AGARD-AR-274

AGARD

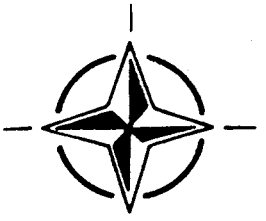
ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT
7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

AGARD ADVISORY REPORT 274

Validation of Flight Critical Control Systems

(Validation des Fonctions Critiques pour le Pilotage)

This report has been prepared as a summary of the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD.



NORTH ATLANTIC TREATY ORGANIZATION

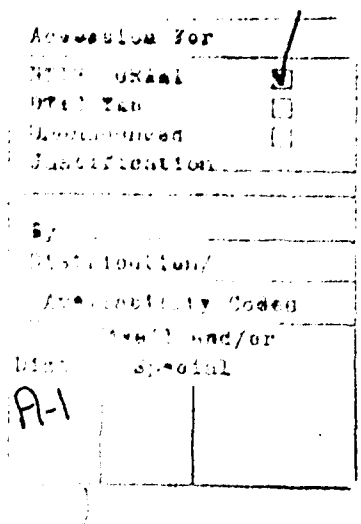
Published December 1991

Distribution and Availability on Back Cover

AGARD-AR-274

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT
7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE



AGARD ADVISORY REPORT 274

Validation of Flight Critical Control Systems

(Validation des Fonctions Critiques pour le Pilotage)

Edited by

Mr Gordon Belcher, Mr Duncan E. McIver
and **Mr Kenneth J. Szalai**

This report has been prepared as a summary of the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD.

North Atlantic Treaty Organization
Organisation du Traité de l'Atlantique Nord

The Mission of AGARD

According to its Charter, the mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community;
- Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Exchange of scientific and technical information;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

Published December 1991

ISBN 92-835-0650-2

Printed by Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ

Preface

This report has been prepared as a summary of the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD. The terms of reference were approved by the National Delegates Board of AGARD and the objectives of the Working Group were:

(1) To provide guidance to those concerned in the Flight Critical Control System (FCCS) validation, namely system designers and certification authorities.

(2) To identify the areas of research which need to be explored to enable validation of the next generation of FCCS.

The Working Group tried to review all flight critical control system validation activities which had been completed or were under active consideration, in Europe and the United States. Detailed technical presentations of these relevant examples were made to the Working Group for their deliberation. In addition, emerging technologies which could have a significant impact on validation of future FCCS, were discussed at length by the members of the Working Group.

The Working Group started work in the fall of 1986 and met at six month intervals up to October 1989. The Group was composed of members from France, Germany, Italy, the United Kingdom, and the United States, all of whom were expert in guidance and control, and the validation of FCCS. This report represents the consensus view of the Group, but it should not be construed as representing the views or policies of any of the nations, organizations, or individuals represented on the Working Group.

Final editing of the report took place during the last half of 1989 and during 1990 and 1991. The final report was prepared with the support of the NASA Langley Research Center in the United States, with essential help from Mrs Carolyn Wilt, of the Langley Office of Director for Flight Systems.

Préface

Ce rapport est un résumé des déliberations du groupe de travail No. 09 du Panel AGARD du guidage et du pilotage. Le mandat de ce groupe a été approuvé par le Conseil des délégués nationaux de l'AGARD (NDB) et ses objectifs ont été les suivants:

(1) De fournir des conseils à la communauté de la validation des systèmes de commandes de vol critiques (FCCS), c'est à dire, aux ingénieurs systèmes et aux autorités de certification.

(2) D'identifier les voies de recherche à suivre pour permettre la validation de la prochaine génération de FCCS.

Le groupe de travail s'est donné comme objectif de passer en revue toutes les activités connues dans le domaine des systèmes de commandes de vol critiques, qu'il s'agisse de travaux déjà accomplis ou de projets à l'étude, et ceci en Europe et aux Etats-Unis. Des présentations techniques détaillées de ces exemples pertinents ont été données au groupe pour leur considération. En outre, des technologies naissantes, susceptibles d'avoir un impact sensible sur la validation de futurs FCCS ont été discutées dans le détail par les membres du groupe de travail.

Le groupe s'est réuni pour la première fois à l'automne de l'année 1986 et ensuite à des intervalles de six mois jusqu'en octobre 1989. Il a été composé de membres de la France, de l'Allemagne, de l'Italie, du Royaume-Uni et des Etats-Unis; tous experts dans le domaine du guidage et du pilotage et de la validation des FCCS.

Ce rapport, s'il représente le consensus d'opinion du groupe, ne doit en aucun cas être interprété comme la représentation des opinions ou des politiques d'un quelconque pays, organisme ou individu, membre du groupe.

Les travaux de mise en forme définitive du rapport se sont déroulés pendant la deuxième semestre de 1989 et courant 1990–1991. Le rapport définitif a été élaboré avec le soutien du NASA Langley Research Center aux Etats-Unis, avec notamment la coopération de Mme Carolyn Wilt, du bureau du Directeur systèmes de vol de Langley.

Working Group 09 Membership

FRANCE

Mr Luc Baron
Sextant Avionique
Centre Avionique de Toulouse
15 Avenue Didier Daurat
BP 43
F-31708 Blagnac Cedex
France

Mr Alain Coupier
Labo Genie Logiciel
Ceat
23 Avenue Gaillaumet
F-31056 Toulouse Cedex
France

GERMANY

Mr H.Juergen Kaul
Flugzeuge und Hubschrauber
MBB GmbH — Abt. FE 34
P.O. Box 80 11 60
D-8000 München 80
Germany

Mr Günter Mansfeld
DLR EV — Dept 112-14
Institut für Flugführung
Postfach 32 67
D-3300 Braunschweig
Germany

ITALY

Dr Luciano Rovere
Aeritalia Saipa
Gruppo Velceli
Combattimento
Direzone Technica
Commandi di Volo
Corso Marche 41
Italy

UNITED KINGDOM

Mr Gordon Belcher
Group Technical Manager
GEC Avionics Limited
Airport Works
Rochester, Kent ME1 2XX
United Kingdom

Mr David J. Walker
Research Manager
British Aerospace Plc
Military Aircraft Division
Brough
North Humberside HU15 1EQ
United Kingdom

UNITED STATES

Dr Jeremiah F. Creedon
NASA Langley Research Center
Mail Stop 113
Hampton, Virginia 23665-5225
United States

Mr Robert D. Evans
Director, Research Projects Division
6510th Test Wing/DOR
Edwards AFB, California
United States

Mr Pio de Fio
Director, Software & Systems Tech.
Sparta, Incorporated
23041 De La Carlota, Suite 400
Laguna Hills, California 92653-1507
United States

Mr Duncan E. McIver
104 Montague Circle
Williamsburg, Virginia 23185
United States

Mr Ray Hood
NASA Headquarters
600 Independence Avenue, SW
Washington, DC 20546
United States

Mr James K. Ramage
Chief, Advanced Development
Branch
Wright Research & Development
Center
WRDC/FIGX
Wright-Patterson AFB
Ohio 45433-6553
United States

Mr Kenneth J. Szalai
Chief, Research Engineering Division
NASA Ames-Dryden OF
P.O. Box 273
Edwards AFB
California 93523-5000
United States

Mr John Watson
General Dynamics
P.O. Box 748
Mail Zone *9310*
Fort Worth, Texas 76101
United States

Contents

List of Acronyms

ACT	Active Control Technology
ADOCS	Army Defense Operations and Control System
ADIRS	Air Data & Inertial Reference Systems
AFB	Air Force Base
AFTI	Advanced Fighter Technology Integration
AGARD	Advisory Group for Aerospace Research and Development
ATP	Automatic Test Procedure
BIT	Built In Test
CAD	Computer Aided Design
CAS	Control Augmentation System
CDR	Critical Design Review
CI	Configuration Item
CR	Change Requests
CPU	Central Processing Unit
CSAS	Command Stability Augmentation System
DDP	Declaration of Design and Performance
DFBW	Digital Fly-by-Wire
DOD	Department of Defense
ELAC	Elevator & Aileron Computers
EMI	Electromagnetic Interference
FAC	Flight Augmentation Computers
FCA	Functional Configuration Audit
FBW	Fly-by-Wire
FCC	Flight Control Computer
FCD-FBW	Flight Critical Digital Fly-byWire
FCCS	Flight Critical Control System
FCF	Functional Check Flight
FCS	Flight Control System
FDL	Functional Descriptive Language
F/H	Failures per Hour
FMEA	Failure Modes and Effects Analysis
FRR	Flight Readiness Review
FTA	Fault Tree Analysis
FSD	Full Scale Development
FTP	Flight Test Program
HOL	Higher Order Language
HQDT	Handling Qualities During Tracking
HZ	Hertz

