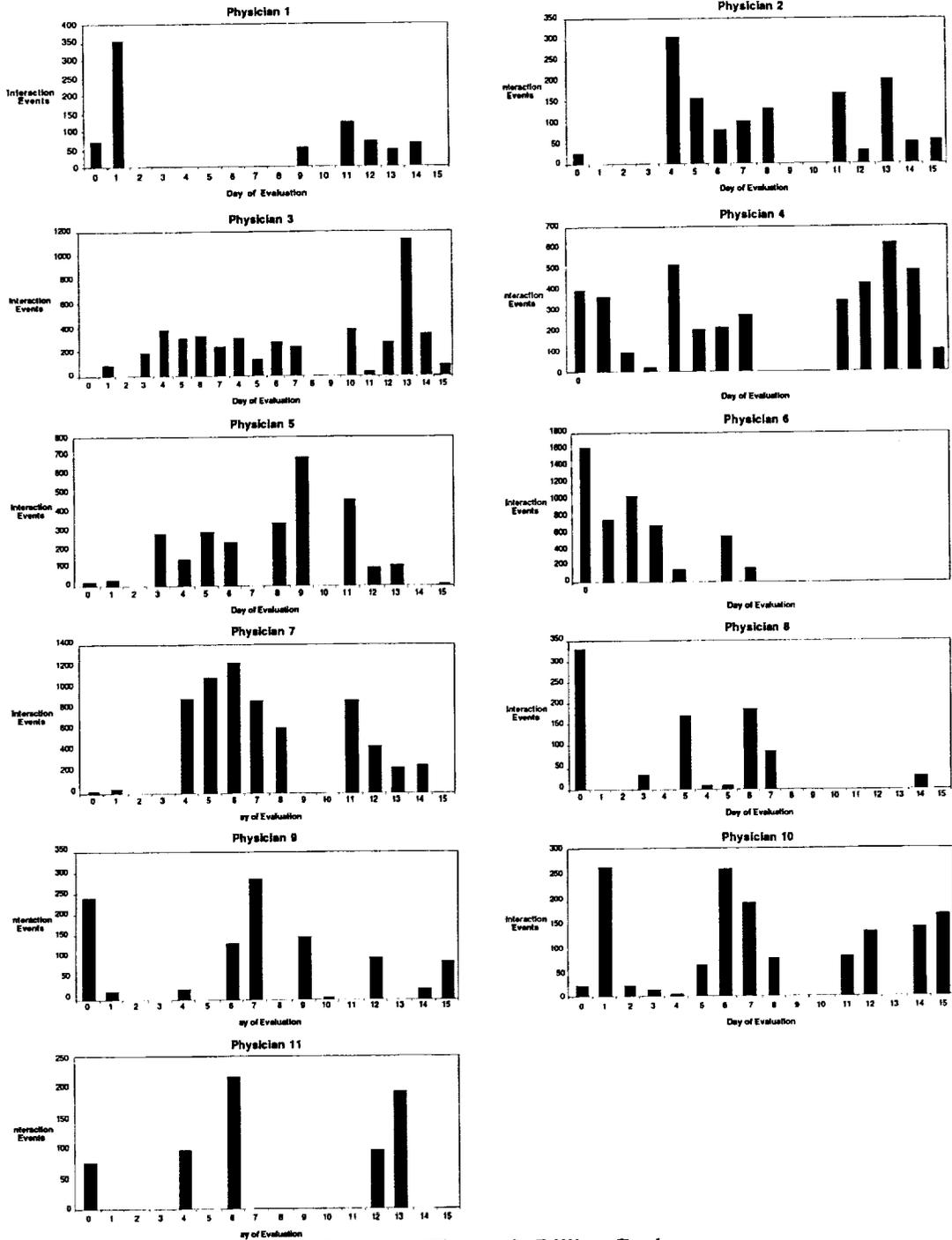
Figure G.6:   Physician Interactions with Electronic Billing Card
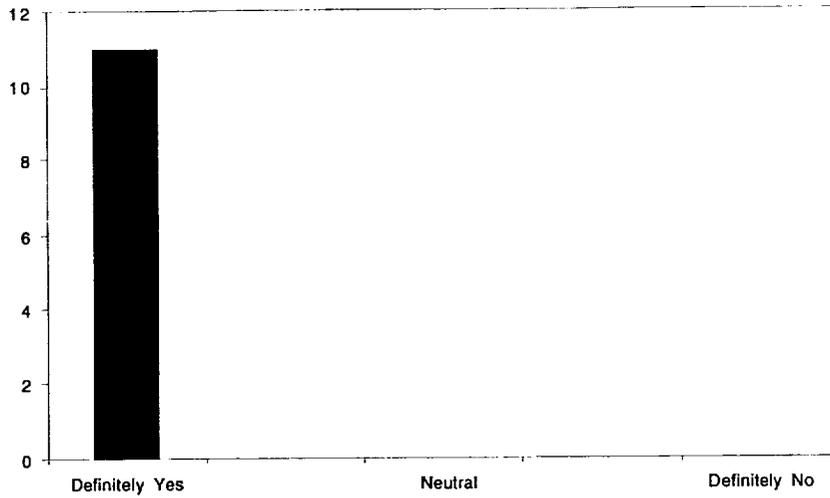
Figure G.7:   Did you find the electronic billing card training adequate

