DOT/FAA/AR-11/28

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Flight-Critical Systems Design Assurance

July 2012

Final Report

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/AR-11/28		

4. Title and Subtitle	5. Report Date
FLIGHT-CRITICAL SYSTEMS DESIGN ASSURANCE	July 2012
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Herbert Hecht	

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
SoHaR Incorporated 5731 West Slauson Avenue, Suite 175 Culver City CA 90230	11. Contract or Grant No. DTFACT-08-C-00003

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation Federal Aviation Administration Northwest Mountain Region – Transport Airplane Directorate 1601 Lind Avenue, SW Renton, WA 98057	Final Report
	14. Sponsoring Agency Code ANM-113

15. Supplementary Notes

The Federal Aviation Administration Aviation Research Division COR was Robert McGuire.

16. Abstract

The safety and reliability of aircraft systems, and particularly of flight control and associated systems, are very high. When failures do occur, they are mostly due to the system encountering rare events that had not been foreseen by the equipment and system developers. The purpose of this research is to identify areas where the certification process can be improved to further minimize the incidence of mishaps.

This report describes a number of accidents and critical incidents and analyzes their causes. Issues of how requirements of certification and design guidance documents relate to the accident and critical-incident causes and how commercial design tools can contribute to minimizing the probability of critical incidents are also discussed. Requirements for handling rare events in highly integrated and complex systems are not explicitly addressed in the guidance documents; and the available design tools, at most, check for consistency among requirements but assume that they are complete. This report, therefore, deals with requirements for handling rare events, such as mode changes, and proposes a structure for the review process that may help eliminate the hazards that led to the observed incidents. A number of the most severe incidents were due to faulty redundancy management. Therefore, a follow-on effort should be undertaken to generate review guidelines for the certification of redundancy management provisions. There also was lack of guidance for the interfaces of flight control systems with area navigations systems, Traffic Collision and Avoidance System and supervisory functions; this also should be addressed in future efforts.

17. Key Words	18. Distribution Statement
Flight critical systems, Aircraft incidents, Review of requirements	This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	75	

ACKNOWLEDGEMENTS

Tony Lambregts, Chief Scientific and Technical Advisor, Federal Aviation Administration (FAA) Flight Guidance and Control Systems, shared his insight and information on failures in flight-critical systems. This report owes much to the guidance received from him.

Thanks are also due to the technical representative for the effort at the FAA William J. Hughes Technical Center, Robert J. McGuire, who arranged contacts with key FAA personnel and eased the way through administrative difficulties.

Many other FAA specialists provided essential information for this report, with contributions from Robert C. Jones and Linh Le being particularly noteworthy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AADL	Architecture Analysis and Design Language
AC	Advisory Circular
AD	Airworthiness Directive
ADC	Air data computer
ADIRU	Air data inertial reference unit
AFM	Airplane Flight Manual
AGL	Above ground level
AoA	Angle of attack
ARP	Aeronautical Recommended Practice
ASIAS	Aviation Safety Information Analysis and Sharing
ATSB	Australian Transportation Safety Board
CFR	Code of Federal Regulations
DGAC	Direction Generale de l'Aviation Civile (French equivalent of NTSB)
ECAM	Electronic Centralized Aircraft Monitor
EFIS	Electronic flight instrument system
EICAS	Engine-indicating and crew-alerting system
EIS	Electronic instrument system
EIU	EFIS/EICAS interface units
ESS	Essential services (Bus)
FAA	Federal Aviation Administration
FCA	Fault containment area
FCM	Fault containment module
FMEA	Failure Modes and Effects Analysis
GCU	Generator control unit
HB	Heartbeats
IDU	Integrated display unit
IR	Inertial reference
ISIS	Integrated Standby Instrument System
LCD	Liquid crystal display
MM	Maintenance Message
NASA	National Aeronautics and Space Administration
NGC	Northrop Grumman Corporation
NTSB	National Transportation Safety Board
OEB	Operations Engineering Bulletin
OO	Object-oriented
PFD	Primary flight display
PMS	Performance management system
PRIM	Primary flight control computers
RM	Redundancy management
rms	Root mean square
SAE	Society of Automotive Engineers
TCAS	Traffic Collision and Avoidance System
UML	Unified Modeling Language

EXECUTIVE SUMMARY

Although commercial air travel in the United States has an enviable safety record, the availability of powerful digital components motivates a drive to higher complexity, particularly in operating modes and redundancy management, that raises concerns about the ability to completely test the system under all conditions as well as to assess pilot proficiency in mastering all available resources under emergency conditions. For the incidents examined in this report, it was concluded that

- most of the incidents were due to multiple rare events, e.g., an equipment failure and vulnerability in the software that was intended to recover from the failure.

- tests for multiple rare events are currently, at best, performed on a hit-and-miss basis; the incidents are evidence of misses.

- to avoid such misses, requirements for coverage of rare events must be generated and reviewed throughout the system life cycle. The waterfall model, that assumes that requirements are known at the start, is inadequate for this purpose.

Other significant factors were unannounced and announced mode changes that were associated with side effects that the pilot did not expect or had not been trained to handle. The root cause could be traced to missing requirements that caused these mode changes to not be performed in the test program and to be omitted from pilot training. Requirements for handling rare events in highly integrated and complex systems are not explicitly addressed in the guidance documents; and the available design tools, at most, check for consistency among requirements but assume that they are complete. This report, therefore, deals with requirements for handling rare events, such as mode changes, and proposes a structure for the review process that may help eliminate the hazards that led to the observed incidents.

A number of the most severe incidents were due to faulty redundancy management. Therefore, a follow-on effort should be undertaken to generate review guidelines for the certification of redundancy management provisions. There also was lack of guidance for the interfaces of flight control systems with area navigations systems, Traffic Collision and Avoidance System and supervisory functions; this also should be addressed in future efforts.

None of the incidents reported in this document were due to historically important failure modes, such as lack of stability margins and software errors. Improvements in guidance materials and development and analysis tools may contribute to these advances.

1. INTRODUCTION.

1.1 PURPOSE.

Although scheduled carriers show the lowest fatality rates of any form of long-distance transportation, flight incidents still occur that call for improvement in the certification procedures. The purpose of this research was to identify potential areas for improvement, particularly in flight control and related aircraft systems.

1.2 BACKGROUND.

This report was generated by SoHaR Incorporated and issued by the Federal Aviation Administration (FAA) William J. Hughes Technical Center. The effort was structured into the following four tasks:

1. Review in-service safety incidents that may be attributable to deficiencies in the certification process

2. Perform gap analyses to determine the adequacy of guidance documents and recommend improvements where indicated

3. Evaluate the capabilities of current design analysis tools

4. Recommend guidance material for application to the certification process

The major portion of this material was obtained from FAA and National Transportation Safety Board (NTSB) websites (in a few instances, from equivalent websites from other countries) and the specific references are cited where appropriate. It was learned that practically all material that is submitted for certification is returned to the originator, and the material that is retained is regarded as proprietary and could not be made available without consent of the aircraft manufacturer. Attempts to obtain this consent from The Boeing Company were unsuccessful. Other manufacturers were not contacted because access to them was considerably more difficult. While the overall results were not impacted by lack of access to certification material, the findings and recommendations could have been more specific with cooperation from the manufacturers.

In this report, the primary reference for design requirements for aircraft systems, in general, was Title 14 Code of Federal Regulations (CFR) 25.1309 [1] with CFR 25.671 [2] and CFR 25.672 [3], which establish additional requirements for flight control and stability augmentation systems. Detailed references to these documents, as well as to those that interpret or apply these requirements, are discussed in section 4 of this report.

2. REPRESENTATIVE INCIDENTS.

Only incidents after 1994 were considered, with a major emphasis on the last 10 years, i.e., 1999-2009. Also, the focus of this investigation was on flight controls and associated digital systems.

2.1 SOURCES AND SELECTION CRITERIA.

Domestic aircraft accidents (involving deaths, serious injuries, or substantial aircraft damage) and incidents (involving lesser degrees of injuries or damage) are reported in FAA and NTSB databases, which are accessible via Aviation Safety Information Analysis and Sharing (ASIAS). Despite good search capabilities, these sources were of limited usefulness for this study because of U.S. airspace restrictions and because only a few significant aircraft events caused by flight control and avionics malfunctions have occurred in the U.S. in recent years. ASIAS also contains a subset of the World Aircraft Accident Summary, but the criteria are total loss of the aircraft; also, it does not seem to be currently maintained (the last entry at the time of this writing was 2007). ASIAS was, therefore, used primarily for background information and to make sure no domestic incidents were overlooked.

Referrals by FAA personnel who monitored this investigation and were aware of significant international events were a more productive source. *Aviation Week and Space Technology* was also helpful because it frequently identified accident reports issued by the foreign equivalents of the NTSB that could be searched for authoritative findings of the accident investigation. Another source was the Aviation Safety Network Aviation Safety Database [4] that lists worldwide accidents but has limited search capability. These sources are collectively referred to as event sources in the following sections. Because only six incidents were available from these sources, all were selected for analysis.

Airworthiness Directives (AD) were another important data source for aircraft problems caused by or affecting flight controls. ADs are sometimes issued as a result of inspections and tests conducted by aircraft manufacturers and operators, but more often, they are issued in response to operational incidents. Because of the liaison maintained with foreign regulatory agencies, ADs reflect the global experience. A limitation of ADs is the focus on corrective actions, which means that the description of the incidents is very terse or sometimes missing entirely. The corrective action is frequently expressed as implementing a manufacturer's service bulletin or installing a new software version, thus, masking the deficiency that caused the issuance of the AD.

In some cases, the AD identified FAA personnel who were able to supply further information; but more frequently, the individuals listed in the AD were no longer associated with the organization, or the requested information was considered proprietary to a commercial company. Also, documentation submitted in support of certification is routinely returned to the originator once the certification is granted.

The AD was excluded from further consideration if the underlying difficulty could not be determined or if an incident was due to causes outside the normal certification process. An example of the latter category is actions to prevent the use of magnetic deviation data known to be out of date (AD 2005-05-05).

2.2 INCIDENTS FROM EVENT SOURCES.

Incidents obtained from event sources are shown in table 1 in chronological order. The first two entries in the table are based on preliminary findings; the investigations into these incidents are

still in progress as of this writing. Column headings are self-explanatory except for the last column, which reflects underlying causes that are explained in sections 2.2.1 through 2.2.6. The abbreviation RM in this column means redundancy management. Where "Maint" is listed, it indicates that required maintenance was omitted or incorrectly performed, but this may be caused by failure to communicate the safety-critical nature of a maintenance action, an issue that could be addressed in the certification procedure. Further discussion of causes can be found in section 3.

Table 1. Incidents From Event Sources

Description	Date	Aircraft	Location	Source	Cause Code
Inadvertent throttle retardation	2/2009	B-737	Amsterdam, Netherlands	Onderzoeksraad voor Veiligheid 2009	Mode Maint RM
Northrup Grumman Corporation ADIRU*	10/2008	A330	Learmouth, Australia	ATSB AO-2008-070	RM Mode Monit
Honeywell ADIRU	8/2005	B-777	Perth, Australia	ATSB AO-2005-03722	RM Maint
Alpha-prot—hard landing	5/2001	A320	Europe	AvWeek May 25, 2001	Mode
Alpha-prot—near collision	10/2000	A340	West of Scotland	AIRPROX	Mode
Pilot-induced oscillations	9/1999	Falcon 900	Bucharest, Romania	Romanian Ministry Transport Nr. 711/Jan 2000	Mode Maint

Note: Alpha-prot refers to angle-of-attack protection.
*A similar incident also occurred in December 2008.

ADIRU = Air data inertial reference unit
ATSB = Australian Transportation Safety Board
Maint = Maintenance
Monit = Monitoring

As shown in the last column, many incidents are due to more than one cause.

In the following sections, the text in italics is taken from the official accident report.

2.2.1 Inadvertent Throttle Retardation.

An aircraft crashed in a field on approach to Schiphol Airport with 9 fatalities and 86 injuries [5].

> *As far as known the flight proceeded uneventfully up until entering Dutch airspace. The aircraft was directed by Air Traffic Control towards runway 18R for an ILS approach and landing. The crew performed the approach with one of the two autopilots (autopilot B) and autothrottle engaged. The aircraft descended to 2000 feet above mean sea level and was vectored towards the localizer. The landing gear came down and flaps 15 were set.*

At this point, the aircraft was above the glide slope and had to descend to meet it.

> *At approximately 1950 feet the recorded value* [of the left radio altimeter] *suddenly changed to -8 feet and remained at that value up until shortly before impact.*

The recorded value of the left radio altimeter had previously been 8191 feet. When the altimeter output changed, the cockpit voice recorder reported several aural warnings (check landing gear and flaps).

> *The warnings sounded because the computer systems receive their data from the left radio altimeter, amongst others, which erroneously transmitted that the aircraft was near the ground…. The cause of the aural warnings and the reaction of the crew to these warnings are still being investigated.*
>
> *The values of the right radio altimeter and pressure altimeter were correct during approach.*
>
> *At approximately 770 feet, the crew set the selected airspeed to 144 knots. At that moment the actual airspeed was 144 knots. The autothrottle system should have maintained the speed selected by the crew but, with the thrust levers at idle and the autothrottle system still in the retard mode, speed continued to decay. Because the automatic pilot wanted to maintain the glide scope, the automatic flight system, in response, commanded increasing nose up pitch and applied nose up stabilizer trim.*
>
> *The stick shakers activated at approximately 460 feet, warning the crew that the angle of attack (AOA) was too high. The data of the digital flight data recorder show that the thrust levers were immediately advanced but moved back to idle. When the thrust levers returned to idle, the autothrottle was disengaged. Whether these actions were performed by the crew or automatically is still under investigation.*
>
> *At 420 feet the autopilot was disengaged by the crew and attempts were made to recover. At 310 feet the pitch attitude had reached 8° nose down. Almost simultaneously the thrust levers were advanced to their most forward position after which the aircraft ascended somewhat and the pitch angle increased*

Shortly after, the aircraft crashed in a nose-up attitude in a field about 1 mile short of the runway threshold.

The flight data recorder retained information from prior flights.

> [These] *show instances of left radio altimeter malfunctions on some of the nine previous flights. In the recorded cases, the autothrottle also entered the retard mode above the intended flare altitude, and the thrust levers moved to idle, because of a malfunction of the left radio altimeter on two of the nine flights. The data of these flights are being investigated.*

… Dutch Safety Board has issued a warning to Boeing in which extra attention is asked for a part of one of the manuals (737 Dispatch Deviations Guide) of the Boeing 737. In this guide is stated that if, preceding flight, the radio altimeters are malfunctioning, the associated automatic pilots and autothrottle systems cannot be used for approach and landing. The Board has given Boeing into consideration to investigate if these procedures should also be valid during all phases of a flight.

Boeing has issued a Multi Operator Message' (MOM) the same day concerning malfunction of the radio altimeters.

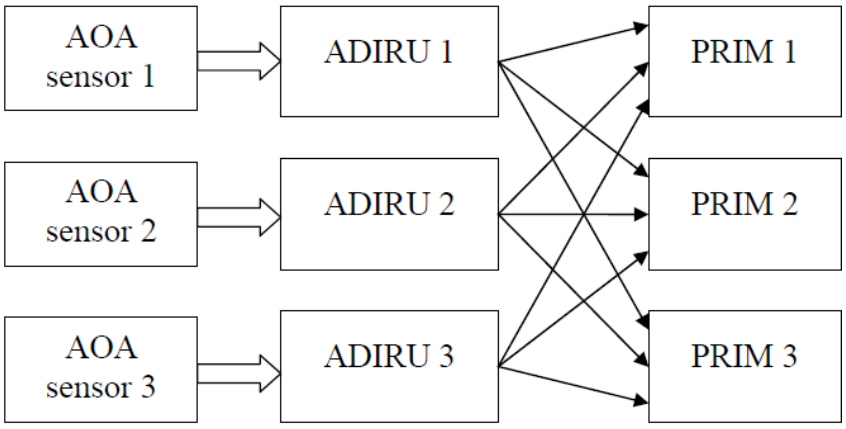
The Mode cause code was assigned because of the apparently insufficient autothrottle status information being displayed to the crew. The Maint cause code was due to a lack of follow-up on the previous altimeter failures. While there was no failure of RM, the RM cause code was assigned because there were no automatic provisions to use the redundant radio altimeter information.

2.2.2 Northrop Grumman Corporation Air Data Inertia Reference Unit.

On October 7, 2008, an Airbus 330-303 (operated by Qantas) on a flight from Singapore to Perth, Western Australia, experienced violent pitch-up and pitch-down maneuvers that left 12 people seriously injured and over 100 less seriously [6]. The cause was traced to faulty data from one of the Air Data Inertial Reference Units (ADIRU).

The Qantas A330 was equipped with three ADIRUs manufactured by Litton Industries, Inc., now a part of Northrop Grumman Corporation (NGC). The data flow from the ADIRUs to the three primary flight control computers (PRIM) is shown in figure 1.

Each AOA sensor utilised two identical outputs (A and B for each sensor) for redundancy. The relevant ADIRU checked the A and B signals to ensure that they agreed. If they agreed, the data was passed on to other systems.



AOA = Angle of attack

Figure 1. The ADIRU to Autopilot Data Flow

At 1240:28 the autopilot disengaged…The captain took manual control of the aircraft using the sidestick. The crew received aural stall warning indications at this time, and the airspeed and altitude indications on the captain's primary flight display (PFD) were fluctuating. At 1242:27 the aircraft abruptly pitched nose-down. The captain reported that he applied backpressure on his sidestick to arrest the movement. The crew reported that the messages on the Electronic Centralized Aircraft Monitor (ECAM) were constantly scrolling and they could not effectively interact with the ECAM to action and/or clear the messages.

There were further upsets, but through the use of standby instruments, the crew was able to land the aircraft safely at Learmouth, Australia.

Each ADIRU consists of an air data part and an inertial reference (IR) unit part. The air data and IR unit parts can be switched off separately. In addition, there are rotary switches that permit selection of the ADIRU feeding the autopilot. The multiplicity of switches is intended to permit the aircraft to fly with a failed air data part on one ADIRU and a failed IR unit part on another; but in this case, it may have affected the crew's ability to respond to the upsets. An analysis of this issue is presented in section 2.2.3.

The postflight examination showed that the ADIRU 1 transmitted numerous faulty air data spikes (AoA and airspeed). These spikes should have been filtered by the autopilots (they receive multiple ADIRU inputs, see figure 1), but the filtering algorithm was not robust.

The aircraft manufacturer advised that the AOA [angle-of-attack] *processing algorithms would prevent most types of erroneous AOA inputs provided by the ADIRUs having an influence on flight control ommands. This included situations such as an AOA 'runaway' (or a continuous divergence from the correct value), single AOA spikes and most situations where there were multiple AOA spikes. However, the manufacturer identified that, in a very specific situation, the PRIMs could generate an undesired nose-down elevator command. This specific situation involved multiple AOA data spikes with the following properties:*

- *there were at least two short duration, high amplitude spikes*
- *the first spike was shorter than 1 second*
- *the second spike occurred and was still present 1.2 seconds after the detection of the first spike.*

Recorded flight data from the accident flight showed that there were 42 recorded spikes in AOA 1 data. Due to recorder sampling rate limitations, it is likely that there were additional AOA 1 spikes that were not evident in the recorded data and it is not possible to reconstruct the exact duration and timing of any of the spikes.