IBU	Independent Back-up
ICD	Interface Control Documnent
I/O	Input/Output
ISE	Integrated Software Environment
JAR	Joint Airworthiness Environment
LVDT	Linear Variable Differential Transformer
MIL-STD	Military Standard
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NOE	Nap-of-the-Earth
PCA	Physical Configuration Audit
PDL	Programming Design Language
PDR	Preliminary Design Review
PRR	Production Readiness Review
PTL	Program Target Language
RFP	Request for Proposal
SAS	Stability Augmentation System
SB	Service Bulletin
S&C	Stability and Control
SCFCS	Safety Critical Flight Control System
SCP	System Concept Paper
SDR	System Design Review
SEC	Spoiler and Elevator Computer
SIL	Service Information Letter
SIT	System Integration Test
SOA	State -of-the-Art
SOW	Statement of Work
SRR	System Requirements Review
SSA	System Safety Analysis
TEMP	Test and Evaluation Master Plan
VSLI	Very Large Scale Integration
VTOL	Vertical Take-Off and Landing
WG	Working Group

CHAPTER 1

INTRODUCTION

1.1 Background

Automation and digital electronic control systems are being used in ever increasing levels in aircraft flight control systems. The benefits of these advanced control systems have been dramatic, contributing to major improvements in aircraft performance and mission effectiveness. Full time digital fly-by-wire control systems with active controls of unstable aircraft modes have reached the point of being essential to the aircraft's safe operation. The safety-critical nature of these modern flight control systems requires an extremely high level of reliability and integrity, equivalent to that of the basic aircraft structure itself.

Although the first manned spacecraft and earliest experimental aircraft digital fly-by-wire control systems used single thread elements[1], today's flight control systems employ extensive redundancy for any element whose failure could jeopardize vehicle safety. Triplex or quadruplex redundancy is often used for sensors, computers, actuators, and data communication links between them to provide continued operation in the event of failures.

Control algorithms have grown in complexity as well, often involving several modes of control, with complex gain scheduling and interfaces with various other aircraft systems and subsystems. Control laws are used to provide artificial longitudinal stability where the aircraft's stability margins have been relaxed or designed to be negative to gain maneuvering or cruise performance advantages. In some aircraft, the unaugmented divergence rate is so high, that total loss of the electronic flight controls would result in vehicle dynamics which could compromise the crew's ability to egress.

The multiple redundant systems and the sophisticated control laws have resulted in a complex and time-consuming design, development and qualification process. Entire AGARD symposia and lecture series' have been devoted to the design and development aspects of advanced flight control systems[2-5]. The qualification process, however, has also grown to be a major critical activity in the overall process of achieving a safe and effective production flight control system, and has a technology aspect of its own. This has been recognized in a recent AGARD Working Group effort addressing the verification and validation of real time software for flight control systems[6].

The qualification of the entire flight-critical flight control system, including both hardware and software, presents some difficult challenges. Hardware components can usually be tested quite readily for functionality and performance in a variety of environmental conditions, and in many cases, sufficient supplier test data exists to provide credible failure rate data. However, it is not sufficient to combine all of these data in any credible manner to establish the reliability or failure tolerance of the overall flight control system, given the usually complex interaction of all of the components and subsystems, and the action of the software in managing all of the elements in the system.

It is furthermore impractical, or impossible to test directly, the entire flight control system reliability or failure tolerance, which is often expressed in terms of the "probability of loss of control per hour". The design targets are often so small for this probability (10^{-5} to 10^{-9} per hour), that direct testing cannot be accomplished in any credible manner, due to the length of time it would take to acquire statistical evidence of the system's characteristics.

Although analyses do play a very important part in the design, development, qualification, and certification process, analytical techniques alone are not sufficient to to assure that the extremely small control loss probabilities have actually been achieved in a real design.

The practical qualification process must, then, be some combination of analysis, component testing,

subsystem testing, and integrated system testing, involving both the hardware and software in some logical process, all designed to assure the company, the customer, and the independent certifying authorities, that the flight control system is both effective and safe. At the present time, there is no universally accepted procedure for qualifying and certifying a flight critical flight control system. Test processes, procedures, and philosophies differ among airframe manufacturers, suppliers, and customers. There is little conformity in the certification requirements, except at a high level, and the companies and certifying authorities are both on a learning curve in this area.

The cost and impact of the qualification process is of such significance, that advances in the technology of system qualification are expected to have a strong positive influence on vehicle safety, life cycle cost, and program schedules. Yet these processes have developed in more or less an ad hoc manner. It was these facts that led AGARD to establish a Working Group to closely examine the qualification process for flight critical flight control systems with the objective of improving the process and providing some guidance for the future.

1.2 Scope

The scope of this report, as it reflects the scope of Working Group 09 itself, is the assessment and projection of the state of the art of the qualification process for flight critical flight control systems, with regard to flight safety. The establishment of aircraft performance or mission suitability is not included, except as it directly has a bearing on flight safety.

The focus of this report is on the validation process. Recognizing that verification can be considered to be a sub-process to validation, this report uses the term "validation" in its broadest sense, which includes the necessary verification steps that take place within it.

Validation criteria for flight equipment and systems are based on the impact that the loss, or malfunction, of such equipment and systems have on flight safety. The determination of level of criticality is achieved by means such as design description, analyses, simulations, similarity and other appropriate methods. The criticality levels are most effectively negotiated between manufacturers and relevant regulatory agencies at the earliest possible time, because they strongly affect the entire development process including development and test methods; tools, techniques, and environments; documentation requirements; and user operation and maintenance requirements.

The subject of this report is the validation process of flight systems which are critical to the control of the aircraft. Critical systems can include, for example, engine controls, large authority autopilots and weapon systems, and full authority active flight control systems. This document, however, concentrates on the latter and although addressing integration with other critical systems it assumes that there is a direct application of the the principles involved.

1.3 Working Group Objectives

The objectives of the Working Group were established as follows:

1.3.1 Guidance on the Validation Process

To provide guidance to those concerned in the validation of flight critical control systems, namely system designers, aircraft designers and certification authorities.

To achieve this objective, the following aspects are addressed and reported:

(a) The phasing of the validation process and its relationship to systems development.

(b) The structure for accomplishing the definition of the requirements, the testing for compliance and the formal acceptance that the requirements have been met.

(c) The techniques/methods appropriate for each phase of the validation process. This aspect will include guidance on the coverage and depth of the techniques/methods.

(d) Systems design features which facilitate validation within practical constraints.

1.3.2 Future Validation Processes and Needs

To identify the ares of research which need to be explored to enable validation of the next generation of flight critical control systems.

To achieve this objective, the following aspects are addressed and reported:

(f) The range aircraft systems and the technologies to be employed in the next fifteen years which are likely to have a major impact on validation methods.

(g) The new validation techniques which will have to be developed to allow these systems/technologies to be used safely.

1.4 Organization of the Report

This report is partitioned to highlight the validation process, but it is not possible to separate validation from the design and development process itself. Therefore, the report first describes the state of the art in flight critical flight control systems in Chapter 2 to provide a common basis for understanding the validation requirements. This is done by way of four examples which illustrate the range of design variables in modern flight control systems. It is these types of systems which provide the structure for the assessment of the state of the art in validation.

In Chapter 3, a top level generic development and validation sequence is described, as well as a description of the interrelationship of the vehicle development, systems development, and validation process. This serves as both a summary of the process and also as a guide to the detailed description of the processes contained in Chapter 4. In Chapter 5, a critical assessment is provided of the principal elements of the validation process. Chapter 6 contains the projection of trends in flight control design over the next 15 years, along with the expected impact on the validation process. Chapter 7 contains the emerging tools and techniques that will be needed to improve the validation process for the current generation of flight control systems as well as for those projected in Chapter 6. Chapter 8 contains conclusions and recommendations which are intended to provide guidance for future flight control design, development and validation technology developments.

REFERENCES

[1] Description and Flight Test Results of the NASA F-8 Digital Fly-by-Wire Control System, NASA TN D-7843, February, 1975

[2] Advances in Guidance and Control Systems Technology AGARD-CP-411, October, 1986.

[3] Fault Tolerant Design Considerations and Methods for Guidance and Control Systems, AGARD-AG-289, July, 1987.