This result was also reflected in the next question, which showed that 90% of physicians found the system easy to use.
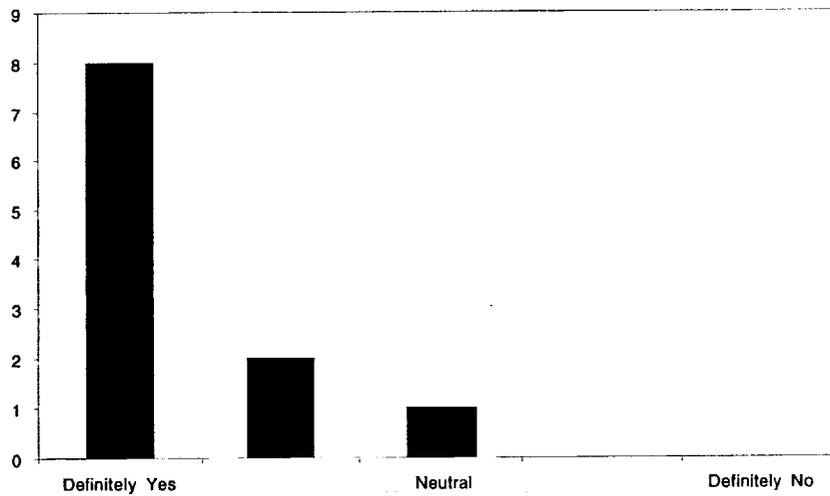
Figure G.8:   Did you find the electronic billing card simple, intuitive, and easy to use

Comparison to Paper-based System

The second section of survey attempted to compare the VIRTMED Billing Card system with the existing paper-based system.
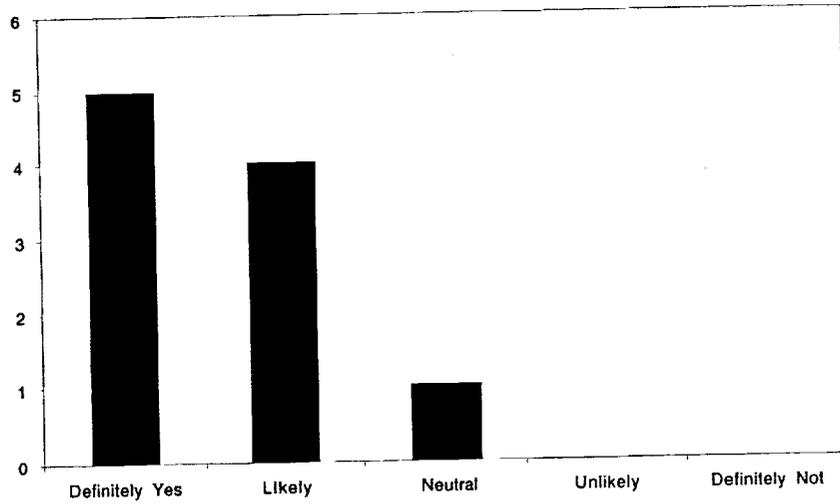
Figure G.9:   Would you want to use the electronic billing card on a permanent basis