Although a large number of AOA 1 spikes occurred on the accident flight, on all but two of those occasions, the processing algorithm filtered them out and they had no influence on the flight controls. The aircraft manufacturer advised that

AOA spikes may occur on many flights, but in its experience, there were usually only a very small number of spikes on any particular flight. It was not aware of any previous event where AOA spikes had met the above conditions and resulted in an in-flight upset.

The report explains the magnitude of the pitch down command as follows:

Two of the flight envelope mechanisms were influenced by the AOA spikes during the accident flight: high angle of attack protection (alpha prot) and anti pitch-up compensation.

Alpha prot was designed to protect the aircraft from high AOAs which could lead to a stall and loss of control. If the PRIMs detected that the aircraft's AOA exceeded a predefined threshold, the computers would command a nose-down elevator movement to reduce the AOA. Alpha prot was only available when the aircraft was in normal law. When the aircraft was above 500 ft above ground level, alpha prot was effective immediately, while below 500 ft it was only active after the AOA exceeded the threshold for 2 seconds or more.

Anti pitch-up was a pre-command included in the control laws to compensate for a pitch-up at high Mach due to aerodynamic effect. The compensation was available above Mach 0.65 and when the aircraft was in a 'clean' configuration (that is, with the landing gear and flaps retracted). The maximum authority of the anti pitch-up compensation was 6 degrees of elevator movement. The aircraft manufacturer advised that the 10-degree elevator command associated with the first in-flight upset, was the result of 4 degrees of alpha prot and the 6 degree authority of the anti pitch-up compensation. The 10-degree command was close to the worst possible scenario that could arise from the design limitation in the AOA processing algorithm.

In response to the Australian Transportation Safety Board (ATSB) findings, Airbus issued an Operations Engineering Bulletin (OEB) A330-74-1, on October 15, 2008, applicable to all A330 aircraft fitted with Northrop Grumman ADIRUs.

The OEB stated that, in the event of a NAV IR FAULT (or an ATT red flag being displayed on either the captain's or first officer's PFD), the required procedure was for the crew to select OFF the relevant ADR and then select OFF the relevant IR. The OEB procedure was subsequently amended in December 2008 to cater for a situation where the IR and ADR pushbuttons are selected to OFF and the OFF lights did not illuminate. If the lights did not illuminate, the new OEB (74-3) required crews to select the IR rotary mode selector to the OFF position.

A similar air data spike situation was encountered by another Qantas A330 on December 27, 2008. By switching the air data sources in accordance with the OEB, the crew was able to avoid the extreme maneuvers experienced in the October 2008 event. It was also found that air data

spikes were recorded on other aircraft: a Qantas flight on September 12, 2006, and another airlines flight on February 7, 2008. Neither of these progressed to violent maneuvers.

The RM cause code was assigned because of the lack of robustness in the PRIM fault isolation algorithm. The Mode cause code was assigned because of the multiplicity of ADIRU operating modes, which can be confusing, as recognized in the Airbus engineering bulletin. The Monit cause code reflects the scrolling ECAM display that hampered the crew's ability to deal with the malfunction. As of this writing, the ATSB had not yet issued a recommendation regarding the simultaneous action of the alpha-prot and pitch-up prevention features.

2.2.3 Honeywell ADIRU.

The flight from Perth to Kuala Lumpur had reached 37,000 ft when it experienced a violent uncommanded, pitch-up maneuver [7]. The no. 5 accelerometer in the ADIRU failed, but such a single failure could be tolerated by the ADIRU, which was specifically designed to require minimum maintenance. The failure event was recorded in a register of the ADIRU but was not visible to the pilot.

> *Maintenance Message (MM) 34-20010 is a latched fault and indicates an internal failure in the ADIRU that does NOT result in a status message* [visible to the pilot]. *The MM indicates the first failure within a fault containment module (FCM), for example a gyro or processor failure, in the ADIRU. The second failure within a FCM will result in an ADIRU Status Message and MM 34-20000.*

The redundancy here is implemented in a single ADIRU, as shown in figure 2. Fault containment areas (FCA) are functions with internal redundancy.

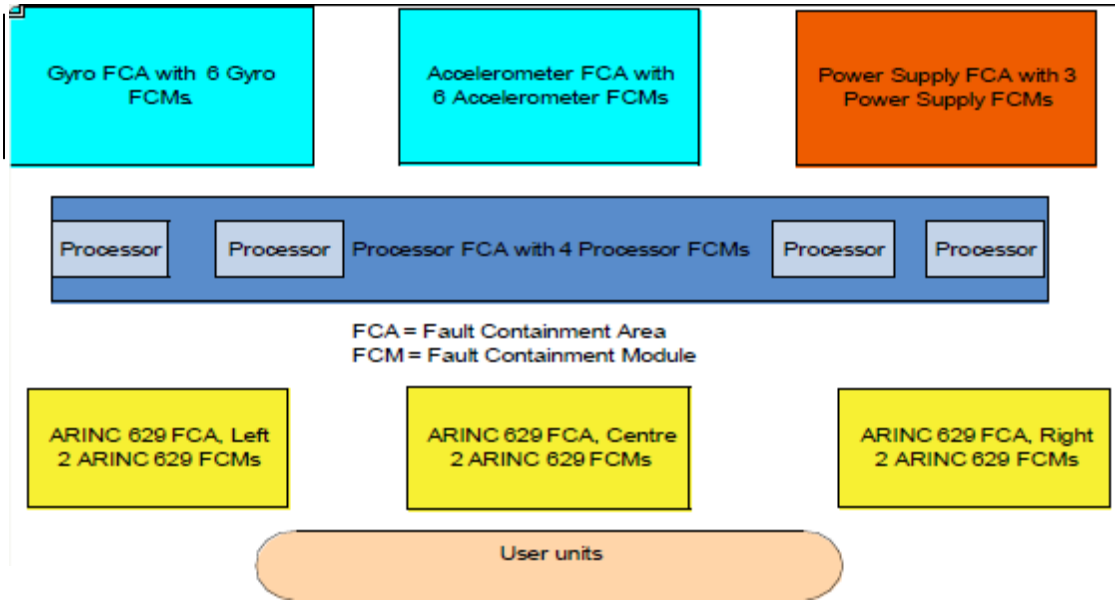


Figure 2. Redundancy in Honeywell ADIRU

The ADIRU on the 777 airplane is a fault tolerant unit. Therefore, operating with MM 34-20010 only means that an "extra" FCM (used for deferred maintenance) has been lost. ADIRU's with MM 34-20010 have sufficient resources to meet the performance requirements of the ADIRU. Also, when the ADIRU Status message is displayed, although redundancy has been lost the ADIRU continues to output its voted solutions for Air Data and Inertial parameters. There is 777 MMEL dispatch relief to operate with an ADIRU Status message for 3 days.

During the flight event in August 2005, the no. 6 accelerometer malfunctioned, and a software error allowed the output of the previously failed no. 5 accelerometer to be substituted. This caused a rapid climb of the aircraft with loss of airspeed and incipient stall. The pilots disconnected the autopilot and were able to return the aircraft to Perth. The passengers were not injured.

The accelerometer FCA is comprised of six accelerometers in a nonorthogonal orientation. Theoretically, this should permit it to compute the correct acceleration data as long as three accelerometers remain operational. The computation of the valid inertial orientation after failure of an instrument in a nonorthogonal array has been recognized as a challenging software problem for several decades. It was selected as the sample problem when, as part of a fault-tolerant software project in the late 1980s [8], National Aeronautics and Space Administration (NASA) wanted to investigate whether failure modes of independently developed software were, in fact, independent. A well-vetted specification was given to five teams (2 graduate students each) at each of five leading software engineering schools at that time. The 20 resulting versions were then subjected to over 800,000 tests. Table 2, based on table 1 of reference 5, shows excerpts for the test cases in which a new failure was introduced. Thus, the entries in the first row represent a first failure, those in the second row, a second failure, etc.

Table 2. Nonorthogonal Instrument Tests

No. of Prior Anomalies	Observed Failures	Total Tests	Failure Fraction
0	1,268	134,135	0.01
1	12,921	101,151	0.13
2	83,022	143,509	0.58

These data showed more than a tenfold increase in overall failure probability after a first instrument failure, which was hardly a basis for permitting a nonorthogonal array with one failed accelerometer to be flown for 4 years without repair. Thus, the Maint cause code was assigned. The RM cause code was assigned because the explicit requirement for the ADIRU was that it would continue operating after a second failure.

2.2.4 Alpha-Prot—Hard Landing.

On February 7, 2001, Iberia Flight 1456, an Airbus 320, made a hard landing, in nose-down attitude, despite the pilot-in-command's decision to go around and the application of maximum power [9]. During a nighttime instrument approach at Bilbao, Spain,

the aircraft encountered heavy turbulence at about 200 feet agl. with gusts up to 65 mph. The aircraft encountered windshear with 1.25G updraft, downdraft and a tailwind gust at just 70 feet agl. When the Ground Proximity Warning System (GPWS) sounded, the captain called for a go-around while pulling on the sidestick, reportedly without pressing his priority control button. The combination of dynamic winds and the crew actions created a situation that triggered the airplane's alpha protection system. As the crew applied TOGA power for a go-around, with both pilots pulling back on their sidesticks, the alpha protection law reduced the elevator nose-up command. Instead of a go-around, the aircraft struck the runway with a vertical speed of approx. 1,200 fpm. The nosegear collapsed and the aircraft skidded 3,280 feet (about 1000m) down the runway before coming to a stop.

This incident prompted Airbus to develop a modification to its flight control software. It will prevent the airplane's built-in protection against stall from being activated by a high rate of change in angle of attack. As an interim action, an AD was issued requiring A.320/A.319 operators to fly at least 10 knots faster and to use only a setting of "CONFIG 3" during approach with gusts higher than 10 knots or when moderate to severe turbulence is expected on short final.

In some other incidents, crews wanted to land in a nose-high attitude (e.g., because of suspected nose wheel problems) but it was prevented by the alpha-prot feature, and the aircraft sustained preventable damage. The Mode cause code was assigned because of unsuitable control limitations imposed by the alpha-prot function.

2.2.5 Alpha-Prot—Near Collision.

On October 2, 2000, a Canadian A330 and a Turkish A340 were flying westbound over the Atlantic Ocean on the same track with 1000-ft vertical separation under reduced vertical separation minimum procedures, and experienced a near collision [10]. The A340, being at the lower level,

was expecting a turbulence encounter around 59°N 20°W and when the aircraft first entered light turbulence he [the commander of the 340] *made a cabin announcement and switched on the seat belt signs. Shortly before the AIRPROX event he experienced moderate turbulence and noticed outside air temperature changes. Suddenly the aircraft began to climb, the Master Warning sounded and the autopilot self-disengaged as the aircraft exceeded the speed limit of 0.86 Mach. The indicated airspeed dropped below V_{LS} (the lowest selectable) as the aircraft climbed and the commander took manual control of the aircraft because neither autopilot would engage. The crew subsequently reported the incident to Shanwick on HF radio and using their TCAS, they descended back to FL 360 in a safe area. At the time of the AIRPROX the commander estimated the aircraft were one mile apart laterally.*

Changes to the A340's flightpath caused by the aircraft's flight control system response to the overspeed warning and autopilot disconnect were negligible until AoA law was triggered. The fact that this law was not triggered until 10 seconds after the autopilot disconnected was a random event driven by the severity of the turbulence. Had the turbulence been more severe at the first encounter and coincident with the overspeed warning, reversion to AoA law could have been triggered as soon as the overspeed condition disconnected the autopilot. Nevertheless, it should be noted that had the autopilot remained engaged, the AoA law would not have been invoked because it is inactive except in manual control.

Once AoA law is active, rearward movement of the sidestick controls angle of attack between alpha prot (neutral sidestick) and alpha max (full aft sidestick). Forward movement of the sidestick disengages AoA protection law and the system reverts to normal pitch law. However, there is no aural or text message which informs a crew that AoA protection law has been invoked. If the sidestick is not moved from its neutral position, the pitch flight control system is programmed to capture alpha prot and not the airspeed that corresponds to alpha prot in 1g flight. Consequently, in turbulence the speed scale will probably be oscillating, the aircraft pitch angle could also be oscillating, and the change from normal pitch law to AoA protection law could be difficult to detect.

Further on, the report continues:

The commander's reported sighting of an 'Alpha Lock' message was probably an alpha floor warning on the flight mode annunciator portion of the PFDs [Primary Flight Display]. *Alpha floor is an autothrottle function which applies full thrust, irrespective of the position of the thrust levers, if the airspeed is likely to reduce to a value approaching alpha max. In this incident, the A340's calibrated airspeed decreased from around 270 kt before the turbulence encounter to 205 kt at the apogee of the climb.*

As a result of this incident, the major safety recommendation by the Aircraft Accident Investigations Board is to encourage pilots to provide lateral offsets from the assigned track when overtaking an aircraft at a lower flight level. However, the unintended consequences of the alpha-prot and the crew's unfamiliarity with them were obviously a significant factor in this near catastrophe and are the basis for the following comments.

When an AoA spike due to turbulence caused the AoA protection (alpha-prot) mode to be engaged the aircraft pitched up to capture and hold that alpha-prot value (about 4.5° for a cruise condition), which is 3° to 4° above the trim AoA, causing a 0.3- to 0.4-g climb initiation. At that point thrust was at idle, therefore, the speed started to drop quickly. The actual AoA response may have overshot the alpha-prot value and reached alpha-floor mode, triggering the

advancement of the throttles to takeoff power, while the speed dropped as low as 205 kt. These multiple, and at least partly unannounced, changes in control modes of the aircraft raise the following issues for the certification authorities.

- Were the certification authorities fully informed on the detailed design and behavior of the alpha-prot and alpha-floor functions of the Airbus aircraft in a variety of operating conditions?

- Did the certification authorities accept those modes based on informed consent after a reasonable amount of investigation all of the operational safety aspects under normal and abnormal conditions, including adverse atmospheric conditions (turbulence and clear air turbulence associated with jet stream phenomena, including airspeed altitude and air temperature fluctuations) and failures?

- Did the certification authorities know and approve the latching feature of the alpha-prot and alpha-floor modes?

- Did the certification authorities know that a momentary AoA spike could cause the alpha-floor mode to engage, causing an inadvertent pitch-up and climb to be initiated?

- Did the certification authorities know about and approve the alpha-floor design feature that requires a nose-down stick action to disengage the alpha-prot mode?

- Did the certification authorities know and approve the alpha-floor mode behavior in which alpha-prot engagement can be triggered when flying through unstable atmospheric conditions and subsequent AoA-response overshoot can trigger alpha-floor engagement?

If the answer to any of these questions is no, then this may be evidence that the certification process probably did not adequately explore all possible behaviors and the safety consequences of this relatively new technology and its specific design implementation.

2.2.6 Pilot-Induced Oscillations.

In September 1999, a Falcon 900 operated by Olympic Airways for the Greek government experienced severe pitch oscillations during the final stages of a flight from Athens to Bucharest, Romania [11]. Shortly after takeoff, when flaps and slats were retracted, a pitch feel warning light illuminated and remained on for most of the flight. On descent into Bucharest the autopilot disengaged, apparently without deliberate crew action, probably as a result of the pilot pulling on the control column.

From this point, there were ten major oscillations, which lasted for about 2 seconds and caused vertical accelerations of almost +5 and -3 g's, as shown in figure 3.

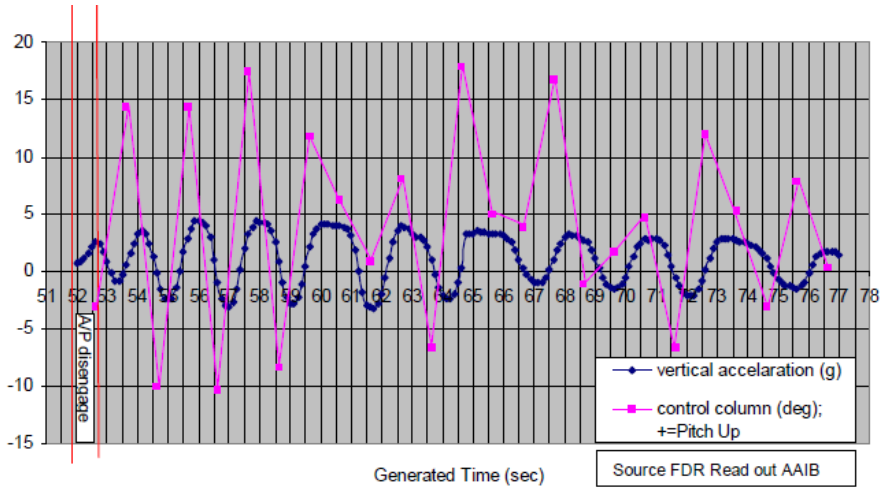


Figure 3. Column Movement and Vertical Acceleration

Although the aircraft was nearing its destination, most of the passengers did not have their seatbelts fastened and were violently thrown about during the oscillations. Seven passengers were killed, the flight attendant and one passenger were seriously injured, and two passengers and the cockpit crew sustained minor injuries. The oscillations subsided when the airspeed was reduced, and the aircraft was able to land normally.

The official investigation by the Romanian Ministry of Transport identified the following causal factors.

1. Inadequate risk assessments of the pitch feel malfunctions

2. The crew overriding the autopilot on the pitch channel

3. Inappropriate inputs on the control column at high speed and artificial feel unit failure in low-speed mode leading to pilot-induced oscillations

4. Seatbelts not fastened during descent flight phase

With regard to causal factor 1, the investigation established that the pitch feel warning light had illuminated during previous flights and maintenance actions failed to rectify the malfunction. During the accident flight, the pilot could not identify the cause of the warning light coming on and did not consider it a real malfunction. Yet it was concluded that

> *the continuous oscillations due to pilot inputs would have been probably lessened with an operative Arthur Q* [Pitch Feel] *unit.*

With regard to causal factor 2, the analysis for the time just prior to disengagement of the autopilot concludes

> *Since the applied command voltage would move the elevator control trailing edge down, whereas it is moving up, there appears to be an external force of FGC* [flight guidance computer] *moving the elevator controls in the opposite direction to that commanded by the autopilot.*

A possible motivation for overpowering the autopilot could be that, exactly at that time, the crew received clearance to descend beyond the previously authorized altitude of 15,000 to 5,000 feet. If the new altitude was set into the flight director panel while 15,000 feet was being approached, the autopilot will revert to basic pitch mode without the pilots being aware of this.

> *If the selected altitude was changed while the aircraft was in the SEL ALT CAP mode, the A/P [will] revert to Pitch Hold mode. Available data are not adequate to determine for sure whether the mode change occurred this way.*

This finding and lack of indication of autopilot disengagement are the reasons for assigning the Mode cause code. The Maint cause code was assigned due to a lack of follow-up on earlier indications of the artificial feel unit.