[4] Computing Systems Configurations for Highly Integrated Guidance and Control Systems, AGARD-LS-158, 1988.

[5] Fault-Tolerant Hardware/Software Architecture for Flight Critical Functions, AGARD-LS-143, 1985

[6] The Implications of Using Integrated Software Support Environment for Design of Guidance and Control Systems Software. AGARD Advisory Support N. 229 February 1990.

CHAPTER 2

STATE OF THE ART IN FLIGHT CRITICAL FLIGHT CONTROL SYSTEMS

2.1 Functional Requirements

Functional requirements and the continuous evolution of FCCS automatic control system architectures is shown in Figure 2.1[1]. Limited authority, analog, stability augmentation systems were developed during the 1950's; an example is the F-104. These were followed by the development of flight critical, analog Fly-By-Wire (FBW) systems, which began during the early 1970's; examples are the F-16 and Mirage 2000. The development of digital FBW (DFBW) systems started in the early '70's, and is still evolving. Examples are: NASA F-8 DFBW, the Jaguar DFBW and the F-16C/D. The trend is clearly established towards systems which are increasingly complex and which include more flight critical functions.

Architectural complexity is increasing due to the increased functional criticality and the resulting need for satisfying stringent reliability, availability and fault tolerance requirements. Moreover, the flight critical control functions which DFBW systems are asked to provide, typically require frequent inputs to the control effecters, which cannot be effectively and consistently provided by the pilot, during some, or all flight regimes and conditions. Examples of this type of flight critical function include the integrated engine and nozzle control of Vertical Take Off and Landing (VTOL) aircraft, and the functions commonly referred to as active controls technology (ACT) functions, which are discussed in the next paragraph. Additional increased complexity results from the requirement of integrating many existing and new functions for improving performance, for extending the flight envelope, and for decreasing pilot workload. Examples of functions which are being considered for integration include flight control, propulsion control, weapons control, guidance, navigation, flight management, thermal management, etc.

The design of many high performance aircraft rely on augmentation systems for providing some of the safety margins traditionally provided by inherent aerodynamic stability and the structural strength and stiffness of the basic airframe. During the design cycle of the aircraft, the availability of ACT is taken into account to relax the constraints in the aerodynamics, structures and propulsion systems, while achieving the same effective margins with the active system. Typical applications of ACT include: load alleviation and structural mode control, relaxed stability margins, aerodynamic configuration management, maneuver enhancing or limiting and other complex functions. Examples of aircraft which use ACT systems are: a) the Boeing B-52 (G and H) gust loads alleviation for increasing the wing structural fatigue life; b) the F-16, and the X-29 stability augmentation systems for providing stability and enhancing maneuvering and cruise performance; c) the Lockheed L-1011-500 maneuver load control to extend wing span without structural changes of the wing; and d) the AIRBUS A-320 envelope limiting system providing protection from intrusions into unsafe regions of the flight envelope. Clearly the stability augmentation system of an aircraft which is as inherently unstable as the X-29 is a critical function. In some cases, other ACT functions may also be flight critical.

The boundaries of flight critical control functions have also grown beyond classical control systems, especially in the case of military applications. Flight control functions and avionics sensory functions are integrated in common architectures to satisfy the mission requirements of advanced military aircraft. Examples are helicopter nap-of-the-earth (NOE) and high speed terrain-following/terrain avoidance missions. These missions require that sensory functions such as obstacle detection, terrain data, radar altitude, target acquisition and tracking, and inertial reference system data, be carefully integrated with flight path control functions. These types of systems are evolving to flight critical as a result of increasingly stringent mission requirements.

2.2 State of the Art Digital Flight Control System Configurations

Critical components of DFBW flight control systems include the primary sensors, the digital processors, the data distribution system, and the actuation systems of the primary control surfaces. The safety and fault tolerance requirements of a Flight Control System (FCS) configuration can be met by using several levels of redundancy more efficiently than by applying the same level of redundancy throughout the configuration. Several examples of FCS configurations which use different levels of redundancy are discussed later in this chapter.

A large number of architectural options, which have been designed to satisfy the same or similar safety and fault tolerance requirements, are available for use by system designers. Examples of redundant flight control computer configurations which have been designed, validated, and flight demonstrated in recent years are shown in Table 2.1. [2]

Aircraft	Primary		Back-Up			Function
	Maturity Config	Type	Config.	Type		
F-18	Quad	Digital	Simplex	Mech.	CAS,Pitch	Production
ADOCS	Dual-trplx Experimental	"	"	N/A	N/A	FBW FCS
AFTI/F-16	Triplex	"	Triplex	Analog	"	Tech. Demo
X-29	Triplex	"	Triplex	Analog	"	Experimental
A310	Dual	"	N/A	N/A	Spoilers	Production
JAS-39	Triplex	"	"	"	FCS	Prototype
LAVI	Triplex	"	Simplex	Analog	"	Prototype
Jaguar	Quad	"	N/A	N/A	"	Experimental
DC9-80	Simplex	"	"	"	Autoland	Production
L1011	Dual-dual	"	"	"	Autopilot	Production
B767	Triplex	"	"		"	Production
F-16 C/D	Quad	"	Quad	Digital	FCS/Fire Con.	Production
F-8 DFBW	Triplex	"	Triplex	Analog	FCS	Experimental

Table 2.1. Redundant flight control system Configurations

The variety of computer system configurations which have been implemented reflects the different requirements of each application and also the designer's choice from the many available design options. In general the level of hardware redundancy increases as a function of the fault tolerance requirements. Quad configurations are often used in the case of applications which are flight critical during the entire duration of flight. Lower levels of redundancy have been used in the case of applications which: a) are critical only in a limited portion of the mission, like automatic landing; or b) have a lesser degree of criticality. The selection between two different configurations which provide the same level of fault tolerance, like dual-dual and triplex configurations, is made based on considerations such as commonality with existing equipment, past experience, production and maintenance issues, and economic factors. The process of selecting from competing architectures results in difficult compromises among numerous and sometimes conflicting requirements. Among the requirements which must be considered are: safety and reliability; weight, volume and power; life cycle cost; maintainability; and survivability.

2.2.1 Representative Digital Flight Control Systems

In this section, five advanced digital flight control system configurations, which are representative of the state of the art, are described in some detail. These examples are: The F-16 C/D, F-18, the X-29, the FBW Jaguar, and the Airbus A-320.

2.2.1.1 F-16 C/D Digital flight control system

The F-16 C/D flight control system is a digital mechanization of the existing F-16A/B analog implementation. This system has fail-operational/fail-operational capabilities allowing it to sustain two

similar failures and still provide full performance. This is achieved by a quad architecture of critical sensors and digital processors, and a redundant actuation system.

The flight control system includes four Mil Std 1750A processors and an Independent Back-Up (IBU) system, also quad, which uses the primary system processors and independently developed software. The IBU provides protection against possible generic software errors. Reversion to the IBU is made as a result of either automatically detected failures or pilot selection.

The configuration is asynchronous. Each primary channel uses 24K words (16 bits/word). The software is coded in Jovial J73, using floating point arithmetic. Each back-up channel uses 4K words and is coded in Assembly language. Reversion to the back-up software is made by switching the Computer Processing Unit (CPU) to the memory banks of the back-up system. The primary flight software has a multi-rate execution structure. The basic rate is 64 Hz, which corresponds to a frame time of 15.6 msec. Comparison monitors among the four primary channels are the primary failure detection mechanisms. The thresholds for the trip levels are set as a function of the rate of change of the variable. A schematic of the digital flight control system is shown in Figures 2.2 and 2.3.

The flight critical sensors (rate gyros and normal acceleration), and the pilot stick sensors are also quad redundant. The sensor inputs are voted in software. Output commands to the servo-actuators are voted. Fail-operational/Fail-operational capabilities are also provided for the servo actuators. The proven electro-hydraulic servo-actuators of the analog F-16 have been retained in the F-16 C/D.

2.2.1.2 F-18 Digital flight control system

The F-18 FCS is a digitally mechanized quadruplex fly-by-wire system providing stability, control and autopilot functions, and interfaces with many of the highly integrated avionic systems through a Mil Std multiplex data bus. A functional diagram of the flight control system is shown in Fig. 2.4[3].