The first issue was whether physicians wanted to use the system on a permanent basis. 90% of the responding physicians found that this was likely or definite after only having used the product for the short duration of the study. No physicians were opposed to using the system.
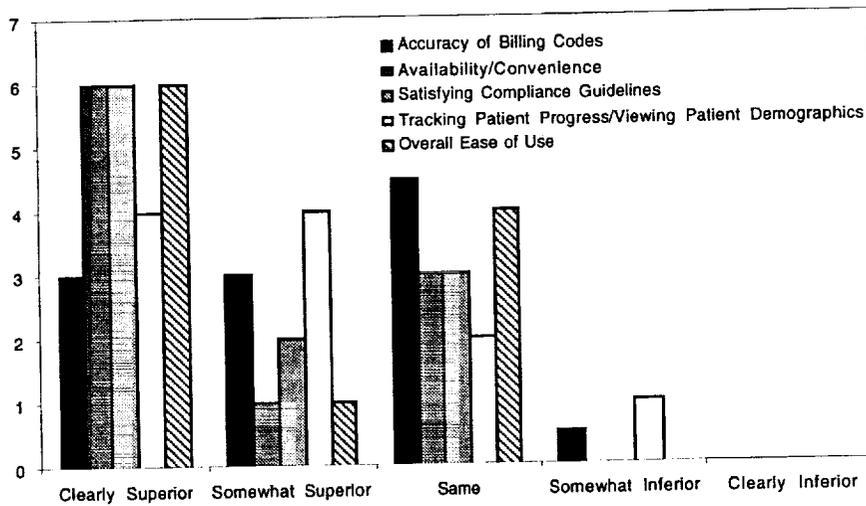
Figure G.10:   How do you feel the electronic billing card compares with the current paper system for...

The next set of questions explicitly compared specific attributes of the existing paper system to those of billing card system. Physicians found that the electronic system was superior in both "Availability/Convenience" and in "Satisfying Compliance Guidelines." The "Overall Ease of Use" was also considered superior. The perceived "Accuracy of Billing Codes" was not found to be detrimental. "Tracking of Patient Progress/Viewing Patient Demographics" was found to be somewhat inferior to the paper-based version by 18% of physicians, which is not unremarkable since this was not designed as a primary function of the tool. The questions were included to gain insight into directions for future products.

Specific Attributes

The final section of the survey asked physicians to comment on specific attributes of the electronic billing system in terms of their perceived value.
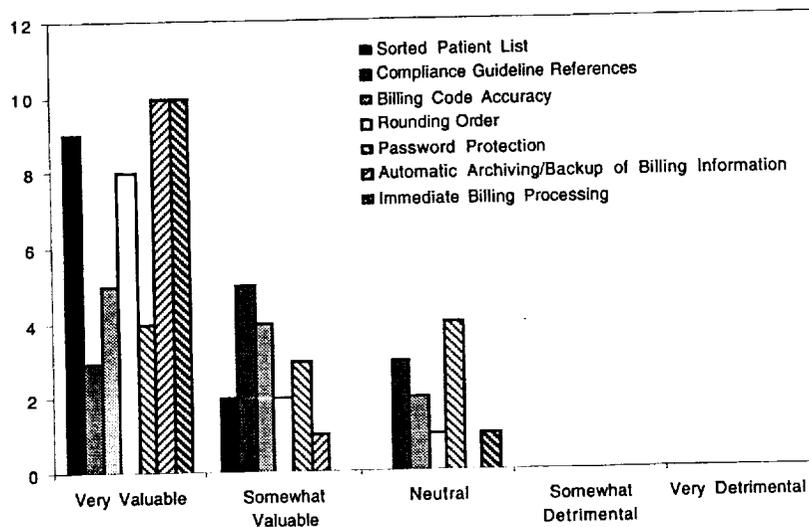
Figure G.11:   Please rate the value of each of the following features of the electronic billing card

Physicians found the "Automatic Archiving/Backup of Billing Information" and "Immediate Billing Processing" of billing information to be of paramount importance. In addition, the availability of a "Sorted Patient List" and control over the "Rounding Order" were found to be valuable.

G.6.3 Analysis of Operator Directed Process Evaluation

The rapid acceptance and minimal training required of the electronic billing card system is a single datapoint in the evaluation of the Operator Directed Process. Based on the subjective results, the physicians were able to be trained in the use of the system extremely quickly and were able to utilize it effectively for the duration of the evaluation. While this does not constitute a controlled experiment comparing the ODP with a conventional development process, the results are still compelling.