2.3 INCIDENTS INFERRED FROM AIRWORTHINESS DIRECTIVES.

An equivalent summary of incidents inferred from the specified ADs is shown in table 3. The AD data do not associate the incident with a location and, therefore, that column was omitted.

Table 3. Incidents From Airworthiness Directives

Description	Date	Aircraft	AD No.	Cause Code
Loss of instruments	2/2009	A320 family	2009-01-04	RM (hardware) interface with electric power
Blanking of instruments	8/2008	B-747	2008-13-22	RM
Integrated Standby Instrument System reset	12/2004	A330/340 A320 family	2004-25-07 2004-25-08 2007-13-07	Interface with electric power Design
Unexpected autopilot disconnect	12/2004	B-747	2004-25-06	Mode Design
Loss of liquid crystal display instruments	10/2004	A320 family A330/340	2004-20-05 2004-20-06	RM
Nuisance alarms	6/2004	B-747	2004-10-05	RM
Glide slope oscillations*	7/2003	B-727	2003-11-19	Design
Pitch trim failure	6/2000	Embraer 145	2000-09-09	Monit
Autopilot disengage problem	4/2000	Allied Signal/ Honeywell autopilot	2000-05-24	Hardware design
Wind shear disengage not operating	12/1999	Enbraer 135/145	99-24-13	Mode
Miswired actuators	7/1999	A300 and A310	99-16-14	Hardware design

*This incident occurred at Chicago O'Hare International Airport, and an NTSB report was available.

The multiple-cause hypothesis is probably true, as shown in table 3 where, due to the lack of detailed information, only a single cause is listed in many rows. Failure of an electronic, electromechanical, or hydraulic component may have initiated an incident, but these components are known to be subject to random failures that are accounted for in the certification basis. Several hardware items are included in table 3 to permit a discussion of the difference in the approach to certification between hardware and software. Details on each of the AD incidents are presented in sections 2.3.1 through 2.3.11.

2.3.1 Loss of Instruments.

> *Some operators have reported occurrences of loss of the AC BUS 1 with subsequent loss of the AC ESS BUS and DC ESS BUS, resulting in the loss of 5 upper Display Units and the loss of integral lighting. In this situation, flight crew[s] have reported concerns in reading the standby instruments when the DOME lights were selected to OFF. This situation, if not corrected, could increase the workload of the flight crew.*

Loss of a bus is an RM issue that can cause outages of both the primary displays and the lighting for the standby instruments. This can cause an electric power interface issue when all five primary displays are powered by the same bus.

2.3.2 Blanking of Instruments.

> *This AD results from two instances where all six integrated display units (IDUs) on the flight deck panels went blank in flight. We are issuing this AD to prevent loss of the IDUs due to failure of all three electronic flight instrument system/engine indicating and crew alerting system (EFIS/EICAS) interface units (EIUs), which could result in the inability of the flightcrew to maintain safe flight and landing of the airplane.*

The AD also refers to AD 2004-10-05 for a related condition on air data computers (ADC) (see section 2.2.6).

Under some conditions, a sensor disagreement could cause all three EIUs to shut down. Boeing issued a Flight Crew Operations Maintenance Bulletin that recommended a restart procedure. However, the AD required installation of at least one EIU with improved RM logic within 24 months.

2.3.3 Integrated Secondary Instrumentation System Reset.

> *This AD requires regularly performing a complete electrical shutdown of the airplane to reset the integrated standby instrument system (ISIS). This AD also provides an optional terminating action. This AD is prompted by reports indicating that an airplane lost the ISIS, then, during the same flight, lost all electronic instrument system (EIS) display units. We are issuing this AD to prevent loss of the ISIS, which, if combined with loss of all EIS display units,*

could reduce the flightcrew's situational awareness and contribute to loss of control of the airplane or impact with obstacles or terrain.

The shutdown was caused by a timer that timed out when power had been applied continuously for about 1000 hours. In initial operations, the aircraft had undergone a complete power shutdown before this timer reached the cutoff point. Some operators found it more efficient to keep the power on continuously, which caused this problem, i.e., AD 2007-13-07 introduces a software modification that reduces the need for shutdowns (the terminating action). This is also an example of assumptions about aircraft operation that may not stand the test of time, hence, the Design cause code.

2.3.4 Unexpected Autopilot Disconnect.

This AD requires revising the airplane flight manual to prohibit operation of the autopilot/flight director in command mode with performance management system selected on the speed mode switch during cruise in reduced vertical separation minimum (RVSM) airspace.

[The FAA has] *received reports of two separate incidents in which a Boeing Model 747-200 airplane equipped with a performance management system (PMS) had an unexpected autopilot disconnect induced by the passing of another airplane within 1,000 feet below the airplane while operating in reduced vertical separation minimum (RVSM) airspace. In both incidents, the PMS-equipped airplane lost 300 to 400 feet of altitude, causing it to come within approximately 650 feet of the other, lower aircraft (starting at 1,000 feet separation), and received a traffic collision and avoidance system (TCAS) resolution advisory (RA) with instructions to "climb, climb."*

The PMS installed in certain B-747 aircraft has an interlock that is activated with radar altitude. This interlock disconnects the autopilot upon receipt of a valid radar altitude signal of less than 2500 feet. Because there is no means to accurately determine how the aircraft is trimmed when using the PMS, it cannot be predicted which direction the aircraft will fly or how far it will depart from an assigned altitude once the autopilot is disconnected. The disconnect logic is probably unfamiliar to the crew, and the disconnect event, although visible on the display, may not be recognized. This is an example of potentially dangerous complexity in the operating mode design. It is also an example of how assumptions about aircraft operations are not always valid (e.g., low radar altitude = approach to landing); hence, the Design cause code.

2.3.5 Loss of Liquid Crystal Display Instruments.

The Direction Generale de l'Aviation Civile (DGAC), which is the airworthiness authority for France, notified the FAA that an unsafe condition may exist on certain Airbus Model A318, A319, A320, and A321 series airplanes. The DGAC advises that there have been several reports of total loss of all six liquid crystal display (LCD) units for the electronic instrument system (EIS) of a certain EIS2 standard during cruise for a short period of time. The flightcrew used the standby instruments, and the LCD units were eventually recovered. Subsequent

investigation revealed that the three display management computers had received erroneous data from one LCD unit.

Erroneous data from one LCD unit causing shutdown of three display management computers is an RM problem. This is a rare occurrence, which indicates that it is in response to very specific erroneous data or under specific conditions. Thus, a requirements problem may have caused these conditions to be missed during test.

2.3.6 Nuisance Alarms.

[The action is] *proposed to require a modification of the air data computer (ADC) system, which involves installing certain new circuit breakers, relays, and related components, and making various wiring changes in and between the flight deck and main equipment center." "These changes are intended to allow the flightcrew to silence an erroneous aural overspeed or stall warning by switching away from a failed ADC that is generating the warning.*

The failure in a single ADC (out of the three that are installed), which can generate serious warnings (that can cause crew distraction), indicates there is an RM problem.

2.3.7 Glide Slope Oscillations.

[The AD] *requires, under certain conditions, replacement of the installed autopilot pitch control computer with a modified computer, testing of the modified system, and revision of the Airplane Flight Manual (AFM).*

The design assumptions for the glide slope coupling were that the approach would be flown at an airspeed of 110 knots with flaps at 40 degrees. However, the aircraft was difficult to maneuver in that configuration and many operators were switching to 30-degree flaps, which necessitated an increase in airspeed to about 135 knots. The desensitization for glide slope narrowing was time based and, thus, too slow for the higher approach speed. The autopilot manufacturer had developed modification kits, but these were not installed in the aircraft that landed short of the runway at Chicago O'Hare International Airport.

The investigation revealed that the accident airplane's autopilot was functioning within its design tolerances; however, the autopilot's 150-second desensitization rate was too slow for the accident airplane's approach speed, resulting in divergent pitch deviations at a low altitude at a critical time during the approach.

The AD made the modification mandatory. The accident was also due to visual disorientation of the captain, who kept his sunglasses on during a low-visibility approach under difficult conditions.

2.3.8 Pitch Trim Failure.

Use of the autopilot below 1,500 feet above ground level, emergency procedures for pitch trim runaway, and abnormal procedures for autopilot trim failure and

> *stabilizer out of trim…This amendment requires replacement of a certain integrated computer with a new integrated computer; installation of an upgraded integrated computers checklist; and removal of certain placards and certain limitations in the AFM…The actions specified by this AD are intended to prevent failure of the pitch trim system, which could cause undetected autopilot trim runaway, and consequent reduced controllability of the airplane, uncommanded autopilot disconnect, and excessive altitude loss.*

Failure of the pitch trim system is probably included in the basis for certification. The notion that it can cause undetected pitch trim runaway and uncommanded autopilot disconnect indicates a problem providing adequate monitoring.

2.3.9 Autopilot Disengage Problem.

> [This AD] *applies to all aircraft equipped with a certain Honeywell International Inc. (Honeywell) KAP 140 or KFC 225 autopilot system. AlliedSignal Avionics Inc. manufactured these autopilot systems before transferring the design data to Honeywell. This AD requires that you inspect the autopilot servo actuator for a loose fastener and modify the autopilot servo actuator when a loose fastener is found. This AD is the result of a report of failure of the autopilot servo actuator to disengage when the autopilot power was removed. The actions specified by this AD are intended to detect and correct a loose fastener in the autopilot servo actuator, which could cause the autopilot servo actuator to not disengage when power to the autopilot is removed.*

This AD specifies that a mechanical condition that can lead to a potential failure event is handled by requiring periodic inspection, although curative measures are probably available, such as changing to a less easily removed fastener.

2.3.10 Wind Shear Disengage Nonoperative.

> *The* [Brazilian airworthiness agency] *advised that tests indicated that, when the autopilot system is coupled to the co-pilot's flight director (flight director #2), the autopilot system does not automatically disengage when a windshear is detected by the ground proximity warning system at a height below 1,500 feet above ground level (AGL). The cause of this malfunction has been attributed to a software discrepancy in the Autoflight IC-600 integrated avionics computer, which causes the autopilot to remain engaged in windshear mode… The manufacturer has advised that it currently is developing a modification that will positively address the unsafe condition addressed by this AD.*

The autopilot is intended to be disengaged when wind shear is encountered below 1500 feet above ground level. This function is correctly accomplished when the autopilot is coupled to the #1 flight director (pilot), but not when coupled to the #2 flight director (copilot). The incident could have been avoided by requiring tests in all permissible operating modes.

2.3.11 Miswired Actuators.

> *One operator of an Airbus Model A300-600 reported high rudder forces and uncommanded rudder inputs during final approach. The uncommanded rudder inputs caused deflections of the rudder control surface resulting in yawing of the airplane. Investigation of the incident is ongoing, but preliminary results indicate that failure of both the main valve and the clutch valve of the autopilot yaw actuator can lead to the actuator generating uncommanded rudder deflections. The DGAC advises that the same autopilot actuator is used for roll and pitch control during autopilot operation, and this failure scenario can result in uncommanded deflections of the aileron and elevator control surfaces.*
>
> *Preliminary results of the investigation of the incident airplane's autopilot yaw actuator indicate that the electrical connectors between the actuator's two main valves and the airplane's two flight control computers (FCC) were crossed between side 1 and side 2. This hidden failure in combination with a failure of the clutch valve resulted in the autopilot yaw actuator remaining engaged when the crew disconnected the autopilot, allowing the actuator to remain hydraulically pressurized and provide inputs to the rudder and the rudder pedals.*

Miswiring that is undetectable until a failure occurs is a highly undesirable condition. It is a direct violation of the requirement in 14 CFR 25.671 (b) [2]

> *Each element of each flight control system must be designed, or distinctively and permanently marked, to minimize the probability of incorrect assembly that could result in the malfunctioning of the system.*

Many techniques are available for preventing accidental crossover, such as keyed connectors or different cable lengths. Alternatively, tests should be performed after any maintenance to prevent this condition. The latter action is now required by this AD.

3. CAUSE ANALYSIS.

3.1 REDUNDANCY MANAGEMENT AND OPERATING MODES.

The most frequent cause codes in tables 1 and 3 are RM and Mode. Malfunctions attributable to these two causes could be avoided with greater emphasis on simplicity and transparency. Both RM and Mode are listed as causes for the NGC ADIRU failure in table 1. An understanding of this malfunction illustrates issues that must be faced in future certification.

As the nomenclature ADIRU indicates, it is a multifunction unit combining air data and IR computation. Both elements are essential for guiding the aircraft along the desired flight path, and both are used for display and automatic control. Because of this essential contribution of the ADIRU output to flight safety, there is a requirement for redundancy; and in the case of the A330 aircraft, aircraft redundancy was tripled. Also, because it contains precision inertial instruments, the ADIRU is an expensive component; therefore, it is desirable to minimize the number of stocked spares. To avoid having to immediately replace an ADIRU after a failure and

to reduce the need to stock spares in multiple locations, the system designers employed partitioned redundancy.

An example of partitioned redundancy technique is shown in figure 4. The objective of redundancy is, in both cases, to maintain at least one functional path after a random failure in a component. The reduction in failure probability against this criterion is shown in figure 5. The following discussion addresses partitioned redundancy, in general, without reference to the ADIRU design.

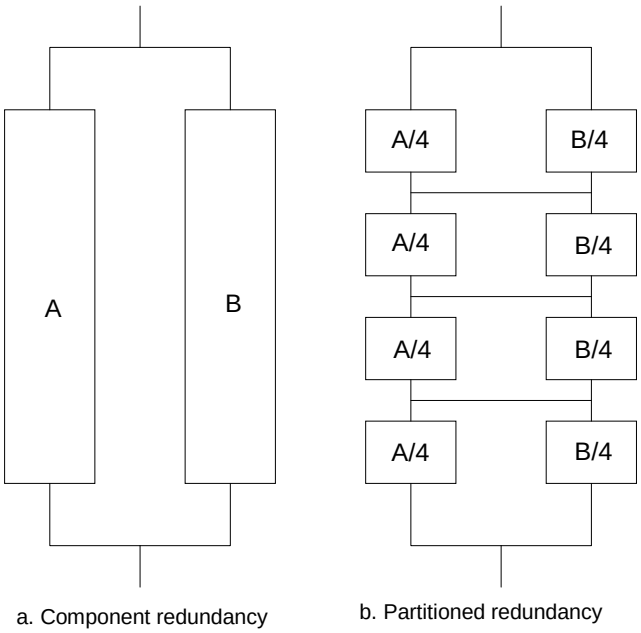


Figure 4. Component and Partitioned Redundancy

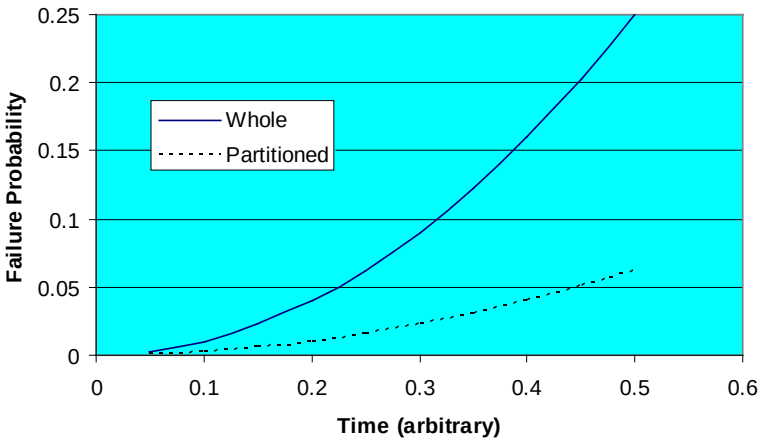


Figure 5. Failure Probability vs Time

The physical resources required for both techniques in figure 4 are approximately the same, and figure 5 shows that partitioned redundancy has a significant reliability advantage, particularly if longer replacement intervals are involved[*]. The partitioned redundancy structure provides a functional path for up to four failures, as long as they do not involve the same horizontal partition.

A major motivation for the development of partitioned redundancy was the NASA planetary mission program, which involved long mission times without the possibility of maintenance [12]. As shown in figure 5, the disadvantage of the partitioned approach is that it was assumed that all of the cross-ties would function perfectly and that only data from normally functioning elements would be sent on to the next partition. That premise is not easily achieved in practice and that was the reason for both ADIRU failures in table 1. Figure 4 examines dual redundancy in which a comparison can be used to detect a failure but additional cues are required to indentify the failed unit. The ADIRU fault management involves three or more units in which, in principle, the failed unit can be identified by majority rules. Erroneous data from a failed ADU continued to be passed on to the displays and the controls.

The ADIRU switches available to pilots of the A330 is shown in figure 6, which is taken from the ATSB interim report [6]. The upper part of the figure shows how IR and air data parts could be deactivated individually by pushing the OFF button. The selector switches permitted full data (NAV) or only attitude and heading (ATT) to be used or to shut off an ADIRU completely. The lower part of the figure shows how ADIRU output could be switched to the Captain's and the First Officer's displays.

[*] The time axis may be interpreted as either total mission time without replacement or time to a replacement opportunity.

Figure 9: ADIRS control panel schematic

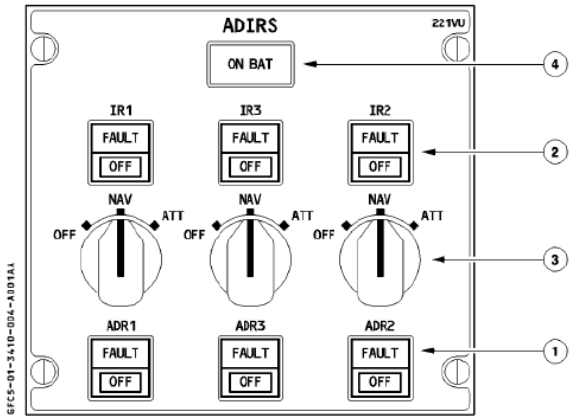


Figure 10: ATT HDG and AIR DATA switches

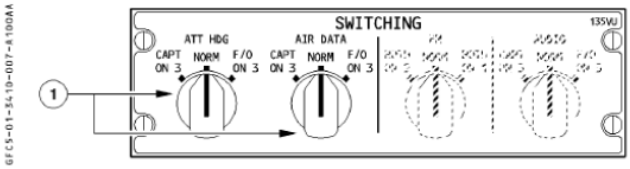


Figure 6. The ADIRU Switching in the A330

In addition to manually switching, the flight control computers could automatically deselect ADIRU input that was identified as failed. The multiple selection possibilities increased the possibility of operational errors and may have contributed only marginally to reducing maintenance and spares requirements. Therefore, the following quote from the ATSB is significant:

> *On 15 October, OEB-A330-74-1 was dispatch [by Airbus], applicable to all A330 aircraft fitted with Northrup Grumman ADIRUs. The OEB stated that in the event of a NAV IR FAULT (or an ATT red flag being displayed on either the captain's or first officer's PFD), the required procedure was for the crew to select OFF the relevant ADR and then select OFF the relevant IR. [6]*

Each combination of air data and IR output of the three ADIRU represents a separate equipment mode that should be periodically tested to detect failures in the switches and wiring that could interfere with using the signals in the rest of the aircraft. When this implicit requirement becomes explicit as part of the certification process, it may inhibit partitioning for marginal maintenance benefits.