The primary control law computations are performed by four digital computers operating in parallel. Redundancy provides fail-op/fail-op capabilities. A mechanical back-up system is provided to the stabilator surfaces for pitch and roll control. An unaugmented, analog back-up system is provided for roll and yaw control, through aileron and rudder surfaces. The flight control system used four General Electric 701 microprogrammed, general purpose, 16 bit processors. The operational programs were written in Jovial 73 using fixed point arithmetic. The Vax hosted software development environment which was used is the same as that used for the F-16C/D. The design was documented using a programming design language (PDL). The software is highly modular and static module testing was performed prior to integration and real time testing. The operational programs were identical in all four channels, which were frame synchronized by an executive, which also scheduled all tasks at computation iteration rates of 80, 40, 20, and 10 per second.

It is interesting to note that: a) the control laws require about 22% of the memory; b) preflight and in-flight Built-In-Test (BIT), the largest function, requires 42% of the memory; and c) Input/Output (I/O) processing and redundancy management requires 18% of the memory. This data distribution is rather typical and will be discussed further in following paragraphs.

The sensors have a level of redundancy proportional to the criticality. Rate gyros and accelerometers are dual. The electrical redundancy is quad. Control stick and rudder pedal displacement sensors are simplex, but they too have quad electrical redundancy. Other non critical sensors have a lower level of redundancy.

The redundancy of the actuation systems set by the criticality of each control function [4](Ref. 3). The most critical controls, for which aerodynamic redundancy is not available, are the stabilators and the trailing edge flaps. The redundancy of the stabilator actuator is quad electrical and dual hydraulic. A simplex mechanical back-up control system is also provided, in the case of multiple electrical failures. The hardware redundancy requirements decrease in the case of control functions which have inherent aerodynamic redundancy, like the ailerons.

2.2.1.3 X-29 flight control system

The Grumman Aerospace X-29 advanced technology demonstrator aircraft has an inherent 35% negative static stability margin. As a result, the flight control system which is required to produce

acceptable handling qualities, is flight critical throughout the flight envelope. The flight control system was developed and built by a Grumman/Honeywell team and implemented with HDP-5301 processors. The research objectives of the X-29 program are focused on basic aerodynamic and control technology, rather than fault tolerant flight control system architectures. As a result the X-29 FCS configuration includes some ad-hoc solutions which are not well suited for a production system. The X-29 FCS does, however, provide the very high level of reliability and fault tolerance that such a flight critical system requires. It includes a primary triple redundant digital flight control system that used a majority vote technique to detect and isolate a faulty channel. Each flight control system channel has two digital processors; one control law processor, which uses floating point arithmetic; and one I/O processor which uses integer arithmetic. This configuration was selected to satisfy the tight execution time constraints, which could not be met by a single processor configuration. Both processors are coded in assembly language for optimizing execution speed. If an entire sensor set fails, a reversion mode can be selected which only uses a minimum number of sensors. If two or more of the digital channels should fail, the pilot can select a triple redundant analog system. In summary, the primary flight control system has two triplex reversion systems. One of these is digital, which is selectable in the case of some sensor set failures, and the other is analog. A schematic of the flight control system configuration is shown in Fig. 2.5.

The basic triplex configuration of the flight processors was also adopted for the critical sensors and actuators. Critical sensors include rate gyros and accelerometers (three axes), and the pilot control stick displacements.

Software specifications were first developed by Grumman and by Honeywell. Honeywell specifications included the redundancy management and mode selection, in addition to the control laws specified by Grumman. Honeywell specifications were written first in English and then in structured PDL. Software is highly modular. Module testing was performed prior to module integration and software/hardware integration.

2.2.1.4 Jaguar FCS

The Jaguar Integrated Flight Control System is a full authority, quadruplex DFBW system. The overall system architecture is shown in Fig. 2.6 [5].

The critical sensors are the pilot control stick displacements, rate gyros, and pitch and yaw trim. These are all quad. Other non-critical sensors are either triplex, duplex or simplex.

The primary control surfaces are: left and right tail plane, rudder, and left and right spoilers. The configuration of the actuation system of the primary control surfaces is dual-triplex. For each control surface, six independent electrical drive signals are used to drive six control valves which, in groups of three, drive a dual tandem power stage. The hydraulic supply is dual.

The surface position commands and sensor inputs are processed in the four identical digital flight control computers. The flight control computer was designed around a bit slice processor developed by GEC Avionics, to ease the integrity assessment task. The basic processor configuration is quadruplex to satisfy the requirements of a system probability of loss less than 10^{-7} and to survive any two electrical failures. Triplex configurations were not used because of their reliance on self monitoring techniques. The four processors are loosely synchronized so that the major computational frames are initiated at the same time, and the signal values used by the signal selectors and therefore by the control law algorithms, are taken from the same time samples of the input signals. Sensor data is cross fed to all processors and then voted so that interlane tolerances are reduced to improve failure monitoring.

Two additional identical analog processors have been added to match the dual triplex configuration of the actuation system. Each analog processor receives position commands from the four digital processors, consolidates those inputs and generates the position commands for two of the six first stage control valves of the actuation system. A major objective of the program was to establish whether such architectures could be proven. The software is common to all four processors, which presents the problem of a generic error leading to a safety critical situation. Very tight control measures were then exercised to guard against that possibility.

The software design was based on functional specifications developed by the airframer (BAe). The next stage of functional decomposition was the development of a Software Requirements Document, which detailed how the software would provide the required functions, the structure of the software, the

interfaces, the major algorithms, etc. The major functions provided by the software are: Executive; Data Handling; Signal Selection and Monitoring; Control Laws; Failure Identification and Monitoring; Built-In-Test.

The software is coded in Assembly language. An enhanced version of an existing instruction set (BOEING YC-14) was used. The enhancements improved the throughput of the flight control algorithms. The software support tools were derivatives of tools of proven maturity. The software development process was very structured. The Software Requirement Document was the key document which controlled the design implementation. Strict standards were imposed throughout to emphasize simplicity and clarity.

In order to give an indication of the relative importance of the sources of errors and of the effectiveness of software testing techniques in uncovering the software defects, software Change Requests (CR) generated during this project have been broken down into the categories shown in Table 2.2.

CR Category	% of Total
Design	44
FMEA	24
Rig Testing	12
Code Errors	7
Not Required	13
Total	100

Table 2.2 Software Change Request Experience on the Jaguar FBW program

The most common reason for a change request was modification to the functional specifications due to the fact that the system and the control laws were developed in parallel with producing the equipment. The small percentage of coding errors is a result of the thoroughness of module testing. Failure Modes and Effects Analysis (FMEA) techniques are clearly effective for software error detection. Performance trials of the software carried out in testing rigs is also very effective in detecting interface errors and leads to clarifications and enhancements of the required functions.

A system safety assessment, which covered all aspects of hardware and software, was conducted prior to flight. This assessment was based on analyses of the total system architecture, the hardware design and build, and of the software, including the software production procedures. A quantitative analysis was made of the hardware reliability and fault tolerance. No technique was found, however, which would properly quantify the probability of residual software errors. Permission to fly was based on the confidence of the assessors in the depth and dissimilarity of the testing techniques, the software analysis, and the controlled procedures used.

2.2.1.5 A320 Digital flight control system

The A320 is the first commercial aircraft which incorporates a full authority digital flight control system. The experience gained with the Concorde analog FBW system, with mechanical backup, provided the confidence to proceed with the development of the A-320 critical DFBW system. The systems for pitch, roll, and yaw control are shown in Figures 2.7, 2.8, and 2.9.

In order to meet the safety requirements for certification of critical equipment, the A320 flight control system is based on a highly redundant architecture with the addition of some special features to cope with common design faults. A mechanical backup system is provided in the pitch axis is through horizontal stabilizer trim and in the lateral axis through rudder control. The implementation approach selected by Aerospatiale provided the needed degree of safety by using:

-3 identical Spoiler & Elevator Computers (SEC) made by Aerospatiale
-2 identical Elevator & Aileron Computers (ELAC) made by Thomson-CSF
-3 identical Air Data & Inertial References Systems (ADIRS) with separate sensors made by Honeywell
-2 identical Flight Augmentation Computers (FAC) made by SFENA
-3 separate hydraulic channels (1 common + 1 for each side of airframe)
-2 separate main electrical power supplies plus 3 backups (Auxiliary Power Unit, Ram Air Turbine, batteries)

The loss of control of the aircraft is highly improbable, either in the pitch axis (in that both SECs and ELACs control the elevator) or in the lateral axis (in that SECs can control the spoilers and ELACs the ailerons) due to the following:

-The hardware of SECs and ELACs are dissimilar: i. e. dissimilar specs, dissimilar components up to processors (80186 for SECs, 68000 for ELACs), different suppliers

-The software is dissimilar: i. e. dissimilar specs, different suppliers, dissimilar methodologies, tools, and languages.

Furthermore, each FCC (SEC or ELAC) consists of two separate channels that communicate by an ARINC 429 bus (asynchronous point-to-point serial data link, mainly used in commercial transport aircraft). The control channel computes the control laws and actuates the surfaces (ailerons, spoilers and elevators) whereas the monitor channel computes control laws and checks the correct actuators, and monitors the control channel. Although they are installed in the same box, the control and monitor channels use fully independent hardware (even for power supply modules) and dissimilar software (e.g: for SEC, the control channel is written in assembly language by one engineering team; the monitor channel is written in Pascal by another team; fixed point arithmetic is used; thorough testing to the DO 178A level 1 software requirements). The channels are loosely synchronized.

Output data from the ADIRS are voted by SECs or ELACs before being used. Each ADIRS includes built-in-test (BIT) which provides additional robustness to the voting scheme of the Flight Control Computers with respect to self detected faults (the self detected faulty ADIRS is excluded from the voting process). Threshold values are a function of several factorsp of computational algorithms.

Side stick controllers are used which are not mechanically coupled. Each FCC includes a priority logic based on both sides stick signals. Normally one of the FCC's is the master, while the others are either in standby or in a slave mode. The masters choice varies depending on the surface. In case of failure, reversion inside each computer (alternate control laws) or to an alternate computer is fully automatic.

2.3 Implication of Design Choices on the Validation Process

In this section, the implications of various architectural characteristics on the verification and validation process are discussed.

2.3.1 Architectural Issues

The basic configuration of three of the five flight control systems which have been described is a quadruplex configuration. The X-29 flight control system has a triplex primary configuration which can revert to a triplex analog back-up system.

Considering typical reliability assumptions for single channel failures, quad redundant configurations can be shown to meet the flight failure rate of 10^{-7} Failures/Hour(F/H). These configurations detect and isolate faults, in real time, based on majority voting algorithms. If a failure occurs when only two channels are operational, then reversion to a degraded control mode, or to a fail safe configuration does occur.

Flight critical system fault tolerance and reliability requirements can also be met by triplex configura-

tions, if self test techniques with an appropriate failure coverage are utilized. If the assumption is made that the failure rate of a simplex flight control system channel is of the order of 10^{-3} F/H, then a triplex system can satisfy the 10^{-7} F/H requirement only if an overall failure coverage equal to , or greater than, 96.7% can be achieved with a combination of self test techniques. That coverage, however, is difficult and costly to accomplish, demonstrate and validate. As a result, even if self test techniques can reduce the required hardware redundancy, they are seldom used for that purpose in flight critical applications. Self test techniques, however, are often implemented for the following purposes: a) detecting failures in flight control system components which are only active in limited regions of the flight regimes, like autoland, and b) supporting the off line maintenance process.

Other major configuration issues which effect the validation process of flight critical systems are: a) synchronous vs. asynchronous; b) use of back-up systems; and c) separation of critical and non critical functions. They are briefly discussed in the following paragraphs.

2.3.1.1 Synchronization Techniques

There are three broad categories of synchronization techniques. The boundaries between them are not sharp, and a variety of perturbations of these basic techniques have been used in operational and experimental DFBW systems. All three synchronization schemes have been developed to flightworthy maturity.

Tight Synchronization

The tightest form of synchronization is instruction-level synchronization, where a common clock is used to drive each of the CPU's in step, thus causing all of the CPU's to execute exactly the same instruction at the same time. A voting plane at the sensor input is provided to ensure that each channel secures an identical set of input data. This results in an automatic bit-identical output from each of the computers. This permits straightforward cross-channel checking at the output, at the least significant bit level. The validation of the failure detection and isolation system is simplified because bit-by-bit checking is a relatively simple process and because of the knowledge that all computers are precisely synchronized in time.

Tight synchronization does require a common, fault-tolerant clock to provide timing signals to all computers. This mechanism becomes a source of potential common-mode faults or errors. Such a system was used to synchronize the triplex digital flight control system in an experimental U.S. Army helicopter program, called TAGS (Total Automatic Guidance System).

Frame Synchronization

A looser form of synchronization is "frame" synchronization, the frame being the shortest computational segment in the application program. This is also often termed the "major cycle". In this approach, all processors rendezvous at the end of a computational frame and resume processing after an exchange of information. Typical synchronization skew varies from 20-50 microseconds, depending on the approach used. In this approach, hardware synchronization of the clocks is not required, and the computers are not executing the same instructions at the same time. Voting planes at the sensor input can produce bit-identical outputs, although skewed by the synchronization variation. Because synchronization skew is small, analog voting, and cross-channel comparison at the analog level can be used for fault detection and isolation. The design of output failure detection and isolation requires the small synchronization skew to be accommodated in threshold selection. Validation of this approach is more complex than for tight synchronization. The synchronization algorithm is a source of common-mode faults or errors, and resynchronization following an "upset" is often a challenge.

This form of synchronization is common in contemporary DFBW systems, and is characterized by a moderate amount of design and validation effort required for effective implementation. The F-18 production DFBW system and the experimental F-8 DFBW, Jaguar DFBW system, and X-29 DFBW systems used this form of synchronization.

Asynchronous Systems

In this approach, each channel executes its program independently of the other channels. The computers still exchange information, but all exchanges are designed to be possible for any synchronization skew. One motivation for this approach is to minimize the potential for common-mode faults or errors, using the fact of interchannel skew as a means of avoiding correlated faults in the channels. Input data skew can be reduced by operating the input process at a rate higher than the computational cycle.

The design of the output voting scheme must include considerations of maximum time skew and varying time skew, because output command variations among channels vary with synchronization skew. The validation process must account for the fact that the channels can be in an infinite number of relative states. The asynchronous approach is used in the F-16 C/D production DFBW system.

2.3.1.2 Availability of Dissimilar Back-Up Systems and Reversion Configurations

Alternate control methods have generally been provided to "back up" primary digital fly-by-wire systems, in the event of loss of the entire primary system. These systems provide control over a subset of the aircraft's flight envelope, and usually offer degraded aircraft operational capability. Both hardware and software dissimilarity is often used in flight critical applications. The major reason for using dissimilarity is preventing catastrophic consequences in the case of a) common errors in all channels, including design errors; or b) exhaustion of spares in the primary control channels. As the level of confidence increases relative to the operational reliability of digital channels, the primary concern is undetected common errors in all channels.

All processor channels of the Jaguar Fly-by-Wire flight control system, as an example, use identical hardware and software, and reversion is not provided to a degraded control mode, in the case of design errors or exhaustion of spares. The advantage of using identical software and hardware is that only one set of hardware components, one set of software programs, and a single software development environment are needed. The disadvantages, relative to the validation process, is that it requires exhaustive and labor intensive effort for achieving the confidence that the system is absolutely free of any design error. The approach was clearly successful in the case of the Jaguar DFBW flight control system, a technology demonstrator program. Clearly there is a "trade off" to be made between the increased resource required to validate a system with a similar architecture and the lower recurring cost of the implementation.

The prevalent approach is to use some form of dissimilarity in flight critical applications. In fact, the other four example systems previously discussed all use some form of dissimilarity in the processing elements of the flight control system. The F-16 has a primary flight control system and a back-up system which uses the same hardware as the primary system, and dissimilar software. Certain pilot aids are deleted, but full envelope performance is maintained. The F-18 has a mechanical back-up system for pitch and roll control, and an analog back-up system for all control axes. Finally, the A-320 extensively utilizes dissimilarity in both hardware and software to achieve the high degree of reliability required for certification. The disadvantages of dissimilarity, in the primary or back-up systems, is that it requires additional hardware and software. The advantages are that it diminishes the concerns that residual, undetected design errors could have catastrophic consequences.

In all cases with an alternate control capability, that system must undergo a validation process similar to that for the primary system. In addition, the interfaces between the systems must be shown not to introduce catastrophic failure mechanisms.

2.3.1.3 Partitioning of Functions with Different Criticality

The development and verification costs of flight control system escalate very rapidly as a function of the criticality level. Therefore it is important to partition functions which have different levels of criticality. Function "A" is partitioned from function "B" if no action from function "B" can cause a failure in function "A." If partitioning can not be demonstrated and, function "B" is less critical than "A," then "B" automatically assume the same high level of criticality as function "A," because a failure of "B" can cause a failure in "A." Partitioning can be achieved with a combination of software and/or hardware techniques.