The above paragraphs refer to the multiplicity of equipment modes that can contribute to faulty crew RM. In addition, modern aircraft have multiple operational modes (often with subtle differences in equipment utilization and monitoring requirements) that are difficult for the crew to master under emergency conditions. The alpha-prot in the Airbus family of aircraft is an example of this. This feature is specifically provided only in the fly-by-wire mode but is also used in the autopilot mode in modified form. A significant factor in the incidents discussed in

section 4 was the automatic change (due to violation of some engagement constraint) from the autopilot mode to fly-by-wire and the subsequent flight path change due to alpha-prot. In several cases, the remedial action was to loosen the conditions under which alpha-prot became active. This raised the question whether the certification process considered all circumstances under which protective measures could become unprotective, exposing the aircraft to more hazards.

The issue of mode confusion has been dealt with in greater detail in efforts specifically dedicated to that topic [13]. In the incidents studied in section 2, unexpected side effects following automatically initiated mode transitions were frequently the significant contributors to the hazard at the aircraft level. Examples are:

- Inadvertent throttle retardation as a result of radio altimeter failure and a designed automated mode change when at low altitude (table 1)

- Pilot-induced oscillations in which one initiating factor was autopilot disconnect as a result of setting a new target altitude (table 1)

- Autopilot unexpectedly disengaged when a correctly functioning radio altimeter interpreted passing above another aircraft as proximity to ground (table 3)

- Wind shear disengagement not operative when the autopilot was coupled to the #2 flight director (table 3)

The selection of the number of flight and engine control modes is the responsibility of the aircraft designer. Usually modes are added to reduce pilot workload to increase aircraft safety. However review of the incidents indicates that there are also disadvantages associated with increasing the number of control system modes. This suggests that the certification process consider the following questions.

- Are the safety and functional advantages of adding a mode significantly greater than the disadvantages in terms of pilot workload (or required additional pilot training), additional documentation, and additional test requirements?

- If the mode change is automatic, is it clearly annunciated to the cockpit crew?

- Does mode-switching involve a change in secondary functions and monitoring provisions? If so, is the pilot made aware of this change?

3.2 MAINTENANCE.

Failure to act on unsafe, but not immediately hazardous, conditions was a factor in the following incidents.

- Lack of follow-up on repeated false outputs from the radio altimeter was the initiating cause of the Turkish airliner crash at Schiphol Airport (section 2.2.1).

- The Honeywell ADIRU failure might have been prevented by stipulating a time limit for replacing failed instruments recognizing that, any existing failure increases the vulnerability to additional failures (section 2.2.3).

- Inattention to repeated log entries of a failed artificial pitch feel unit was a major factor in the pitch oscillations of the Olympic Airways Falcon 900 (section 2.2.6).

- Failure to install available modification kits in the pitch computer was one of the factors responsible for glide slope oscillations (section 2.3.7). The AD made the modification mandatory.

- Miswired actuators were the result of faulty maintenance that should have been prevented by compliance with 14 CFR 25.671 (section 2.3.11).

The immediate blame for some of these incidents can be put on the maintenance organization. But the certification process can possibly prevent future occurrences of this type by getting answers to the following questions:

- Are the safety consequences of each maintenance action documented in a manner understood by the maintenance organization?

- Are possible effects of a second failure considered in specifying a maintenance interval?

- Is compliance with existing regulations rigorously examined as part of the certification process?

3.3 MONITORING.

Monitoring was a factor in the incidents either because of its absence or because multiple warnings created an environment in which none of them could be acted on. The lack of monitoring for pitch trim failure (section 2.3.8) made that incident more hazardous than it needed to be. On the other hand, multiple warnings issued by monitors and displayed in the cockpit (scrolling displays) were a source of confusion in the NGC ADIRU (section 2.2.2) and nuisance alarms (section 2.3.6) incidents. Multiple warnings were also cited as a contributing factor in several electric power failures that were not included in section 2.

14 CFR 25.1309 (c) states: "Warning information must be provided to alert the crew to unsafe system operating conditions, and to enable them to take appropriate corrective action." The absence of alarms violates the requirement "…must be provided…" and the multiple alarms violate the requirements "…enable them to take appropriate action." [1] The key documents for determining compliance with these provisions are the Functional Hazards Analysis, which establishes the unsafe operating conditions, and the Failure Modes and Effects Analysis (FMEA), which shows how the hazard is avoided or contained. The role of these documents in the certification of complex systems (e.g., flight controls) is discussed in Society of Automotive Engineers (SAE) 4754 [14]. Neither the original documents nor review notes were made available for this investigation, and it is therefore not possible to point to specific review activities that need to be strengthened. Review questions that may be helpful include:

- Are hazardous effects of equipment failures and crew actions evaluated for all flight conditions, particularly those that may arise as a consequence of the failures or actions?

- Do indications of monitoring and alarm systems direct the crew to the steps that are most certain to contain the hazardous condition?

3.4 ELECTRIC POWER INTERFACE.

The interface with electric power was cited twice as a cause of an in-flight incident in table 3, which, in both cases, affected the displays.

- In section 2.3.1, the remedial action was a more complete separation of electric sources between the normal and standby instruments, which was probably an oversight in the original design.

- In section 2.3.3, the cause was a timer that was initialized when power was applied and overran (timed out) when power was continuously applied. The condition that the timer was intended to protect against is not known, but the ultimate corrective action was to eliminate the timer.

The glass cockpit has made it mandatory that electric power to the displays and their drivers be available under all circumstances. The certification procedure should include a failure modes and effects analysis to ensure that a failure in any part or nonadherence to a procedure, going from the power source to the instruments, will not interrupt power.

3.5 DESIGN.

The glide slope oscillations described in AD 2003-11-19 were due to time-based gain scheduling on the assumption that the descent would always be flown at the recommended airspeed. This is not necessarily a valid assumption because (1) the convergence of the glide slope beam is a governed by ground distance and, where distance information is available, it is a much more meaningful criterion for gain scheduling than time; and (2) if a speed criterion is used, it should be ground speed. In the event that led to the AD, the airspeed for approach had been changed from that originally assumed in the design of the glide slope coupler, which further aggravated the likelihood of oscillations. Communicating design assumption to the organization responsible for aircraft operations is a difficult task.

The problem is that time from intercept is taken as a proxy for distance from the ground end of the glide slope beam. The validity of the relation between time and distance can be affected by a number of factors, e.g., in this case, speed during the approach phase. Whenever proxies are used, the certification process should consider all circumstances that can invalidate the proxy assumptions.

3.6 CAUSE ANALYSIS.

The safety and reliability of aircraft systems, and particularly of flight control and associated systems, is very high. When failures do occur, they are mostly due to the system encountering

rare events that were unforeseen by system developers and not detected by the equipment. This finding is not unique and it is not restricted to the flight controls area, as evidenced by the following quotations:

> *The main line software code usually does its job. Breakdowns typically occur when the software exception code does not properly handle abnormal input or environmental conditions—or when an interface does not respond in the anticipated or desired manner.* [15]

> *Therefore the identification and handling of the exceptional situations that might occur is often just as (un)reliable as human intuition.* [16]

Thus, efforts to reduce the number of adverse incidents due to flight control malfunctions must focus on understanding why developers and reviewers tend to overlook problems that involve handling exceptional conditions. One of the possible causes is that situations that are unlikely to occur do not receive the same attention as those that are more likely to be encountered. Also, situations that involve a combination of several conditions are inherently much more difficult to analyze and protect against than those that are due to a single cause or multiple causes. But, given these limitations, awareness of their existence can be reflected in certification requirements that emphasize simplicity and transparency in the design and presentation of the handling of anomalous conditions.

The most prominent causes identified in the preceding discussion are RM, mode changes, and the interaction with maintenance. The following questions should be asked:

- For RM:
  - Is the reduction in spares and maintenance requirements due to partitioned redundancy really worth the increase in design, verification, testing, training, and exposure to additional failure modes?
  - Can the redundancy architecture be represented by a number of mutually independent and easily analyzed areas, such as the containment areas used in the Honeywell ADIRU (see figure 2)
- For automated mode switching:
  - Is the reduction in pilot workload worth the additional testing and the risk of mode confusion?
  - Is automatic disengagement of flight control and throttle functions clearly necessary, and is it unmistakably announced to the pilots in critical situations? If not, can this be resolved by either changing the criteria for automatic disengagement or by a much more prominent annunciation that disengagement has occurred?

Maintenance lapses that affect safety are not restricted to the flight control area. Supervision and inspection of maintenance activities are obviously topics of much broader interest. But in flight control certification, the following questions should be asked:

- Is equipment status pertinent to maintenance easily available to maintenance and flight personnel?

- Are the safety-critical aspects of all flight control maintenance actions made known to the relevant personnel?

A significant conclusion is that the software implementation processes of functions defined in the design specification (the primary focus of DO-178B [17]) can be considered adequate and, therefore, contributed very little to the causes of the incidents that were examined above.

A more structured and detailed review of the requirements for handling rare events and of the process by which the requirements are reviewed could have prevented many of the incidents. For example, in the first incident from table 1 regarding inadvertent throttle retardation:

- Was there a requirement that stated automated landing could be completed with a failed radio altimeter? If so, were provisions for implementing this requirement analyzed during design reviews, and were they tested?

- Was there a requirement for comparing left and right radio altimeters and to announce discrepancies in the cockpit?

- Was there a requirement to indicate in the cockpit that the throttle controls were in imminent landing configuration?

Section 6 of this report makes recommendations for a structured review of requirements for handling rare events that may reduce the lapses and errors in this area.

Since the focus of this report was on equipment certification, flight crew and maintenance certification was not addressed. However, review of several of the incidents led to the conclusion that the extent of equipment damage, injuries, and deaths could have been reduced by better airmanship, particularly in the following incidents:

- Inadvertent throttle retardation (section 2.2.1)
- Pilot-induced oscillations (section 2.2.6)
- Glide slope oscillations (section 2.3.7)
- Wind shear disengagement nonoperative (section 2.3.10)

4. DESIGN ASSURANCE DOCUMENTS.

4.1 OVERALL ASSESSMENT.

The documents discussed in this section are intended to permit verification that the design provides safe flight and landing of transport aircraft. The term “assurance” must be qualified by

"if properly maintained and crewed by competent and attentive pilots." When flight-critical functions fail and pilot intervention is required to maintain safety of flight, it may be necessary to include alerting and warning devices that assure, with a very high probability, the flight crew's attention and understanding of the corrective action needed.

The following documents convey detailed requirements for maintaining safety of flight and are discussed in this section:

- 14 CFR 25.1309—Equipment, Systems, and Installations [1]
- Advisory Circular (AC) 25.1309 [18]
- RTCA DO-178B [17], DO-254 [19], and DO-160F [20]
- SAE Aeronautical Recommended Practice (ARP) 4754 and ARP 4761 [14 and 21]

14 CFR 25.1309 [1] is a governing document and AC 25.1309 elaborates on it by providing nonmandatory guidance on meeting the requirements of the CFR. The RTCA documents provide further detailed guidance on complying with the certification requirements at the software and component levels. The SAE documents address the design, verification, and testing of complex systems. Aircraft, systems, and equipment that are certified within this framework have been found to be serviceable and to provide a high degree of safety. But all the documents listed are at least 10 years old, and equipment design and development are not static. Advances in digital and semiconductor technologies have permitted a drastic reduction in weight, volume, and power requirements. System designers have taken advantage of this by combining functions (e.g., higher integration and greater complexity) and by increasing the use of redundancy.

The practically unlimited computer memory may have encouraged an empirical approach where code that does not meet all requirements is modified with conditioner if-else statements, rationalized for each individual functional deficiency. This is in contrast to former design discipline that emphasized adherence to first principles and, thus, forced much simpler and more easily reviewed implementations. The increased sophistication of flight and engine control systems is beneficial only if it can be properly used by pilots and if the safety aspects can be reviewed as part of certification. The incidents reviewed in the previous two sections show that this is not always the case.

The RTCA and SAE efforts depended on volunteers, mostly drawn from industry and academia. The present environment sponsorship of working groups for keeping the documents current cannot be taken for granted and some government funding may be required.

System designers attempt to meet requirements with tools, techniques, and components at their disposal. At times, the availability of new tools and components can drive modification of the requirements. In the incidents described in section 2, consider the glide slope oscillations (section 2.3.7) in which a change in instrument approach procedures required a change in the approach coupler software many years after certification. Most designers will recall instances when availability of a new monitoring device required changes in the equipment being monitored and possibly in the system wiring. Circumstances like these make it likely that

requirements change during implementation and even thereafter. Accommodation to this reality would strengthen the certification process.

4.2 TITLE 14 CFR 25.1309—EQUIPMENT, SYSTEMS, AND INSTALLATIONS.

14 CFR 25.1309 [1] is contained in Subpart F of 14 CFR Part 25 "Airworthiness Standards: Transport Category Airplanes." The version cited below reflects Amendment 25-123, which was effective 12/10/07. The portions applicable to flight control and navigation systems are cited below.

> *(a) The equipment, systems, and installations whose functioning is required by this subchapter, must be designed to ensure that they perform their intended functions under any foreseeable operating condition. [italics added]*
>
> *(b) The aircraft systems and associated components, considered separately and in relation to other systems, must be designed so that—*
>
> *(1) The occurrence of any failure condition which would prevent the continued safe flight and landing of the aircraft is extremely improbable, and*
>
> *(2) The occurrence of any other failure condition which would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions is improbable.*
>
> *(c) Warning information must be provided to alert the crew to unsafe system operating conditions, and to enable them to take appropriate corrective action. Systems, controls, and associated monitoring and warning means must be designed to minimize crew errors which could create additional hazards.*
>
> *(d) Compliance with the requirements of paragraph (b) of this section must be shown by analysis, and where necessary, by appropriate ground, flight, or simulator tests. The analysis must consider--*
>
> *(1)Possible modes of failure, including malfunctions and damage from external sources.*
>
> *(2) The probability of multiple failures and undetected failures.*
>
> *(3) The resulting effects on the aircraft and occupants, considering the stage of flight and operating conditions, and*
>
> *(4) The crew warning cues, corrective action required, and the capability of detecting faults.*

It should be noted that failures include physical component failures, as well as design errors or deficiencies, resulting in the required function not being performed in an intended safe manner. To emphasize this requirement, clarification of 14 CFR Part 25 may be desirable.

Guidance on interpretation of paragraphs (b), (c), and (d) is provided in AC 25.1309 1A[18] and is discussed below.

14 CFR 25.1309 is augmented by 14 CFR 25.671 [2] and 14 CFR 25.672 [3]. Full compliance with the following provision of 14 CFR 25.671 could have avoided the miswired actuator failure (section 2.3.11):

> *(b) each element of each flight control system must be designed or distinctively and permanently marked to minimize the probability of incorrect assembly …*

Similarly, compliance with 14 CFR 25.672 [3] could have prevented or mitigated the following incidents:

- Inadvertent throttle retardation as a result of radio altimeter failure and a designed automated mode change when at low altitude (section 2.2.1).
- Pilot-induced oscillations in which one initiating factor was autopilot disconnect as a result of setting a new target altitude (section 2.2.6)
- Unexpected autopilot disconnect when a correctly functioning radio altimeter interpreted passing above another aircraft as proximity to ground (section 2.3.4)
- Wind shear disengagement inoperative when autopilot coupled to #2 flight director (section 2.3.10)

The pertinent provisions are

> *If the functioning of stability augmentation or other automatic or power-operated systems is necessary to show compliance with the flight characteristics requirements of this part, such systems must comply with Sec. 25.671 and the following: (a) A warning which is clearly distinguishable to the pilot under expected flight conditions without requiring his attention must be provided for any failure in the stability augmentation system or in any other automatic or power-operated system which could result in an unsafe condition if the pilot were not aware of the failure.*

4.3 ADVISORY CIRCULAR 25.1309-1A.

The version discussed here is AC 25.1309-1A [18], dated 6/21/88. It describes an acceptable means of complying with 14 CFR and does not preclude other means. Most of the incidents described in section 2 could have been avoided by adherence to AC 25.1309-1A. Therefore, it is not the content of the AC, but rather, the means of determining compliance that need to be investigated.

Section 5 of the AC discusses the requirement for a fail-safe design with particular emphasis on the following techniques:

> "(1) *Designed Integrity and Quality*, including Life Limits, to ensure intended function and prevent failures.
> (2) *Redundancy or Backup Systems* to enable continued function after any single (or other defined number of) failure(s); e.g., two or more engines, hydraulic systems, flight control systems, etc.
> (3) *Isolation of Systems, Components, and Elements* so that the failure of one does not cause the failure of another. Isolation is also termed independence.
> (4*) Proven Reliability* so that multiple, independent failures are unlikely to occur during the same flight.
> (5) *Failure Warning or Indication* to provide detection.
> (6) *Flight Crew Procedures* for use after failure detection, to enable continued safe flight and landing by specifying crew corrective action.
> (7*) Checkability*: the capability to check a component's condition.
> (8) *Designed Failure Effect Limits*, including the capability to sustain damage, to limit the safety impact or effects of a failure.
> (9) *Designed Failure Path* to control and direct the effects of a failure in a way that limits its safety impact.
> (10) *Margins or Factors of Safety* to allow for any undefined or unforeseeable adverse conditions.
> (11) *Error-Tolerance* that considers adverse effects of foreseeable errors during the aircraft's design, test, manufacture, operation, and maintenance."

Another purpose of this AC is to elaborate on the inverse relationship between severity and probability of failure that is implied in paragraph (b) of the CFR. The severity of failure conditions is defined in paragraph 6(h) as summarized below:

- Minor—Failure conditions that would not significantly reduce aircraft safety and involve crew actions that are well within their capabilities. Minor failure conditions may include, for example, a slight reduction in safety and involve crew actions that are well within their capability.

- Major—Failure conditions that might cause

  (i) A significant reduction in safety margins or functional capabilities, a significant increase in crew workload or in conditions impairing crew efficiency, or some discomfort to occupants

  (ii) In more severe cases, a large reduction in safety margins or functional capabilities; higher workload or physical distress, such that the crew could not be relied on to perform tasks accurately or completely; or adverse effects on occupants.

The conditions of (ii) are usually referred to as major severe.

- Catastrophic—Failure conditions that would prevent continued safe flight and landing.

In paragraph 7(d) of the AC, the inverse relationship between severity and probability of failure is stated as

(1) Minor failure conditions may be probable.
(2) Major failure conditions must be improbable.
(3) Catastrophic failure conditions must be extremely improbable.

In paragraph 10 of the AC, the qualitative probability statements are defined quantitatively but with the proviso that qualitative assessments may be acceptable in some circumstances. The quantitative ranges are:

- Probable - $>10^{-5}$ per flight hour
- Improbable - $<10^{-5}$ per flight hour and $>10^{-9}$ per flight hour
- Extremely improbable - $<10^{-9}$ per flight hour

For events that occur only during specific flight phases, these definitions can be modified.

An arsenal version of the AC has been in existence for some time; it categorizes failure conditions as minor, major, hazardous, and catastrophic. This classification, although unofficial, is widely used and is the basis for software classifications used in DO-178B [17] (see table 4).

Table 4. Aircraft Failure Severity and Software Level

Failure Condition	Software Level
Catastrophic	A
Hazardous (severe major)	B
Major	C
Minor	D
No Effect	E

With regard to software, the AC states in paragraph 7(i):

> *In general, the means of compliance described in this AC are not directly applicable to software assessments because it is not feasible to assess the number or kinds of software errors, if any, that may remain after the completion of system design, development, and test. Advisory Circular 20-115A dated August 12, 1986, "Radio Technical Commission for Aeronautics Document RTCA/DO-178A," or later revisions thereto provides acceptable means for assessing and controlling the software used to program digital computer-based systems. Document RTCA/DO-178A dated March 22, 1985, "Software Considerations in Airborne Systems and Equipment Certification," defines and uses certain terms to classify*

the criticalities of functions. For information, these terms have the following relationships to the terms used in this AC to classify failure conditions: failure conditions adversely affecting non-essential functions would be minor, failure conditions adversely affecting essential functions would be major, and failure conditions adversely affecting critical functions would be catastrophic.

DO-178A has been superseded by DO-178B, and that document will be discussed in the next section.

4.4 RTCA DO-178B, DO-254, AND DO-160.

DO-178B [17] was prepared by Special Committee-167 of RTCA and issued in December 1992, superseding and substantially revising DO-178A. The scope of the document can be evaluated from its Table of Contents.

The heavy emphasis on process means that the documentation submitted to the certifying authority primarily verifies the correctness and integrity of the procedures that were used in generating, verifying, and supporting the software product. As far as the authors of this report could determine, none of the incidents discussed in section 2 were due to errors or omissions in the software processes per se.

Only Chapter 2 of DO-178B pertains to the system aspects, and that chapter is discussed below because problems in translating system requirements to software requirements may have contributed to some of the incidents.

DO-178B does not address system requirements. The generation (and to a limited extent the validation) of system requirements is discussed in SAE 4754.

A major part of Chapter 2 pertains to the translation of aircraft failure condition severity (from AC 25.1309-1A [18]) into software levels, where the level is defined as "…the effort required to show compliance with certification requirements" [17]. Table 4 represents the translation.

The software level designations control the effort required for each of the following activities:

- Software planning process
- Software development process
- Verification of software requirements process outputs
- Verification of software design process outputs
- Verification of software coding and integration process outputs
- Testing of integration process outputs
- Verification of verification process results

In most cases, the system requirements for the levels A and B can be met only by redundancy or backup provisions. The presence of these provisions can create uncertainties in the assignment of the appropriate level for the individual channels. These are addressed in an FAA policy statement [22].

As shown in table 4, the major classification in AC 25.1309-1A has been split into two categories, and the difference in the software effort can be quite significant. Some examples from Annex A, Table A-4, Verification of Outputs of the Software Design Process, are shown below. The overall DO-178B verification requirement is shown in italics. The interpretation for a given software level is shown in normal font.

- *Low-level requirements comply with high-level requirements*

  - Verify with independence (meaning independent organization) for levels A and B
  - Verify (no requirements for independence) for level C
  - No verification required for D and E

- *Low-level requirements are verifiable*

  - Verify for levels A and B
  - No verification required for C, D and E

- *Software architecture is compatible with high-level requirements*

  - Verify with independence for level A
  - Verify for levels B and C
  - No verification required for D and E

Verification is expensive and time-consuming. Verification with independence requires much more effort than simple verification. Therefore, the software developer should attempt to get as much software as possible into the lower DO-178B levels. SAE ARP 4754 [14] (see below) together with the aforementioned FAA Policy Statement ANM-03-117-09 [22] establish guidance regarding the classification of software elements and prevent inappropriate lowering of requirements.

The passing of safety requirements from system to software developers may have contributed to some of the incidents. DO-178B depicts two steps: (1) at the beginning of software development, system safety requirements allocated to software are passed to software (together with safety strategies defined at the aircraft level) and (2) at the end of software development, the software architecture, partitioning strategies, and observations on possible error sources are passed back to the system, where the system safety function is responsible for verification of the certifiability of the software. This interface definition does not allow for changes in system requirements during software development process. Also, if the system's functional or safety requirements are incomplete or in error, they are not likely to be discovered by this feedback and testing process, because the tests would be designed to cover the software requirements specification only. A partial remedy may be found in the structured requirements review discussed in section 6.

As discussed in section 2, practically all software-related problems arose from RM and mode-switching (including annunciation), and none from normal functions (e.g., stability in attitude or flight path control). Yet, DO-178B makes no distinction based on the subject matter served by the code. This results in inefficient allocation of resources in development as well as in certification. It is suggested that revisions of DO-178 address this issue.

DO-254 [19] may be regarded as the hardware-related equivalent of DO-178. However, through AC 20-152 [23] the application of DO-254 is restricted to field programmable gate arrays, programmable logic devices, and application-specific integrated circuits.

These components frequently serve the same purpose as firmware (software implemented in read-only memory), and thus, the AC establishes requirements for them that are equivalent to DO-178B.

DO-160F [20] is a standard for environmental testing of aircraft systems and components. In addition to conventional environmental conditions (temperature, vibration, altitude, etc.), it also covers robustness with regard to electromagnetic radiation. DO-160 is frequently used in accident investigations to determine whether environmental effects might have been a cause.

4.5 SAE ARP 4754 AND ARP 4761.

SAE ARP 4754 [14] defines highly integrated systems as systems that integrate multiple aircraft level functions (e.g., air data and IR data). Complex systems are defined as being difficult to analyze by conventional techniques.

Section 5 of ARP 4754 also describes how redundancy can be used to meet safety requirements. It distinguishes between similar and dissimilar redundancy ([14, Appendix B] and between active and passive (standby) redundancy. Little guidance is provided about the means for sharing performance (in the case of active redundancy) or for detecting a failure and transferring performance of a function (for standby systems). Yet, it is in these features that many failures of redundancy schemes arise. Appendix A of this report examines strength and weaknesses pertinent to certification of commonly used mechanisms for implementing redundancy, thereby filling this gap.

SAE ARP 4761 [21] describes techniques for conducting the essential steps of the certification process presented in ARP 4754. Both documents are primarily aimed at electronic, digital, and software-based equipment, and both use the failure severity classifications shown in table 4.

Section 5 of ARP 4754 deals intensively with requirements capture and the basis of requirements for the selection of development assurance levels. It also recognizes the importance of derived requirements.

> *At each phase of the development activity, decisions are made as to how particular requirements or groups of requirements are to be met. The consequences of these design choices become requirements for the next phase of the development. Since these requirements result from the design process itself, they may not be uniquely related to a higher level requirement and are referred to as derived requirements.* [14]

This is a much more realistic assessment than the waterfall model in which requirements are simply propagated from one level to the next. The propagation and expansion of requirements for exception-handling have also received attention in the academic community. In one of these investigations, it is suggested that the derived requirements themselves, and their interaction with the propagated requirements, need to be considered [24].

However, even these extensions do not yet fully capture the true development environment of advanced aircraft in which requirements from unrelated systems can affect the requirements of a new system. These relationships are shown in figure 7.

An example of requirements from unrelated systems that can arise during development is that failure modes of the nose wheel can necessitate the pilot opting to land in a nose-high attitude even with the risk of a tail strike. The alpha-prot function of the flight control system must accommodate this option.

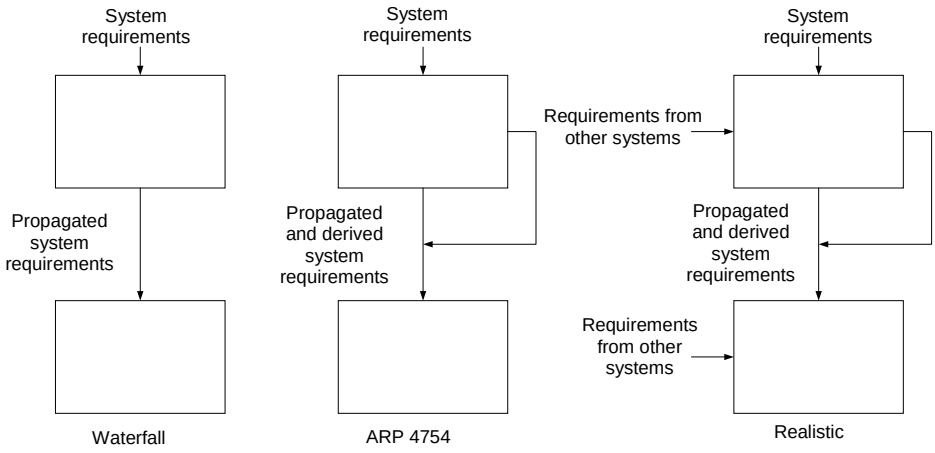


Figure 7. Requirements Generation

Sections 7 (Validation of Requirements) and 8 (Implementation Verification) of ARP 4754 [14] contain checklists that are designed to be used by the developer and by the certification agency, but could be further refined regarding requirements arising from other systems. Strict adherence to these guidelines could have prevented some of the incidents that are described in section 2 of this report. The structured review of requirements outlined in section 6 of this report is a step in that direction.

ARP 4761 can be considered as an extension of Section 6 (Safety Assessment Process) of ARP 4754. The principal steps of the assessment process are

- Functional Hazards Analysis
- Preliminary Systems Safety Analysis
- System Safety Analysis

Among the assessment methods are

- Fault Tree Analysis
- Zonal Safety Analysis
- Common Cause Analysis
- Dependency Diagrams

Some examples of how these methods could have prevented incidents are shown below.
On January 7, 2008, a Qantas Boeing 747-400 on a flight from London to Bangkok, Thailand reported the following instrument and electric system problems while in descent for landing:

- Alternating current bus 1, 2, and 3 not powered
- Autothrottle disconnected automatically
- Autopilot disengaged automatically
- Right (first officer's) displays blanked
- Several pages of messages EICAS
- Lower EICAS display blanked

The aircraft landed using standby power and instruments. Prior to the system anomalies, the cabin crew reported to the cockpit that there was standing water in the forward galley. This galley is directly above the main equipment bay that contained the generator control units (GCU). The drip tray that would have prevented water intrusion was cracked. Four days after the incident, Boeing issued a Multi Operator Message containing instruction for the inspection and repair of the drip shields [25 and 26]. Zonal safety analysis should have revealed the critical nature of the drip shield in preventing water damage to essential electrical equipment. It also might also have discouraged placing all four GCUs in the same location.

In another instance, a transformer in a distribution panel developed a short circuit and overheated. This was identified as a creditable fault, and there was monitoring and switchover to an alternate transformer. However, the short circuit developed very gradually, allowing heat buildup. The mass of the transformer was such that the heat affected the entire circuit board on which it was mounted, and this caused failure of the switching provisions. Common cause

analysis might have identified the monitoring component's susceptibility to heat from a transformer failure.

4.6 DESIGN ASSURANCE DOCUMENT SUMMARY.

The basic FAA guidance for design assurance, 14 CFR 25.1309 [1] and AC 25.1309-1A [18], still forms a good framework for equipment and system certification. The elaboration on the requirements furnished by DO-178B [17], ARP 4754 [14], and ARP 4761 [21] is also valuable and has undoubtedly contributed to the low incidence of aircraft accidents due to equipment design, software, and installation.

The specific electric power system incidents mentioned in the previous section appear to be due to noncompliance with explicit requirements, rather than with deficiencies in the documents themselves. However, most of the incidents described in section 2 were, at least partly, caused by vague requirements and omissions regarding requirements generation and review in some of the guidance documents. The principal areas that need greater coverage are

- validation of requirements for handling of rare conditions, e.g., are the functional and safety requirements specified for a large enough design operating condition space?

- upward and downward traceability of requirements during all phases of the equipment and software development cycle.

- recognition that requirements change during development and operation, feedback of the in-service experience of failures, and design deficiencies to the aircraft and systems developers for a continuous design improvement process.

5. DESIGN AIDS.

5.1 OVERVIEW.

Design aids, as used here, are the software tools that aid in the design of aircraft systems, including their software components. Many of these tools automate routine steps of software generation and are primarily productivity aids. But others, although they may also reduce the labor input, are targeted at the prevention or early detection of design errors and are the subject of the discussion here. One of the earliest tools of that type was the Design Assertion Consistency Checker [27] that validated output assertions against input assertions. The first use of this tool, and the one discussed in reference 20, was to check its own design assertions, and the developers concluded that a tool was definitely needed.

The current tools that are described here are targeted at object-oriented (OO) design and programming. The object is defined by the data structure on which it operates and the operations that it performs. This is somewhat analogous to the way in which electrical components are defined on a schematic diagram, e.g., a resistor by its resistance value and power dissipation and its connection to the rest of the circuit. The circuit designer does not have to be concerned with the internal structure or the manufacturing process of the component. Similarly, in OO design, the internals of the object can remain hidden. OO design lends itself to modeling in the same

way as circuit schematics. This has led to the development of modeling tools, many with capabilities for live simulation of the entire hardware/software system and with the ability to generate code for execution on target computers. These tools also generate documentation for tracing the development from requirements to design to code, thus, automatically satisfying many of the process verification steps in DO-178B (see section 4). The same documentation also forms a basis for performing several of the analyses described in ARP 4754 and ARP 4761, such as FMEA and fault tree analysis.

An important concept of OO design is the class. Objects in a class share properties, and subclasses (children) can be generated that inherit these properties but also have additional properties.

The OO tools of interest for design assurance are described under two headings: domain-independent and domain-oriented. Domain-independent, or general purpose tools, are discussed in the following section and those oriented for the control systems domain (domain-oriented) are discussed in section 5.3. The distinction between these is not clear. The general purpose tools have been used for control systems, and the domain-specific tools may be used in other applications as well. An evaluation of the tools for certification of aircraft control systems can be found in section 5.4.

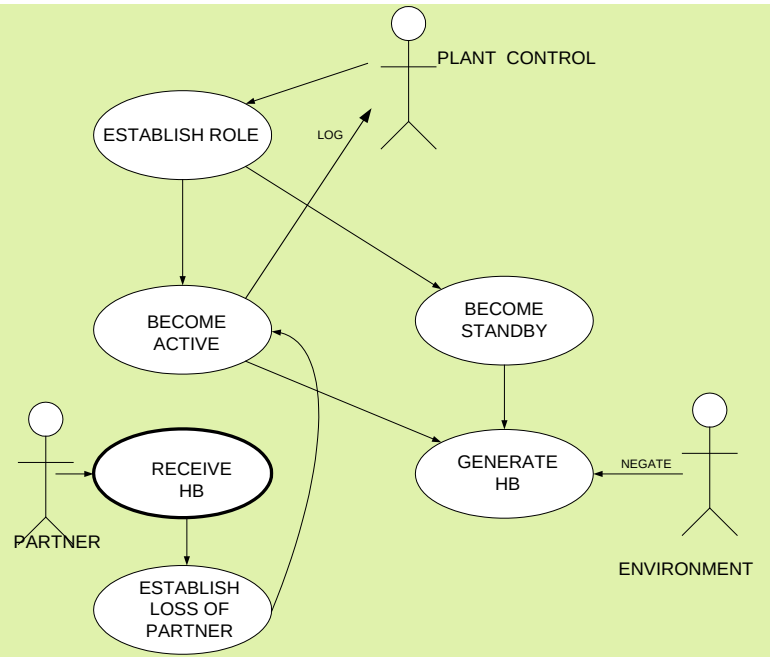
5.2 DOMAIN-INDEPENDENT MODELING TOOLS.

A prime example of domain-independent modeling tools is the Unified Modeling Language (UML), the specification for which is being administered by OMG (Object Management Group, Inc.), a not-for-profit organization based in Needham, Massachusetts [28].

The inherent complexity of the systems being modeled is usually very high, which necessitates working with models that permit isolation of areas of concern. The UML framework supports such isolation by providing three types of models: requirements, structural, and behavioral models.

5.2.1 Requirements Model.

A requirements model is a black-box view of a system that hides all implementation decisions. The emphasis is to identify and characterize the system-level requirements rather than the individual objects and implementation. The set of concepts that the UML brings to a requirements model is based on use-case diagrams, such as the one shown in figure 8.



HB = Heartbeats

Figure 8. Use-Case Diagram for Active/Standby Scheme

The stick figures, called actors, represent external factors, and the ovals represent functions that must be implemented in subsequent development steps. In the above example, an external agent (Plant Control) assigns either active or standby status to the component. If the active status is assigned, its primary function is to generate heartbeats (HB); this can be negated by the environment (e.g., equipment failure). If the assignment is to standby status, the primary function is to listen to the HB generated by the partner (active component). The use-case diagram is a convenient medium to focus on safety concerns.

The heavily outlined "Receive HB" function is subject to two unsafe failure modes:

- Spurious HBs that mask failure of the partner
- Declaring the partner failed due to an internal failure in the function

The expansion of the Receive HB function shown in figure 9 provides an example of protection against these failure modes. Spurious HBs are made highly unlikely by requiring exactly three pulses per interval. Internal failures are detected by normally sending alternating symbols when no failure is detected.

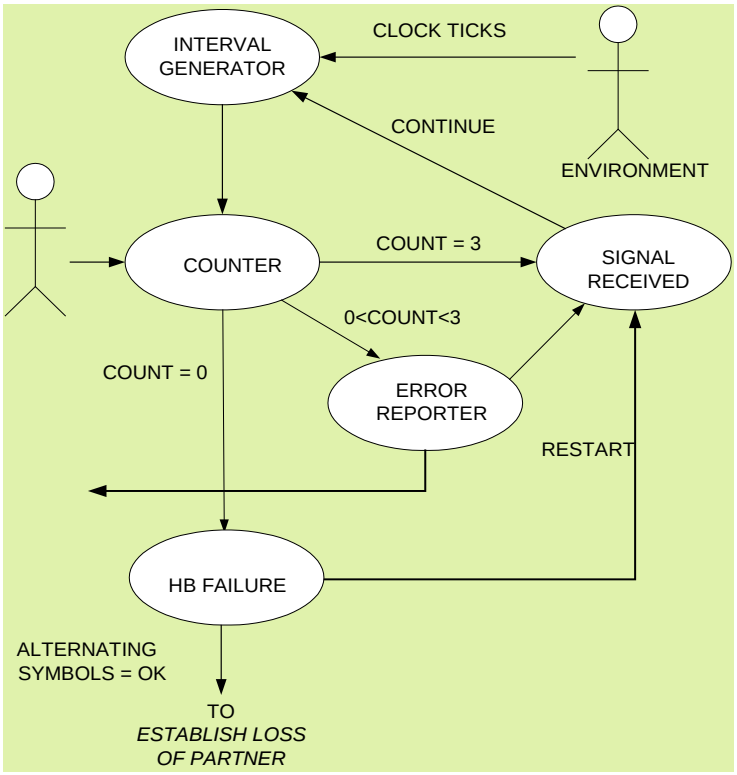


Figure 9. Expansion of Receive HB Function

5.2.2 Structural Model.

A structural model is the means by which the UML provides semantic elements for describing the structure of a system. Structure can be divided into two aspects: (1) the logical structure of a system that reveals the inherent relations among semantic elements regardless of location, and (2) the physical structure that defines what the physical pieces of the system are and how they map to the logical elements. An example of the physical structure model for the Receive HB function (figure 9) is shown in the first two columns of table 5. The information in the remaining columns is generated by the MOCET tool [29] based on failure mode information for each class stored in a library file. The FMEA worksheet has a format and content consistent with its hardware equivalent and, thus, can be a part of the system FMEA.