2.3.2 Software Issues

The structure of the embedded flight software reflects the critical and complex nature of the application. The structure, by design, is very simple. It involves the repetitive execution of sequences of application tasks, at fixed execution rates which are multiple of a basic frequency. Common values of the basic frequency are 200, 100, 80, and 60 Hz. The application tasks are not interruptible. Once initiated they must execute to completion, within the allocated time. The main advantage of this structure is that it significantly reduces the number of system states which must be verified, by eliminating the uncertainties related to random interruptions of the execution of critical tasks.

If additional processing time is available, low priority, interruptible tasks are executed. This commonly used foreground/background structure makes optimum use of the available resources and, at the same time, it minimizes the complexity of flight critical software.

The software needed to perform the most critical control functions is typically replicated in all channels. Less critical functions might be performed within some of the channels only. To gain a better understanding of the computational requirements of such systems, the software is partitioned according to the major functions which must be performed. They are:

a) Application. The algorithms included in this function are those required for sensor processing and filtering, control and navigation algorithms, computation of control command, etc.

b) Logic. Modules in this category perform the computation required for switching control and flight mode, and engaging/disengaging logic. They use almost exclusively Boolean statements.

c) Testing and Voting. These modules perform real time tests on processors, memory, sensors and actuators. They manage and control the overall system configuration, as a function of detected failures. BIT is included in this category.

d) I/O. The modules perform data handling and formatting, data transmission and display. Peripherals drivers and included in this category.

e) Executives. The modules perform the task of initializing power-up procedures, synchronization, scheduling and timing.

The typical memory requirements of each software functions, as a percentage of the total, are shown in Table 2.3.

Software Functions	Memory Requirements/Percent
Application	20
Logic	25
Testing	20
I/O & Executives	20
Miscellaneous	15
TOTAL	100

Table 2.3 Memory Requirements for flight control system Software

The data shown in the table represents typical results from the combined experience of the AGARD Working Group members. Clearly differences exist from case to case. As an example, in the case of a system which relies significantly on self test for the purpose of failure detection and isolation, testing would take a larger percentage of the total memory than that shown in the table. Highly redundant, distributed FCCS configurations, have a high proportion of software related to self test, failure management, communication, and system management. The software of simplex configurations, with limited integration of functions, has a high proportion of a application code.

Equally important for understanding the software of critical, redundant flight control system is an analy-

sis of the nature of the software errors which are detected during software and system development. A common conclusion for all programs is that the majority of software errors are generated during the early phases of the development process. Five general error categories have been identified. A typical frequency of occurrence of each category is shown in Table 2.4 [6].

Software Error Category	Frequency/Percent
Computational	10
Logic	25
Data Handling	35
Interfaces	10
Miscellaneous	20
TOTAL	100

Table 2.4. flight control system Software Error Distribution

It is important to notice that the table does not include the occurrence of trivial coding errors, which are easily detected during the software coding and debugging and even during module testing. The table includes only those errors which were detected while the software was in configuration control, and therefore had already been extensively tested at the module level and at the level of software integration. Those errors were detected during the software/hardware integration test, iron bird test, and flight test. The table shows that errors associated with logic, interfaces and data handling, including testing and voting are the biggest contributors. There are some reasons why this occurs. These functions are in fact rather new, relative to the application functions. The necessary tools are not well developed and the understanding of complex architectures (and therefore complex interfaces), logic and failure detection and reconfiguration algorithms is not as mature and established as that of the application algorithms.

Relative to the validation process, the major software issues are: a) the use of High Order Languages (HOL) vs. assembly language; b) the use of dissimilar software; and c) the methods for supporting the early phases of the design process. They are discussed further in the following paragraphs.

2.3.2.1 HOL vs Assembly

The use of HOL is increasingly accepted in flight applications. HOL is used in two of the previously described systems (F-16 C/D, and F-18). Two others (X-29 and Jaguar) use assembly languages. The A-320 uses both HOL and assembly languages. There are many reasons for the increasing use of HOL:

a) HOLs such as ADA can be used as a PDL and therefore provide a great benefit by increasing the commonality among the aircraft's systems and traceability between design documents and corresponding code source.

b) Compilers are getting more efficient. The memory and time penalties associated with the use of HOL instead of assembly are becoming smaller (<50% and <20% respectively), and are becoming less significant due to increasing computational speeds and decreasing cost of memory.

c) There is a strong indication that HOL programs are more reliable than assembly programs. This is due to the greater level of abstraction HOL provides.

d) HOL programs are easier to develop, test, understand, modify and maintain than assembly programs for FCS, because they usually use only very simple and straightforward software due to criticality.

e) Significant experience is now available relative to the performance and the code generated by mature software tools, like compilers, assemblers and linkers while at the same time, great advances in memory and computing power have been achieved, reducing the overhead of HOL.

f) HOL easily allow the use of "safe subsets" that have been proven to be reliable, testable, and readable. Checkers for verifying the compliance to the subset can be easily built.

It is anticipated that the trend towards increasing use of HOL will continue. The main problem with this approach will certainly be the validation of the compilers (with respect to the safety) and that of the real-time kernel. However, specific parts of the software (I/O, interrupts, handlers) could still be written in assembly for some years and be integrated with other parts written in HOL.

2.3.2.2 Dissimilar Software

The effects of using dissimilar components and functions was discussed in previous paragraphs of this paper. In this paragraph the effects of dissimilar software, in particular, are analyzed. Dissimilar software can be used in three ways:

a) as a failure detection mechanism. In this case results from both versions of software are compared periodically and, in the case that the results differ by more than certain values, a fault condition is detected. In that case, reversion to a back-up control mode must be made, which does not utilize either of the two software versions:

b) as a back-up control mode. In this case, if a common failure is detected in the primary software, reversion is made to the alternate version. Alternate software versions typically provide limited capabilities only. The F-16 C/D flight control system is an example of this application.

c) as a way to achieve software fault tolerance. In this case redundant, but not identical, versions of the software are implemented for detection of software failures and for providing alternate, usually degraded, computational paths. Recovery blocks and N-version programming are two techniques used for these purposes.

For the A320, points a) and c) described above are addressed by control/monitor channel implementation, as well as ELAC/SEC architecture.

2.3.2.3 The Early Phases of the Development Process

Clear evidence exists that the software errors which are most difficult and costly to detect are often introduced early in the development process. This points to the need of tools, techniques and methodologies capable of effectively supporting the specification and design phases. The entire design and validation process must be supported by integrated development environments, which include specification and design languages with powerful diagnostic capability, and which are easy to use.

It is extremely important to define, as early in the development cycle as possible, design disciplines which make the software traceable, testable, maintainable and easy to understand. Design and coding standards must also be established, like:

a) limiting the complexity of the smallest software blocks within the human analysis capabilities; complexity depends mainly on the number of embedded constructs (if-then-else, loops, gotos....) and also on the number of lines;

b) avoiding design features or coding constructs whose dynamic behavior is untractable or which may result in memory overflows (either in the stacks or data allocated areas) or timing overrun; complex event-driven schedulers, dynamic memory allocation, recursive/unlimited embedded calls should therefore as much as possible be banned for FCS applications;

c) enforcing the use of "robust" programming; this may include reasonableness testing within the operational software or exceptions handling.

2.3.3 Sensor/Actuator Issues

The fault tolerance and reliability requirements of advanced flight control systems often require redundant configurations of critical sensors and actuators for supporting functions required for continuous safe flight and landing.[7] Not all sensors and actuators have the same level of criticality, so it is rather common that different levels of redundancy are used within the same flight control system. The required level of hardware redundancy is also affected by system level considerations, like the function of analytical redundancy which might exist among different groups of sensors or actuators, and the availability of back-up systems.

Analytical redundancy has been employed at the sensor plane. The objective is to provide a synthesized feedback signal, in the case of failure of the primary sensor suite, by analytically combining (or fusing) information from other sensors. Functional redundancy is employed at the aircraft effector plane. The objective is to generate forces and moments about some control axis, in the case of failure of the primary effector, by appropriately modulating a combination of other operational effectors. The functional redundancy among many effectors existing in advanced vehicles makes this approach feasible. Another good example of functional redundancy is provided in roll control of the A320: roll control may be achieved via ELAC (ailerons) and SEC (spoilers), with reduced efficiency, if ELAC's or SEC's are lost.

The availability of analytical and functional redundancies has profound effects in the validation of flight control systems. They might reduce the criticality of some sensors and effectors and correspondingly decrease the validation effort of the equipment involved. They might also reduce the hardware redundancy needed for satisfying specific fault tolerance requirements. In this case additional validation effort will be required to demonstrate the availability and the effectiveness of those redundancies.