Table 5. Failure Modes and Effects Worksheet Constructed From the Structure Model

ID	Item/Function	Failure Mode and Causes	Local Failure Effect	Failure Detection	Compensation	Severity Level
1.1.1.1	Interval generator	No interval started. Loss of clock ticks or internal failure.	HB failure	Self-test	Note 1	IV
1.1.1.2	Interval generator	Long interval. Missing clock ticks or internal failure.	HB failure	External	Note 1	IV
1.1.2.1	3-pulse counter	No count. External or internal failure	HB failure	Self-test	Note 1	IV
1.1.2.2	3-pulse counter	Spurious count ≠3 per interval. Internal failure	HB failure	External	Note 1	IV
1.1.2.3	3-pulse counter	Spurious count, exactly 3 per interval. Internal failure	Spurious HB	External	Note 1	II
1.1.3.1	HB failure	Does not send restart. Internal failure	None	Self-test	Note 2	
1.1.3.2	HB failure	Spurious restart. Internal failure	HB failure	External	Note 1	IV
1.1.3.3	HB failure	No or random output to loss of partner. Internal failure	HB failure	External	Note 1	IV
1.1.3.4	HB failure	Spurious defined alternating signals	Spurious HB	External	Note 1	II
1.1.4.1	Signal received	No continue output. External or internal failure.	HB failure	External	Note 1	IV
1.1.4.2	Signal received	Spurious continue output. Simultaneous errors in input and Restart processing	None	External	3-pulse counter	None

5.2.3 Behavioral Model.

Behavioral models include two primary ways of representing behavior in the UML map to the scope of the behavior. State machines define the behavior of individual classifiers, such as classes and use cases. Interaction diagrams depict the behavior of collaborating sets of classifiers. Depending on the type of system being modeled, any single, or a combination of the above models can be used to represent and analyze the entity. Figure 10 is an example of a state machine model generated by IBM® Rational® Rhapsody® [30] a popular tool for OO design and implementation. The entity being modeled is the leader/follower selection for a pair of unmanned aircraft; this is similar to the active/standby selection but at the aircraft level. States are represented by rectangles and allowed transitions are indicated by directed edges (lines).

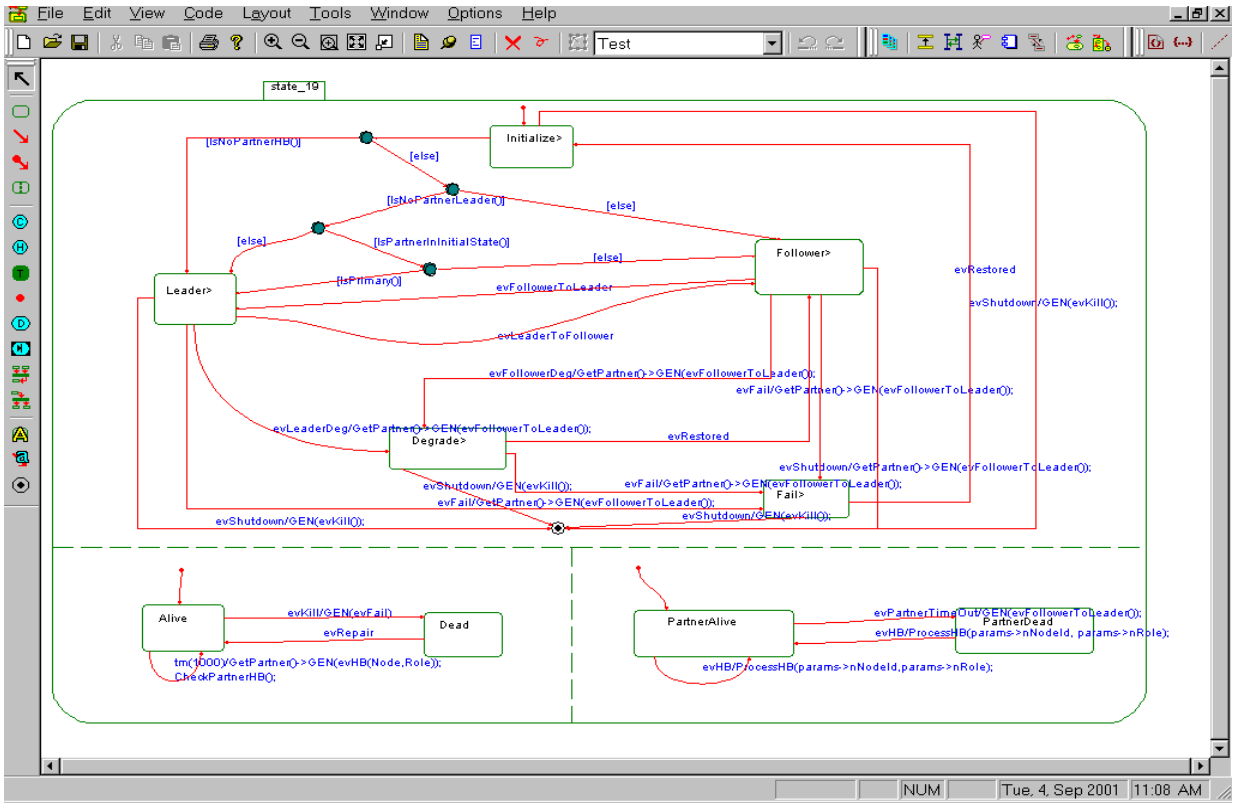


Figure 10. Example of Statechart Representation

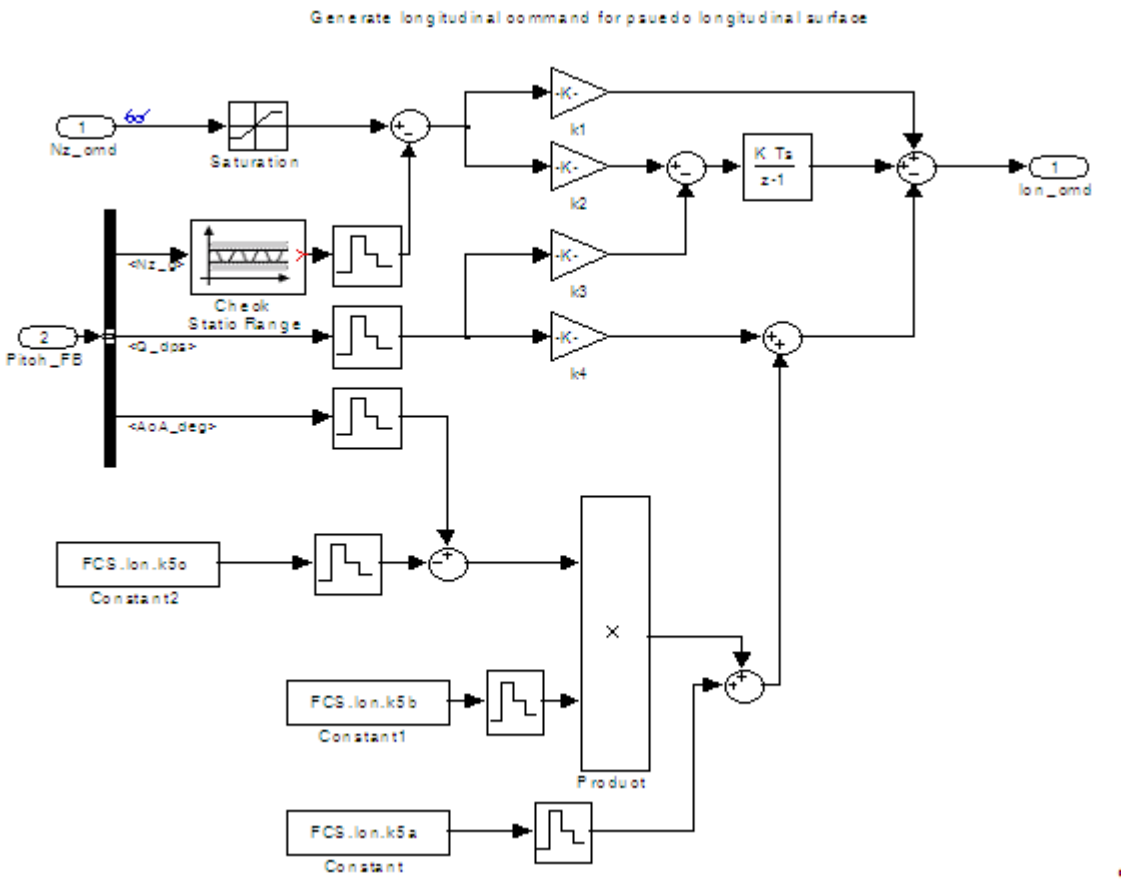
Another important domain-independent modeling tool is Alloy, developed and hosted by MIT [31]. It uses a strictly declarative syntax and is well-suited for some aspects of RM. An example of the modeling syntax for a file system is shown on figure 11. Text shown in italics represents comments and is not part of the model proper.

```
// A file system object in the file system
sig FSObject { parent:  lone Dir }

// A directory in the file system
sig Dir extends FSObject { contents:  set FSObject }

// A file in the file system
sig File extends FSObject { }

// A directory is the parent of its contents
fact { all d:  Dir, o:  d.contents | o.parent = d }

// All file system objects are either files or directories
fact { File + Dir = FSObject }

// There exists a root
one sig Root extends Dir { } { no parent }

// File system is connected
fact { FSObject in Root.*contents }

// The contents path is acyclic
assert acyclic { no d:  Dir | d in d.^contents }

// Now check it for a scope of 5
check acyclic for 5

// File system has one root
assert oneRoot { one d:  Dir | no d.parent }

// Now check it for a scope of 5
check oneRoot for 5

// Every fs object is in at most one directory
assert oneLocation { all o:  FSObject | lone d:  Dir | o in
d.contents }

// Now check it for a scope of 5
check oneLocation for 5
```

Figure 11. Example of Alloy Declarations

5.3 DOMAIN-SPECIFIC MODELING TOOLS.

A widely used domain-specific modeling tool for flight control applications is MATLAB®/Simulink®, developed and marketed by the MathWorks™, Inc. The MATLAB facilitates calculation of aerodynamic parameters and Simulink permits time domain investigations using these parameters. Figure 12 shows the Simulink representation of the longitudinal axis of a typical flight control system.



FCS = Flight control system

Figure 12. Longitudinal Flight Control System

The primary input is a normal acceleration command, Nz, and the output is an elevator command, lon_cmd. There are also three pitch axis feedback signals.

- Actual normal acceleration, Nz_g
- Dynamic pressure, Q_dps
- Angle of attack, AoA_deg

The actual normal acceleration is compared with the commanded normal acceleration, and the difference forms the input to the proportional plus integral control law. The dynamic pressure and AoA signals are used to avoid unsafe speed or AoA states.

The commanded normal acceleration, Nz_cmd, is limited to a safe value that may be a function of airspeed and proximity to ground by the saturation element. The actual normal acceleration, Nz_g, is expected to fall within the same range with allowance for short-term aerodynamic disturbances. When that range is exceeded, it is likely that the flight control system has malfunctioned, when such a condition is encountered, corrective action is necessary, e.g., switching to an alternate control channel. The check range function, which is detailed in figure 13, can identify the out-of-range condition. When it is detected, the normal output of the longitudinal channel is stopped, and by means of the assertion output, remedial action is

initiated. The use of this construct is not limited to normal acceleration inputs and, therefore, a generalized input symbol, u, is shown in the figure.

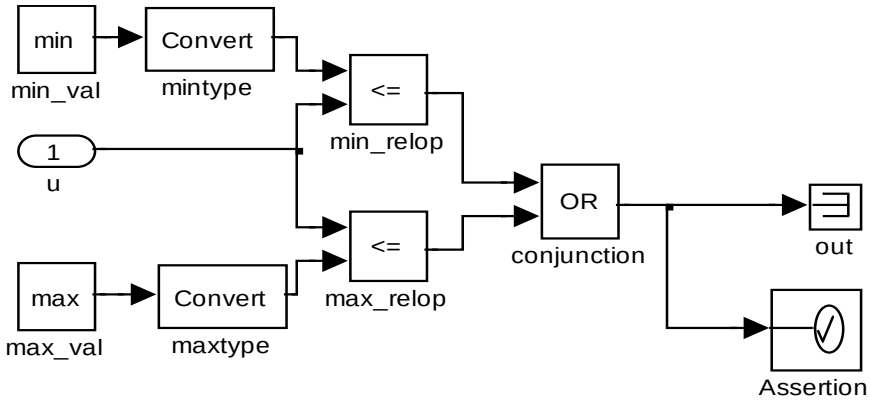


Figure 13. Check Range Function

Passive failures in the sensor or transmission path can also lead to unsafe conditions. These failures will manifest themselves by a zero or quiescent-signal value. The check zero function shown in figure 14, raises an assertion when a signal value has not changed for three computing cycles, and it can initiate remedial action.

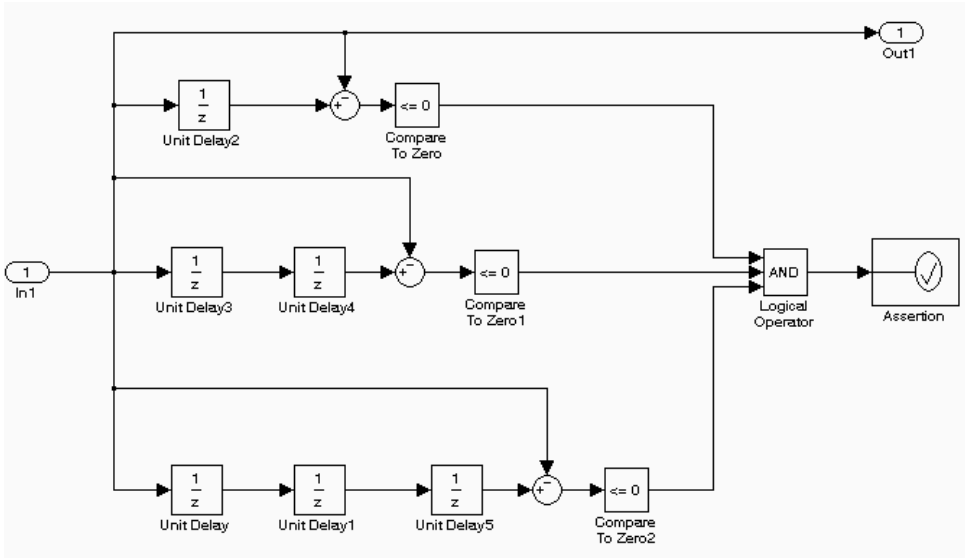


Figure 14. Check Zero Function

MATLAB/Simulink generates structure and behavior files similar to those described for UML in the preceding section. Each symbol represents a class with given properties. The construction of an FMEA is therefore simpler than in UML because failure modes (and in many cases, the detection and mitigation methods) can be directly and permanently associated with a symbol.

Another model-based methodology with domain-specific support for aeronautical and automotive applications is the SAE Architecture Analysis and Design Language (AADL) [32]. It is an outgrowth of the MetaH architecture description language that had been developed at the Honeywell Advanced Technology Center. It is now being supported by major aerospace concerns including Boeing and the parent company of Airbus Industries. It was from the outset intended for both hardware and software components, whereas, UML was originally targeted at only software. However, AADL is similar to UML and shares its code generation capability. It is more text-based than the largely symbol-based representations in Simulink. However, the same properties that can be associated with a symbol (e.g., failure modes) can also be associated with a declared text object. Thus, an FMEA worksheet (like the one shown in table 4) could also have been generated from AADL documentation.

5.4 CERTIFICATION ISSUES REGARDING THE USE OF TOOLS.

Two aspects of tool usage raise certification issues: whether the tool yields consistent and repeatable output, and whether the data generated by the tool demonstrate conclusively that certification objectives have been met. All the tools mentioned in this chapter were developed by major organizations and have been in use for at least 5 years. MATLAB/Simulink and Alloy are widely taught in colleges in language, as well as application, aspects. Thus, there is high confidence that the tools yield consistent and repeatable outputs.

Another issue is a more difficult one. Note that a model is a simulation. Thus, a model can show that there may be a problem, but it cannot be depended on to show that there is no problem. The previously discussed output of the modeling tools dealt with the logic representation of failure modes of flight control systems. But even if the logic is flawless, there can be timing problems. All the tools can furnish sequence charts (or equivalents) that can be used to investigate timing problems. But these must be separately generated and analyzed.

Timing problems can transcend the domain of flight control applications. A thoroughly analyzed RM program for inertial sensors was rendered useless when the watchdog timer of the executive timed out before the recovery routine could complete.

Another aspect that can be evaluated with simulations supported by these tools is the cockpit presentation of anomalous conditions. In several of the in-flight failures discussed in section 2, the overload of alarms and information (scrolling displays) interfered with crew decision-making and taking remedial action. But this cockpit presentation aspect of failures must be independently recognized and appropriately investigated.

Thus, tool usage must be guided by the knowledge of the hazards at the application, flight control system, and overall aircraft levels. While the tools do not qualify as an automated, crank-turning approach to certification, they eliminate or significantly simplify many tasks necessary for assessing the safety of flight control systems. They provide transparency when going from requirements (e.g., use-case diagrams) to state chart presentations and autocode generation. Thus, they really make some of the process verification steps in DO-178B unnecessary.

Like most software tools, they are subject to the "garbage in – garbage out" phenomenon. The "garbage in," in this case, is mostly represented by incorrect or incomplete requirements. Thus, to capitalize on the capabilities of these tools, there must be emphasis on the guidelines for requirements generation.

6. GENERATING REQUIREMENTS FOR HANDLING OF RARE EVENTS.

6.1 HOW REQUIREMENTS ARE GENERATED.

The incidents examined in section 2 of this report were initiated by at least one rare event, such as an equipment failure or an unusual cockpit procedure. In each incident, an exceptional condition was not handled correctly and propagated into a hazardous flight condition. Events, such as equipment failures or unusual cockpit procedures, must be, and usually are, foreseen in the requirements. But in these specific instances, they were not covered by the requirements and, hence, were not tested (the significant part of a system test program is to test for conformance with every requirement). This section investigates how such lapses in requirements generation occur and how they can be eliminated or at least minimized. One way may be to allow for random, possibly unscripted "abusive" testing, meaning outside the defined or intended operating environment, including multiple simultaneous inputs to exercise complex system state space conditions in search of the system functional failure points. This is equivalent to the highly accelerated life testing that evaluates hardware reliability. Only testing to conditions resulting in failure provides evidence of sufficient safety margins or shows areas where design must be made more robust.

The misconception that requirements are fixed at the beginning of a design is at the core of the problem. Actual requirements generation is a much more complex process, as discussed earlier (e.g., in connection with figure 7). The vast majority of requirements are derived particularly when handling rare events, i.e., the need for the requirement arises from a design decision. For example, the decision to use radar altitude when establishing the approach profile necessitates requirements for monitoring the radar altimeter performance and for initiating actions when the performance parameters drop below a threshold.

To evaluate the completeness and correctness of requirements for exception handling, it is helpful to examine typical sources for these requirements. In a typical flight or engine control system, the following may be encountered:

- Operational requirements of the system, including specification of environmental conditions that must be met simultaneously
- Implementation details
- Computing environment
- Monitoring and self-test of system functions
- Application software

Details of each of these are discussed below. As this list indicates, it is fairly obvious that (1) the totality of the requirements cannot be expected to be available at the start of the project, (2) the requirements will originate from multiple organizational units and disciplines, and (3) there will inevitably be differences in format and the level of detail in which the requirements are stated. This is in addition to the acknowledged difficulties of requirements formulation for critical systems [33].

6.2 SOURCES OF REQUIREMENTS.

The following paragraphs present typical requirements for handling rare conditions that can come from the above mentioned sources.

6.2.1 Operational Requirements of the System.

The most frequently encountered operational requirements for exception handling arise from the need for power, communications, and thermal control. How should the system respond when these fail, and how much time is available for recovery? Also, where the system can operate in several modes, safeguards for the issuance of mode changes, and verification of the accomplishment of mode changes, give rise to exception handling requirements.

Aerospace systems are essential for continued safe flight and are frequently made redundant in their entirety or on a channel basis. Exception handling is required to isolate the failed component or channel and to reconfigure the remaining units into a survivable structure. Exception handling may also be required to avoid undesirable states (e.g., excess fuel consumption) by initiating corrective measures or by activating an alarm.

These operational requirements for exception handling are frequently known early in the life cycle and are less likely to change than other requirements. Also, these provisions may exist in predecessor systems, and thus, not only the requirements, but also parts of the implementation, can be specified early.

6.2.2 Implementation Details.

As the system enters the design stage, additional operational and structural details will emerge that give rise to exception handling requirements. In sensor calibration and in monitoring of internal temperatures of actuator states or of output channels, exception conditions may be encountered for which handling needs to be specified. Safeguards against potentially harmful operator commands must be provided. Also, as more detail about interfaces with cooperating systems emerges, safety checks on acceptance or delivery of information will need to be implemented. All these features can result in requirements for exception handling.

Maintenance provisions are another important area under this heading. Is a distinct maintenance mode required? Can spares be hot-swapped? Do maintenance personnel require authentication? How will the system be restored to operational use after maintenance, and how will the completion of this transition be verified? The response to every question can indicate a condition that requires exception handling, including initiation of remedial action to restore the system to operational use.

As the heading implies, these requirements for exception handling are frequently not foreseen at the project initiation. However, once recognized, they can be expected to remain reasonably stable during the rest of the development phase.

6.2.3 Computing Environment.

Some requirements for exception handling arise from the computer hardware, such as handling of memory errors, divide-by-zero exceptions, and over- and underflows. These requirements are usually known shortly after the hardware is specified and can be expected to remain stable unless the computer is replaced.

A substantial part of the exception handling originates in the software component of the computing environment, including the executive or operating system and the middleware. For real-time systems that operate on fixed cycle times, these software components detect when timing constraints are about to be violated, and they may invoke their own exception handling to deal with these conditions, not always with desirable results. Higher-level application programs are usually better informed about the consequences of a missed cycle and about alternative means for accomplishing a function. In some instances, it is necessary to modify or disable the native time-out provisions or watchdog timers in commercial operating systems. These possibilities of undesirable interactions must be kept in mind when establishing and reviewing requirements for exception handling.

Requirements arising out of the computing environment software components may become known later than those from the hardware component and are more likely to change during subsequent development phases.

6.2.4 Monitoring and Self-Test of System Functions.

Although the original system concept may include monitoring and self-test functions, the details of these provisions emerge only after selection of the components to be monitored. Even for functions that were fully identified during the concept phase, the full scope of the exception handling requirements became known much later.

Monitors are usually active at all times, and exception handling is required only when they indicate anomalous conditions. Exception handling must distinguish between failures in the monitor and failures in the monitored component. Self-test may be invoked for any system component, but it is particularly important where passive sensors guard against potentially harmful system states, such as an over-temperature sensor for a critical electronic component. Exception handling must provide for appropriate action under all test outcomes (which may include some unforeseen combination or sequence of test results).

6.2.5 Application Software.

The exception conditions under this heading may be native (arising from a programming error that induces an anomalous state or transition) or external (responding incorrectly to an anomaly in the system). A source for the detection and mitigation of native software failures can be found in the taxonomy compiled by experienced programmers in the field of dependable

computing [34]. Figure 5 of reference 34 lists 12 types of software faults that can impede the intended execution of a program, including maliciously introduced faults that affect the execution during particularly critical operational states. The same reference also includes a taxonomy of recovery provisions that is significant for exception handling. In it, a distinction is made between error handling (replacing erroneous data values) and fault handling (isolation, removal, and replacement of permanently damaged data stores or instructions).

Because these exception conditions are closely tied to the software development, their requirements can only be formulated in general terms prior to software design and must be finalized during the design phase. These requirements are also most likely to change as a result of software test and operation.

It is not claimed that these sources represent an exhaustive listing; certainly, they may need to be augmented in specific situations. Rather, the discussion was intended to show the distribution in time over the development cycle, the diversity of disciplines from which they originate, such as system planners, system and component engineers, and software professionals; and these factors can be assumed to be near universal. The distribution may be helpful in partitioning the effort for review of requirements for completeness and correctness and thereby make it more manageable.

It is also important to recognize that requirements for exception handling can be in conflict, such as when a recovery from a failure causes the allowed time for a function to be exceeded, activating a watchdog timer exception that may negate the recovery. In other instances, one specific requirement interferes with meeting other requirements, e.g., control target capture and tracking versus allowable controller activity in turbulence. In those cases, another design approach may be needed. For all of these reasons, a coordinated, systematic approach to requirements for exception handling is difficult, but it is essential for avoiding the incidents discussed earlier. The following section discusses in more detail how requirements are formulated.

6.3 TIME PHASING OF REQUIREMENTS.

Requirements for exception handling normally arise over most of the system development cycle, and the time phasing of requirements addresses the form in which the requirements are expected to be formulated. First, the typical steps in the evolution of the exception handling requirements are discussed and then how these steps fit into the system development cycle. Evolutionary steps in the generation of a given requirement usually include

- Objectives—these represent the conditions to be prevented or to be achieved; they may reference regulatory or system-level requirements. The operating conditions to which the objectives apply also need to be stated. The document generated at the end of this step permits initiation of the algorithmic descriptions.

- Algorithmic descriptions—these identify how the anomalous condition is to be detected and the action(s) to be taken subsequent to detection, usually in algorithmic format. The document generated at the end of this step permits initiation of the assignment for exception handling and completes the input to software requirements.

- Assignment to a software function—this identifies the software (and sometimes the hardware) function responsible for the execution of the algorithm. Requirements for sampling frequency, execution time for exception handling, and other implementation constraints (e.g., use of a specific sensor output) are usually included in this step. The document generated at the end of this step is an input to software design.

Figure 15 shows a typical distribution of requirements for exception handling over the system development phases. The phase designations are shown along the horizontal axis. Each source category discussed earlier is represented by a horizontal bar. The colored divisions in each bar represent the three steps of requirements development described above. Most requirements inputs are completed prior to the start of the coding phase with only those originating from native application software support showing an overlap. This scheduling should allow for orderly implementation of the exception handling requirements. Guidelines exist for exception handling coding in the most currently used programming languages.

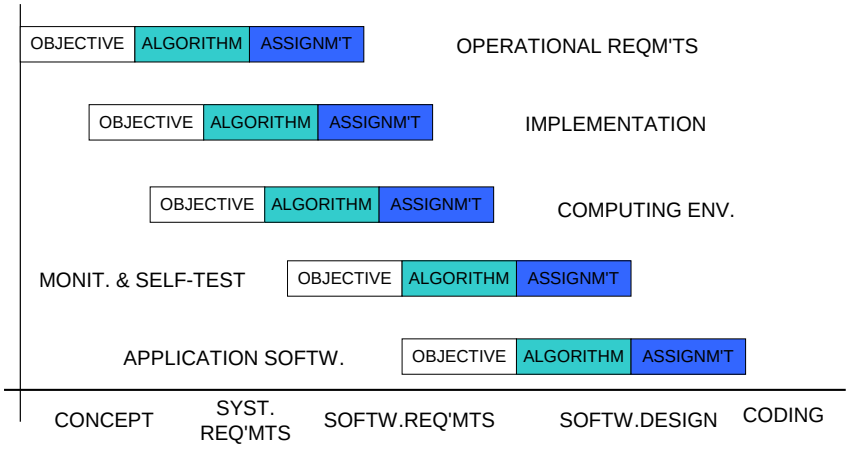


Figure 15. Evolution of Requirements

During testing and in the operation and maintenance phases, it may become necessary to modify the exception handling requirements, and sometimes new requirements may be added. These events are not shown in the figure. Good configuration management demands that the changes not only be made in the program but also that the requirements documentation is updated.

A schedule, such as that shown in figure 15, should be helpful in organizing a systematic approach to the review of requirements for completeness and correctness. Once the review process is partitioned, it may be possible to employ review tools that are not scalable to the totality of the requirements. An example of such a review tool is the condition table [35].

7. CONCLUSIONS AND RECOMMENDATIONS.

Although commercial air travel in the United States has an enviable safety record, the tendency toward higher complexity in both operating modes and redundancy management (RM) raises

concerns about the ability to completely test the system under all conditions as well as to assess pilot proficiency in mastering all the available resources under emergency conditions. This report has shown that

- most of the incidents were due to multiple rare events, e.g., an equipment failure and vulnerability in the software intended to recover from the failure.

- tests for multiple rare events are currently performed on a hit-and-miss basis; the incidents are evidence of misses.

- to avoid such misses, requirements for coverage of rare events must be generated and reviewed for completeness and correctness throughout the system life cycle. The waterfall model, that assumes that requirements are known at the start, is inadequate for this purpose.

These concerns can be addressed by partitioning the review effort in the manner described in this report. The review should predominantly focus on the availability of suitable monitoring and display provisions. In several of the incidents, lack of display or scrolling of multiple warnings contributed to the severity of the effects.

A number of incidents were due to lack of, or improper, maintenance, but this was not all the fault of the maintenance organizations. Allowing the Malaysian Airlines 777 to fly with a failed inertial sensor for 4 years caused a second failure to produce severe pitch-up commands. In other incidents, maintenance personnel were unaware of the safety implications of their assigned tasks. Analyses recommended in the SAE guidance documents should have identified the hazards. From the information available, it could not be determined whether the analyses had been performed, and if so, whether the safety-critical information was transferred to the maintenance manuals.

In many respects, the guidance materials described in this report were adequate, despite their age (well over 10 years in most cases) and the advances in equipment and system design. The complexity of RM has caused uncertainty about the classification of some failures that have been addressed in an internal Federal Aviation Administration (FAA) document. However, guidance on the overall issues of RM will help future certification efforts and is recommended below.

A number of excellent commercial software and system development tools that are available and are being used by developers can also be helpful to reviewers. Possibly, as a result of the use of these tools, many of the historically important failure modes were completely absent from the incidents that were examined. These tools can help identify problems handling rare events when the requirements are complete and correct.

Additional efforts are recommended in the following areas:

- Redundancy management

    - Performance overview of currently used techniques

- Review benefits, disadvantages, and costs (weight, power, etc.) of these techniques. Although these parameters are not direct certification issues, FAA personnel should be aware of them.

- Identify critical factors for review (failure modes, correlated failures, indications of partial failures, etc.)

- Review analysis tools

- Flight control interfaces—review guidelines, partly based on recent incidents, about data interchange with

  - navigation functions (area navigation, Traffic Collision Avoidance System, approach and landing aids)

  - electric power distribution

  - environmental control

  - monitoring and warning systems, such as engine-indicating and crew-alerting system

  - maintenance computers and overall maintenance procedures

8. REFERENCES.

1. U.S. Federal Register, Title 14 Code of Federal Regulations Part 25.1309, "Equipment, Systems, and Installations," Government Printing Office, Washington, DC.

2. U.S. Federal Register, Title 14 Code of Federal Regulations Part 25.671, "Control Systems—General," Government Printing Office, Washington, DC.

3. U.S. Federal Register, Title 14 Code of Federal Regulations Part 25.672, "Stability Augmentation and Automatic and Power-Operated Systems," Government Printing Office, Washington, DC.

4. Aviation Safety Network, ASN Safety Database, http://aviation-safety.net/database, last visited September 23, 2011

5. Onderzoeksraad voor veiligheid (Dutch Safety Board), "Preliminary Report, Accident [of] Boeing 737-800," TC-JGE, April 2009.

6. Australian Transport Safety Bureau, "In-Flight Upset, 154 km West of Learmouth, WA," Occurrence Investigation AO-2008-070, Interim Factual, as of 18 November 2009.

7. Australian Transport Safety Bureau, "In-Flight Upset Event, 240 km North-West of Perth, WA," Occurrence Investigation AO-200503722.

8. Eckhardt, D.E., et al., “An Experimental Evaluation of Software Redundancy as a Strategy for Improving Reliability,” *IEEE Transactions on Software Engineering*, Vol. 17, No. 7, July 1991, pp. 692-702.

9. Aviation Safety Network, ASN Safety Database, Preliminary Report, February 7, 2001.

10. Aircraft Accident Investigation Board (UK), AAIB Bulletin 6/2001, June 2001.

11. Romania Ministry of Transport, Civil Aviation Inspectorate, “Final Report on the Accident of the Falcon 900B Registered SX-ECH, 14 September 1999,” Report No. 711/01.08.2000.

12. Avizienis, A., et al., “The STAR (Self-Testing and Repairing) Computer. An Investigation of the Theory and Practice of Fault-Tolerant Computer Design”, *Proceedings of International Symposium on Fault-Tolerant Computing (FTCS-1)*, Pasadena, California, 1971, pp. 92-96.

13. Miller, S.P., et al., “A Methodology for Improving Mode Awareness is Flight Guidance Design,” *Proceedings of the 21st Digital Avionics Systems Conference (DACS ‘02)*, Irvine, California, October 2002.

14. Society of Automotive Engineers, ARP 4754, April 10, 1996.

15. Hansen, C.K., “The Status of Reliability Engineering Technology 2001,” *IEEE Reliability Society Newsletter*, January 2001.

16. Cristian, F., “Exception Handling and Tolerance of Software Faults,” *Software Fault Tolerance*, Michael R. Lyu, ed., Wiley, New York, 1995.

17 RTCA DO-178B, “Software Considerations in Airborne Systems and Equipment Certification,” RTCA, Inc., Washington, DC, 1992.

18 Federal Aviation Administration, “System Design and Analysis,” Advisory Circular (AC) 25.1309-1A, June 21, 1988.

19 RTCA DO-254, “Design Assurance Guidance for Airborne Electronic Hardware,” RTCA, Washington, DC, 2000.

20 RTCA DO-160F, “Environmental Conditions and Test Procedures for Airborne Equipment,” RTCA, Washington, DC, 2007.

21 Society of Automotive Engineers, ARP 4761, December 1996.

22 Federal Aviation Administration Transport Airplane Directorate, ANM-03-117-09, “Policy Statement on Guidance for Determination of System, Hardware, and Software Development Assurance Levels on Transport Category Airplanes,” January 15, 2004.

23 Federal Aviation Administration, Advisory Circular 20-152, "RTCA DO-254," June 30, 2005.

24 de Lemos, R. and Romanovsky, A., "Exception Handling in the Software Lifecycle," *International Journal of Computer Systems Science and Engineering*, Vol. 16 No. 2, March 2001, pp 167-181.

25 National Transportation Safety Board, Event 20080111X00039

26 Australian Transport Safety Board, AO-2008-003 Prelim.

27 Boehm, B.W., McClean, R.K., and Urfrig, D.B., "Some Experience With Automated Aids to the Design of Large-Scale Reliable Software," *Proceeding of International Conference on Reliable Software*, Los Angeles, California, 1975, pp. 105-113.

28 Object Management Group, Inc., www.omg.org , last visited September 23, 2011.

29 Hecht, H. and Menes, R., "Software FMEA Automated and as a Design Tool," *SAE Aviation Technology Conference*, Wichita, Kansas, 2008.

30 http://www.ibm.com/software/awdtools/rhapsody/ (verified 12 September 2001).

31 Massachusetts Institute of Technology, http://alloy.mit.edu

32 Hudak, J. and Feiler, P., "Developing AADL Model for Control Systems: A Practitioner's Guide," Carnegie Mellon University, CMU/SEI 2007-TR-014, July 2007.

33 Leveson, N.G., "Software Safety: What, Why and How?" *Computing Surveys of the ACM*, Vol 18, No. 2, 1986, pp. 125-63.

34 Avizienis, A., Laprie, J.C., Randell, B., and Landwehr, C., "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 1, January 2004.

35 Goodenough, J.B. and Gerhart, S.L., "Toward a Theory of Test Data Selection," *IEEE Transactions on Software Engineering*, Vol. SE-1, No. 2, June 1975, pp.156-173.

APPENDIX A—CERTIFICATION CONSIDERATIONS FOR ELEMENTS THAT IMPLEMENT REDUNDANCY

The major partitions of redundant systems recognized in Society of Automotive Engineers (SAE) Aeronautical Recommended Practices (ARP) 4754 [A-1] are parallel versus standby and similar versus dissimilar. Certification-related characteristics of these alternatives are described below.

A.1 PARALLEL REDUNDANCY.

In parallel redundancy, all channels are active and contribute to the output. In most cases, no switching is required upon failure of one channel since the remaining channels can override the failed one (by majority voting on digital output or by force balance on mechanical or hydraulic output). Because no switching is required, this implementation is also referred to as static or fault-masking redundancy. A simple example for parallel redundancy is shown in figure A-1. In most direct-current power supplies, failures will reduce the output voltage below the specified value, most commonly to zero. Therefore, dual parallel redundancy can furnish a protected output.

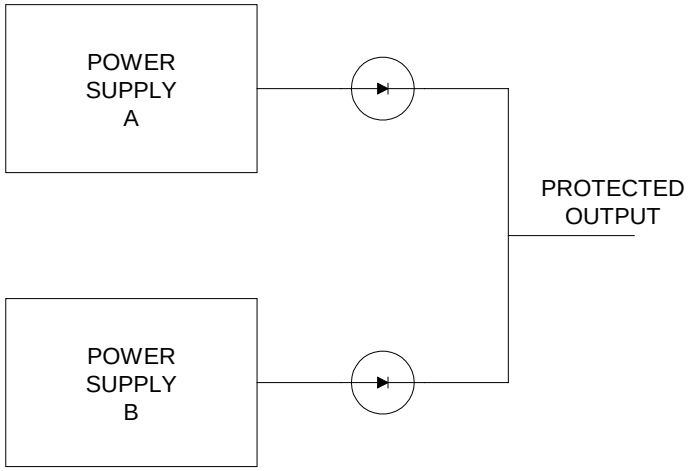


Figure A-1. Simple Parallel Redundancy

However, even in this simple arrangement, the following questions must be asked, if this architecture is to be used for a critical aircraft application:

- Are there any failure mechanisms that can cause the output voltage from one of the supplies to exceed the specified value? If so, what effect will that have on the components serviced by the supply?
- What indications of a supply failure are furnished to the flight crew and to maintenance?
- Are the individual supplies dimensioned to permanently furnish the full load current?