It is important to note that, although a sensor or an actuator is often referred to as having a certain level of redundancy, that level of redundancy often applies only to some, not all, of the elements (and/or functions) that the equipment comprises (and/or performs).

As an example, the sensing component of an Linear Variable Differential Transformer (LVDT) sensor, could be single or dual. The electrical paths to the Flight Control processors, however, could be quad, in the case that the LVDT interfaces with a quad flight control system configuration. The redundancy of the first control stage of an hydraulic system might also be quad, to reflect the flight control system architecture, but the main ram and the hydraulic supply might be only dual. In most flight critical systems it is imperative to eliminate all features where a single point failure can cause a loss of control. In extreme cases where such a point cannot be eliminated the regulatory authorities will insist on a rigorous analysis and demonstration that no realistic failures can occur at that point.

Many considerations determine the final configuration of the control systems of an advanced aircraft. The objective is often to find an acceptable design among many different, and some time conflicting requirements.

REFERENCES

[1] Rediess, H.A., and Buckley, E., "Technology Review of Flight Crucial Flight Control Systems", NASA CR 172332, April, 1984.

[2] a Harschburger, H.E., Glaser, B, and Hammel, J.R., "Backup Modes for the F/A-18 Digital Flight Control System", AIAA 84-2622, 1984

2 b "Landis, K.H., and Glusman, S.I.; "Development of ADOCS Controllers and Control Laws.", NASA CR 177339, July, 1984.

2 cRamage, J.K., "AFTI/F-16 Automated Maneuvering Attack System Configuration and Integration, Proceedings of the IEEE 1986 National Aerospace and Electronics Conference, Dayton, Ohio, May, 1986

2 c Whitaker, A, and Chin, J., "X-29 Digital Flight Control System", AGARD-CP-384, October, 1984

2 d Hillis, A., "A310 Slat and Flap Control System Management and Experience", 5th Digital Avionics Conference, Seattle, WA, November, 1983

2 eMarshall, R.E.W., Snelling, K.S., and Corney, J.M., "The Jaguar Fly-by-Wire Demonstrator Integrated Flight Control System", Proceedings of the United States Air Force Academy Advanced Flight Controls Symposium, 1981.

2 fFlapper, J.A., and Throndsen, E.E., "L1011 Flight Control Systems", Integrity in Electronic Flight Control Systems, AGARDograph No. 244, 1977.

2 g F-16A Anderson, C., "F-16 Multinational Fighter", AGARD-AG-234, November, 1978

2 h Comey, J.M., "The Evolution of the EAP Flight Control System. Presented to the International Symposium on Aeronautical Science and Technology of Indonesia, 24-26 June 1986.

2 iA-320. Aviation Week, May 18, 1987, p. 41-45

2 jKubbat, W., "A Quadredundant Digital Flight Controls System for CCV Application." Impact of Active Control technology on Airplane Design, AGARD-CP-157, June 1975.

2 k Szalai, K.J., Felleman, P.G., Gera, J, and Glover, R.D., "Design and Test Experience with a Triply Redundant Digital Fly-by- Wire Control System, AIAA 76-1911, August, 1976.

[3] Harschburger, H.E., Glaser, B, and Hammel, J.R., "Backup Modes for the F/A-18 Digital Flight Control System", AIAA 84-2622, 1984

[4] Burton, R., Kneeland, Rabin, and Hansen "Flight Testing and Development of the F/A-18 Digital Flight Control System", AGARD-CP-384, Toronto, Canada, October, 1984.

[5] Smith, R, "Flight Clearance of the Jaguar Fly-by-Wire Aircraft", Royal Aeronautical Society Symposium, April, 1982.

[6] Hecht, H.M., "Trends in Software Reliability for Digital Flight Control" NASA CR 166456, April, 1983.

[7] Fault-Tolerance Design and Redundancy Management Techniques, AGARD-LS-109, September, 1980.