The more typical application of parallel redundancy involves three or more channels. The critical certification questions depend on the method for combining the output. When digital selection is used, the reliability and failure modes of the selection elements become critical issues. In the block diagram shown in figure A-2, the voter looks deceptively simple, but the implementation typically involves tens of logic gates. Separate logic may be required for 0 and 1 channel outputs because an AND gate will correctly output a 1 only when both inputs are 1, but it will output 0 for inputs of 1-0, 0-0, and 0-1.

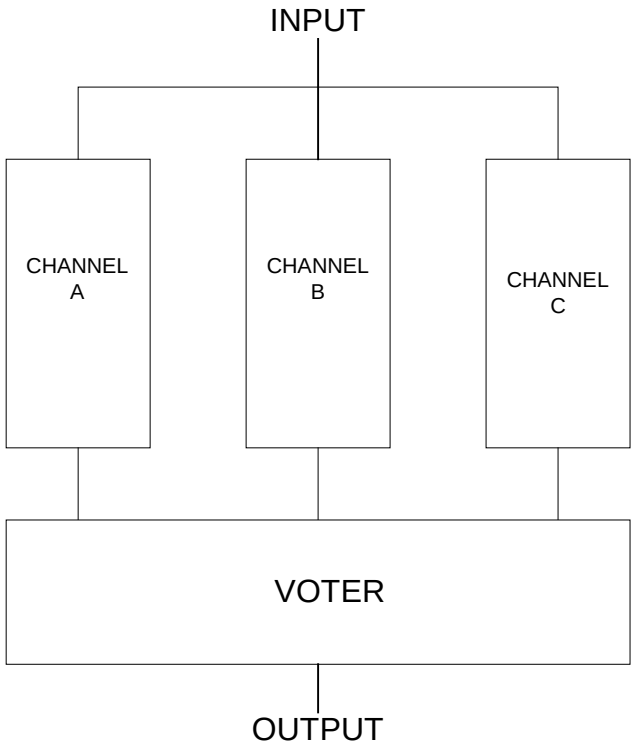


Figure A-2. Triple Parallel Redundancy

Typical certification questions include:

- Is the voter redundant or does it use other means of fault tolerance?
- Are failures in the voter components communicated to the flight crew and maintenance?
- Are channel failures communicated to the flight crew and maintenance?
- Are the channels and the voter operating on a common clock? If so, is the clock monitored? If not, what is the limit of time difference that the voter can tolerate?

This redundancy architecture does not qualify as dissimilar implementation of the channels because of unpredictable timing differences in the arrival of the channel output at the voter.

Mechanical or hydraulic output summing creates its own dependability challenges, as illustrated for two inputs in figure A-3. An equivalent structure for three inputs can be constructed by applying forces (perpendicular to the surface) at the corners of an equilateral triangle. The mechanisms are described for mechanical outputs, but essentially, the same concerns exist for hydraulic summing.

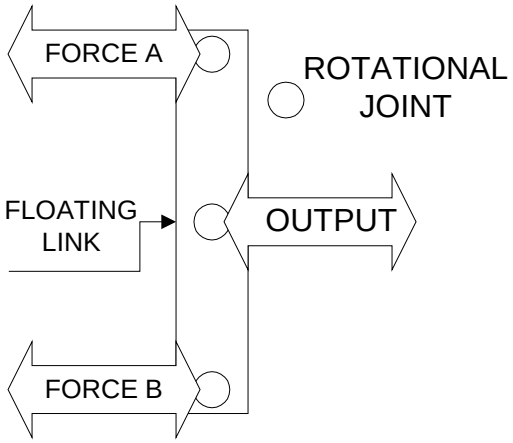


Figure A-3. Mechanical Force Summing

In normal operation, an output of one unit from each of the two channels (A and B) will result in a combined output of one unit. If one of the channels fails in the neutral position, a one-unit output from the other channel will produce a combined output of only half a unit. If one of the channels fails in a hard-over mode, that other channel can, at most, neutralize the output but cannot produce a reverse output. Channel failure is easily sensed (whenever the angle between the force inputs and the floating link deviates from 90 degrees) and can be used to counteract the loss of gain and to reset the failed output to neutral; but these features must be demonstrated to only fail under extremely improbable conditions.

Suitable certification questions are:

- Will the failure of an input be communicated to the flight crew and maintenance?
- Will the gain (output/input) be affected by failure of one of the input channels? If so, is this tolerable for the overall system performance, and if not, what corrective action is required?
- Will full bidirectional output be maintained after all possible failure modes of one channel? If not, what corrective action is required?
- What are the failure modes, their probabilities, and mitigation means for all elements of the combining mechanism?
- Will all failure modes of the combining mechanism be communicated to the flight crew and maintenance?

A.2 FAILURE DETECTION IN BACKUP REDUNDANCY.

In backup redundancy, there is a need to define a means for failure detection and a (usually separate) means for restoration of service. In parallel redundancy, failure detection is usually accomplished by comparison of outputs of the channels, and this has the advantage of very high coverage. It is very difficult to conceive of a significant failure in one of the channels that will not lead to deviation in its output from a normally functioning channel. Because of the high coverage, a comparison may also be used in switched redundancy. One such architecture is shown in figure A-4. The switch is operated by the output of the comparator (indicated by the arrow).

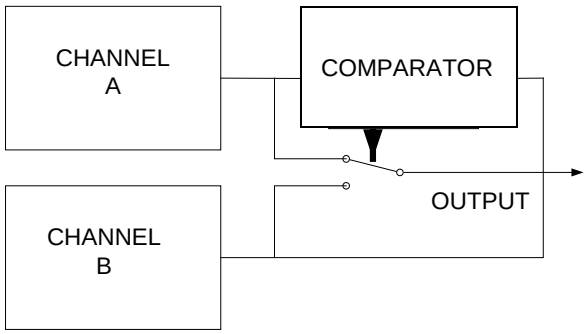


Figure A-4. Comparison in Switched Redundancy

In a two channel comparison, it is unknown which channel failed, and thus, switching between channels may not be the desired action. However, confirmation of channel failure can be obtained by other means, such as the feedback signal (discussed later) or observation of an outer loop (e.g., flight path for failure of the attitude control system). Another method of using comparison in switched redundancy is in the pair and spare configuration, as shown in figure A-5.

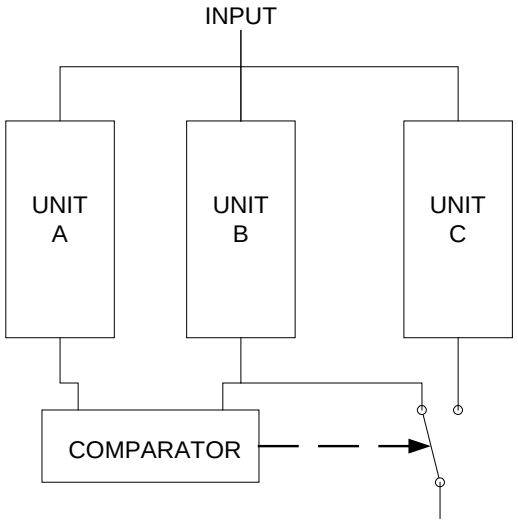


Figure A-5. Pair and Spare Redundancy

This configuration avoids the need to access external information to identify the failed channel. Assurance that unit C will be operational when needed can be obtained by periodic self-test or by switching the roles of the channels, e.g., using unit C in the place currently held by unit A.

When comparison is used for failure detection in a switched redundancy architecture the reviewers should ask the following questions:

- How likely is it that a miscompare will be observed with both units functioning correctly, e.g., because of timing problems?
- Is the comparator independent of the monitored channels?
- Are the data being compared fully representative of the output, e.g., does the comparator receive a checksum or the whole register content?
- Is there a threshold for the comparison?
- Are repeated miscompares required before switching is initiated?

An alternative failure detection mechanism is self-monitoring, which is particularly effective in closed-loop control systems, where the amplitude of the error signal can identify progressive as well as instantaneous failures. A gradual increase in the root-mean-square (rms) value of the error signal can be used as a prognostic for the effectiveness of the control system. The rms value of the error signal will increase due to a reduction in gain of the sensors, electronic components of the system, an increase in friction of mechanical actuators or of leakage, or blocking of hydraulic actuators. An example of this type of self-monitoring for a channel of an attitude control system is shown in figure A-6.

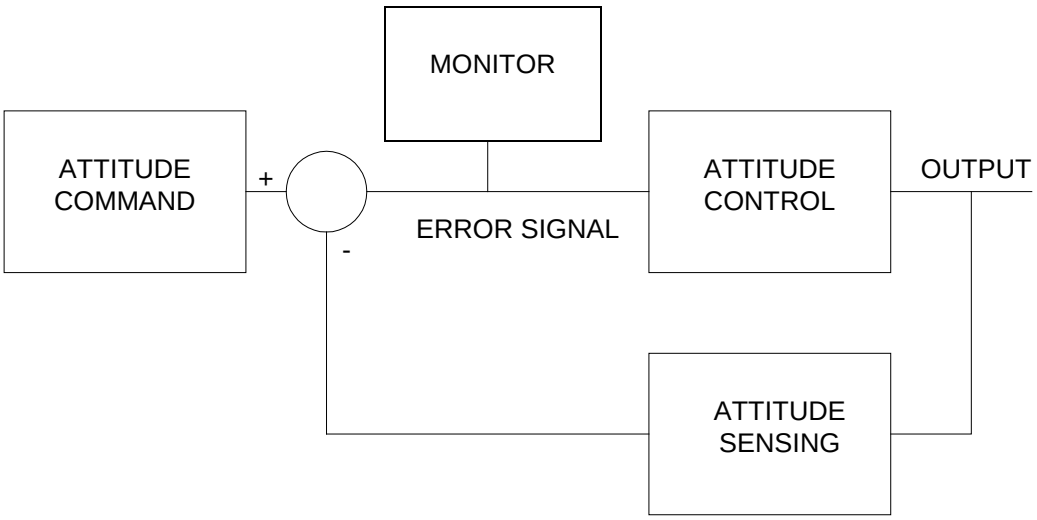


Figure A-6. Self-Monitoring in an Attitude Control Channel

Certification questions for such a configuration include:

- How will failures in the monitor be detected?
- Are there failure mechanisms that can affect the monitor and the monitored system at the same time?
- What is the threshold for declaring a channel inoperative? Does it vary with flight condition?
- Are there failure conditions that will not be sensed by the monitor?

Self-monitoring of individual channels can be combined with interchannel comparison, e.g., to resolve the ambiguity of a miscompare in figure A-4. Another method of increasing the effectiveness of error detection is to provide self-monitoring at an outer loop, e. g., implementing the self-monitoring shown in figure A-6 at the flight path level.

Some flight-critical functions that are not implemented as closed-loop control systems can still be supplied with effective self-monitoring. Examples are the landing gear and flap or slat controls. The operational element in these is typically a solenoid-operated hydraulic valve. Semiconductor solenoid drivers with extensive self-monitoring provision have been available for several years. A representative solenoid driver is the SGS Thomson L294, a schematic of which is shown in figure A-7.

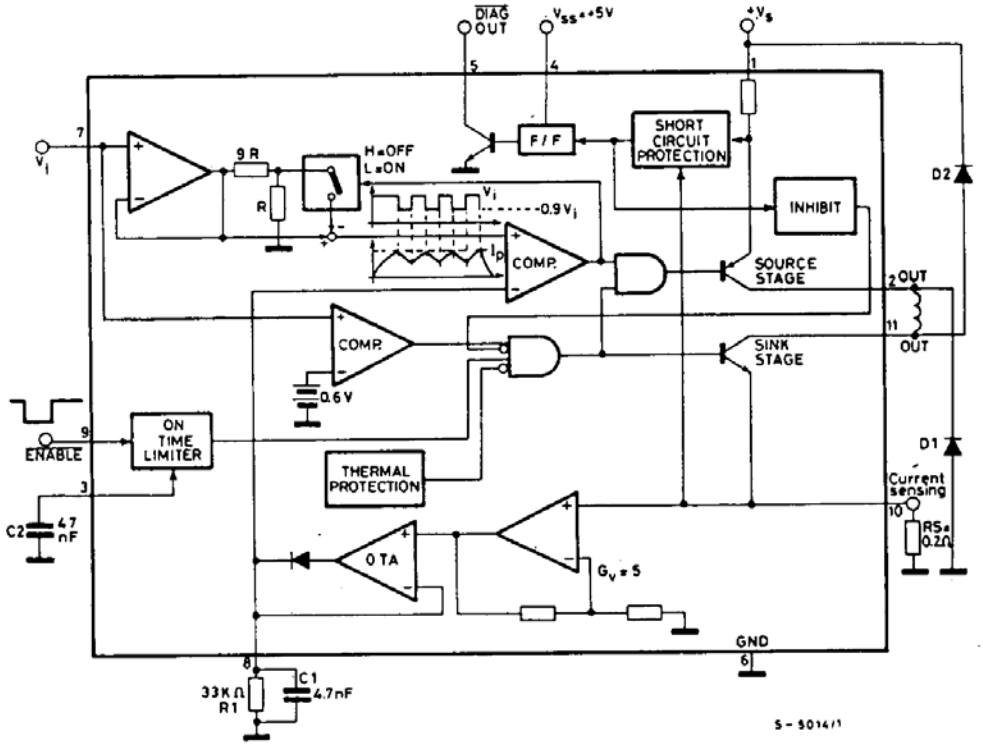


Figure A-7. Self-Monitoring in a Solenoid Driver

An important provision for detecting failures in the device is the thermal protection will be triggered by practically all the common semiconductor failure mechanisms. But the L294 also incorporates several other features to monitor the health of the overall system that it services. These include the on-time limiter, the short-circuit protection for the solenoid and the external connections to it, and the current sensing with an adjustable threshold. The current sensing also is best utilized by external monitoring (e.g., in a maintenance computer) for long term trends that can be used for prognostic purposes. Upon detection of any of these failures the function controlled by this solenoid can be transferred to a backup system.

Certification questions for this type of self-monitoring include:

- Are all available failure detection capabilities used?
- Do thresholds provide timely detection of failures without causing nuisance trips?
- How is the operational status of the monitoring features verified?

A.3 TRANSFER OF CONTROL IN BACKUP REDUNDANCY.

The critical features for transfer of control are

- the functioning of the transfer elements, including connections to associated subsystems.
- the functioning of the backup element.
- purging of possibly corrupted data and substitution of valid data.
- indications of the transfer in the cockpit and to maintenance.

Suitable certification questions for the functioning of the transfer elements are:

- How often is the operation of transfer elements (i.e., solid state or electromechanical relays, switches, and electro-hydraulic components) checked in normal operation?

- Do these checks yield information on degradation? (i.e., transfer time, contact resistance, and noise in data lines)

- Is there an indication that the transfer is complete?

- How are associated subsystems affected by the transfer? (i.e., is the autopilot affected when an air data comes from a backup computer?)

The design assurance for the functioning of the backup element can be explored with the following questions:

a. How often are self-test provisions of the backup element exercised in normal operation?
b. What operational areas of the backup element are not covered by the self-test provisions?
c. By what means are the results of self-test analyzed for signs of degradation?
d. Is the functioning of the backup element tested after switchover?
e. Is partial or complete failure of the switchover indicated in the cockpit?
f. What are the recovery steps from partial or complete failure?

The following questions explore the purging of possibly invalid data and the substitution of valid data:

a. Does the function retain data from previous cycles? (i.e., for evaluating trends, for computing increments from most recent cycle, or for establishing maxima or minima)

b. Does the function send results to associated functions or programs? (Example airspeed input to ground speed computation or to low-speed alarm)

c. How are the clients in a or b notified of possibly invalid data?

d. How are replacement data furnished to the clients in a or b?

e. How is completed data purging and substitution verified?

The availability of required information to the flight crew can be investigated with the following questions:

a. Does operation with the backup element involve limitations in flight or operational envelope? (Examples: restrictions or airspeed or inability to perform Category III landings)

b. Does the operation with the backup element change monitoring provisions or alarm indications? (Examples: angle-of-attack protection and engine-indicating and crew-alerting system (EICAS) messages)

c. How are changes in a or b indicated to the flight crew, and is acknowledgement required?

d. What are required crew actions in the case of an incomplete or suspect switchover?

A.4 REDUNDANCY WITH SIMILAR ELEMENTS.

The key concern in redundancy with similar elements is the existence of common cause or otherwise correlated failure modes. The commonality can arise from design deficiencies, improperly inspected parts lots, places where electrical or hydraulic outputs from different channels necessarily have to be connected, or from environmental effects. The use of identical software is also an area of great concern. These areas are covered by Federal Aviation Administration and industry documents, particularly SAE ARP 4754.

Most of the questions dealing with self-test of the redundant elements and purging of data in the previous section are also relevant here.

Questions that explore problems in redundancy with similar elements (and conformance with ARP 4754) are:

a. How is the absence of design failure modes common to the similar channels documented?

b. By what measures are components protected against correlated failures due to bad part lots?

c. Are channels spatially separated? (zonal analysis)

d. Where outputs or inputs cannot be spatially separated, how are single failures prevented from affecting more than the output of a single channel?

e. How are failures in common electric supplies, timing, and other data sources prevented from affecting more than a single channel?

f. Have tests been conducted to show that allowable radiation levels cannot affect more than a single channel? (These tests have to represent the location of the components in the aircraft.)

g. Is the software fully compliant with DO-178B (or successor documents)?

A.5 REDUNDANCY WITH DIVERSE ELEMENTS (INCLUDING ANALYTIC REDUNDANCY).

Redundancy with dissimilar elements involves a high probability that the performance with the backup channel will not be identical to the primary channel. When analytic redundancy is used this becomes near certainty. These forms of redundancy largely avoid common-cause failures due to design deficiencies and bad part lots, but are still subject to correlated failures due to zonal and environmental causes.

All the certification questions cited in section A.3 for self-testing of the redundant elements and purging of suspect data continue to be relevant here.

The following questions explore the specific concerns:

a. Does operation after the loss of a redundant channel involve limitations in flight or operational envelope? (Examples: restrictions or airspeed or inability to perform Category III landings)

b. Does the operation after loss of a redundant channel change monitoring provisions or alarm indications? (Examples: angle-of-attack protection and EICAS messages)

c. How are changes in a or b indicated to the flight crew, and is acknowledgement required?

d. What are required crew actions in the case of an incomplete or suspect response to loss of a redundant channel? Are channels spatially separated? (Zonal analysis)

e. Where outputs or inputs cannot be spatially separated, how are single failures prevented from affecting more than the output of one channel?

f. How are failures in common electric supplies, timing, and other data sources prevented from affecting more than a single channel?

g. Have tests been conducted to show that allowable radiation levels cannot affect more than in single channel? (These tests have to represent the location of the components in the aircraft)

h. Is the software fully compliant with DO-178B (or successor documents)?