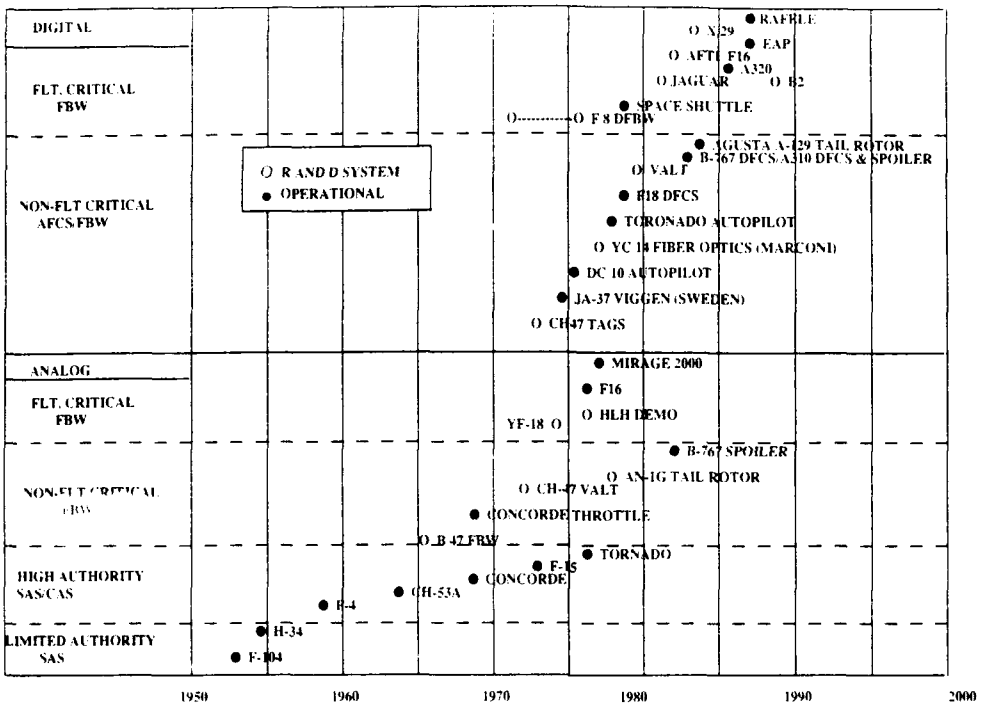


Figure 2.1 Trends in Flight Control Systems

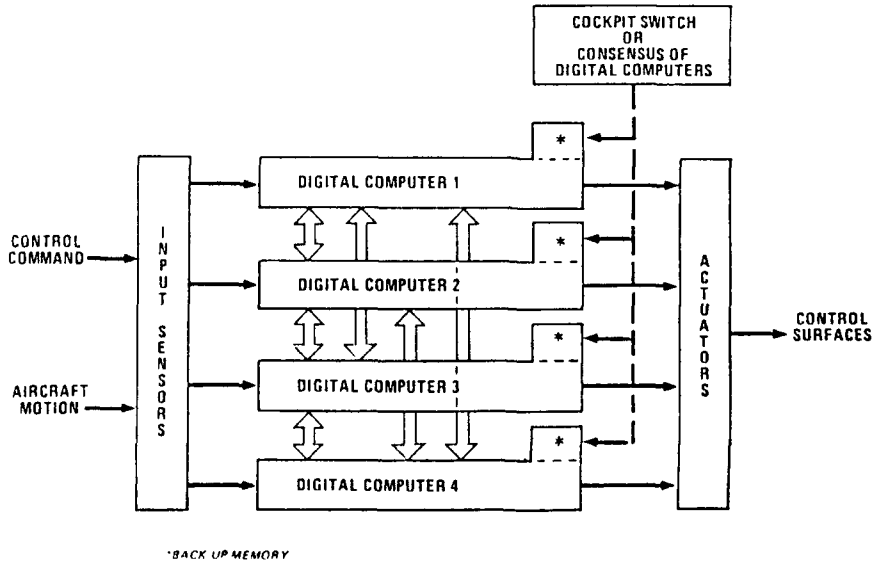


Figure 2.2 F-16 C/D Production Quadruple Digital with Back-Up Memory

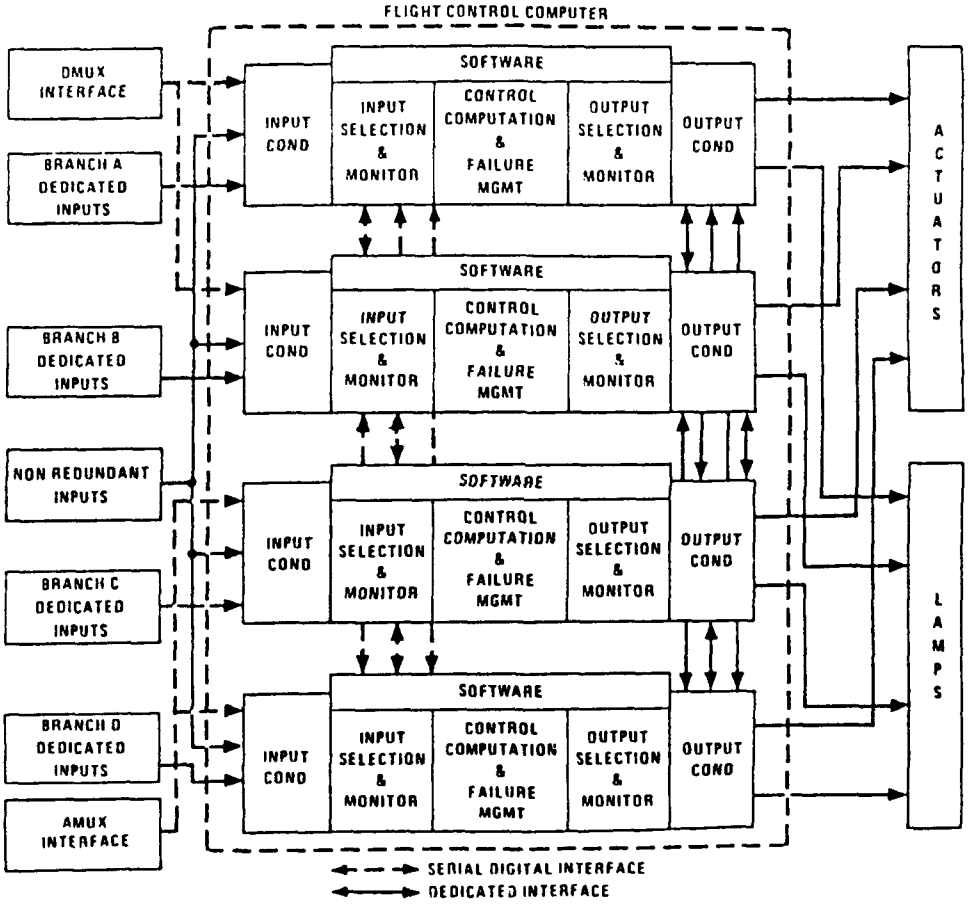


Figure 2.3 F-16 Flight Control System

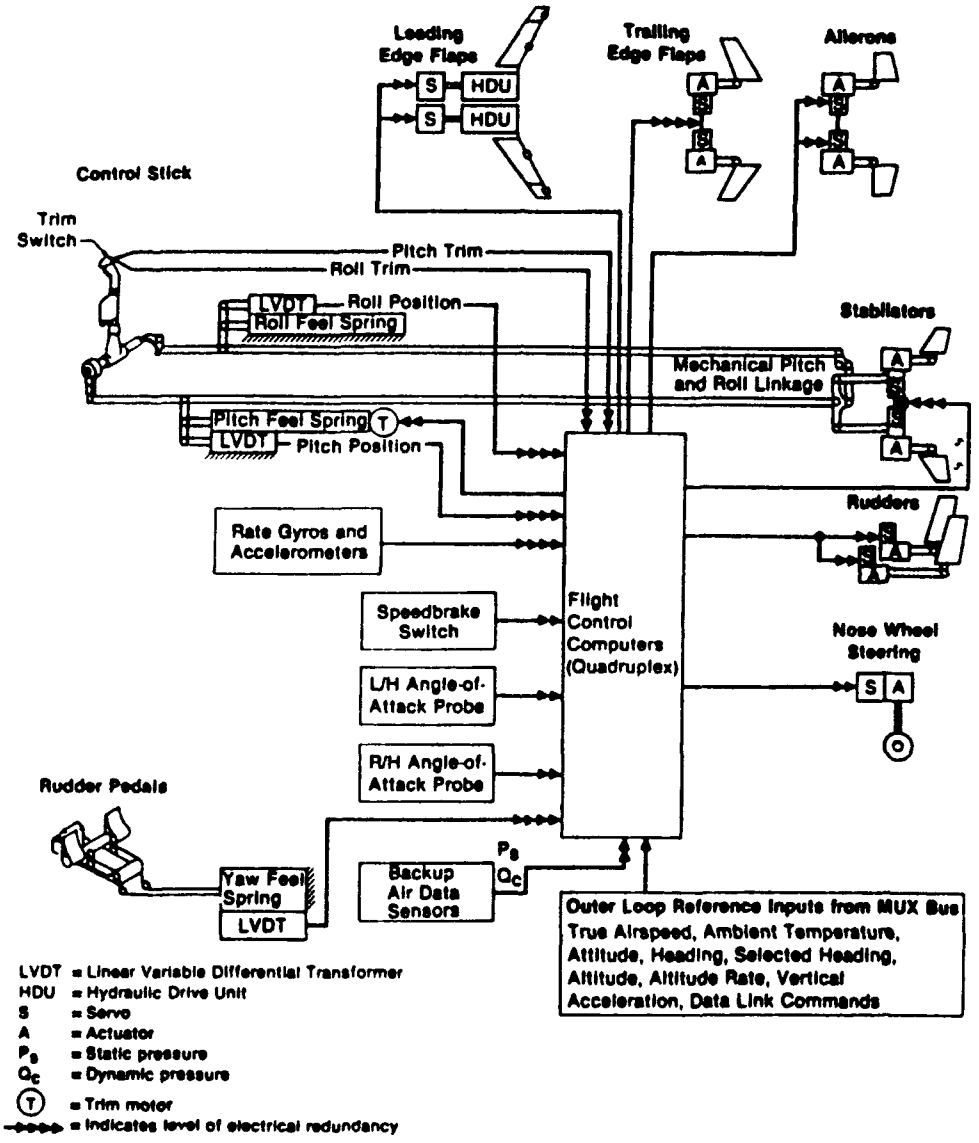


Figure 2.4 F/A-18 Flight Control System Functional Diagram

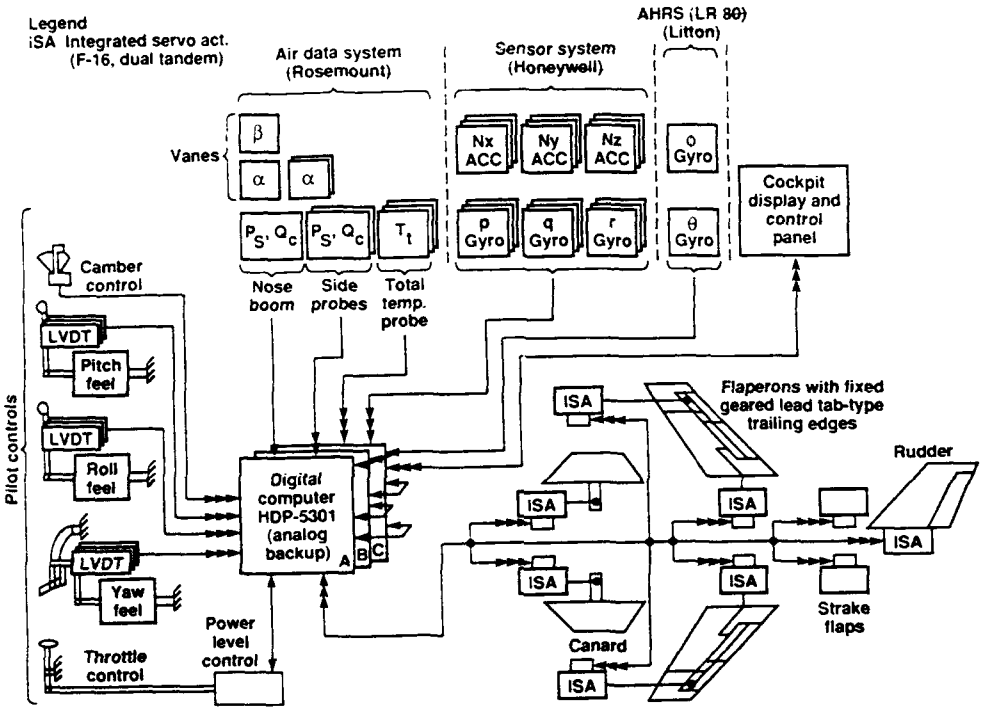


Figure 2.5 X-29 Flight Control System Schematic

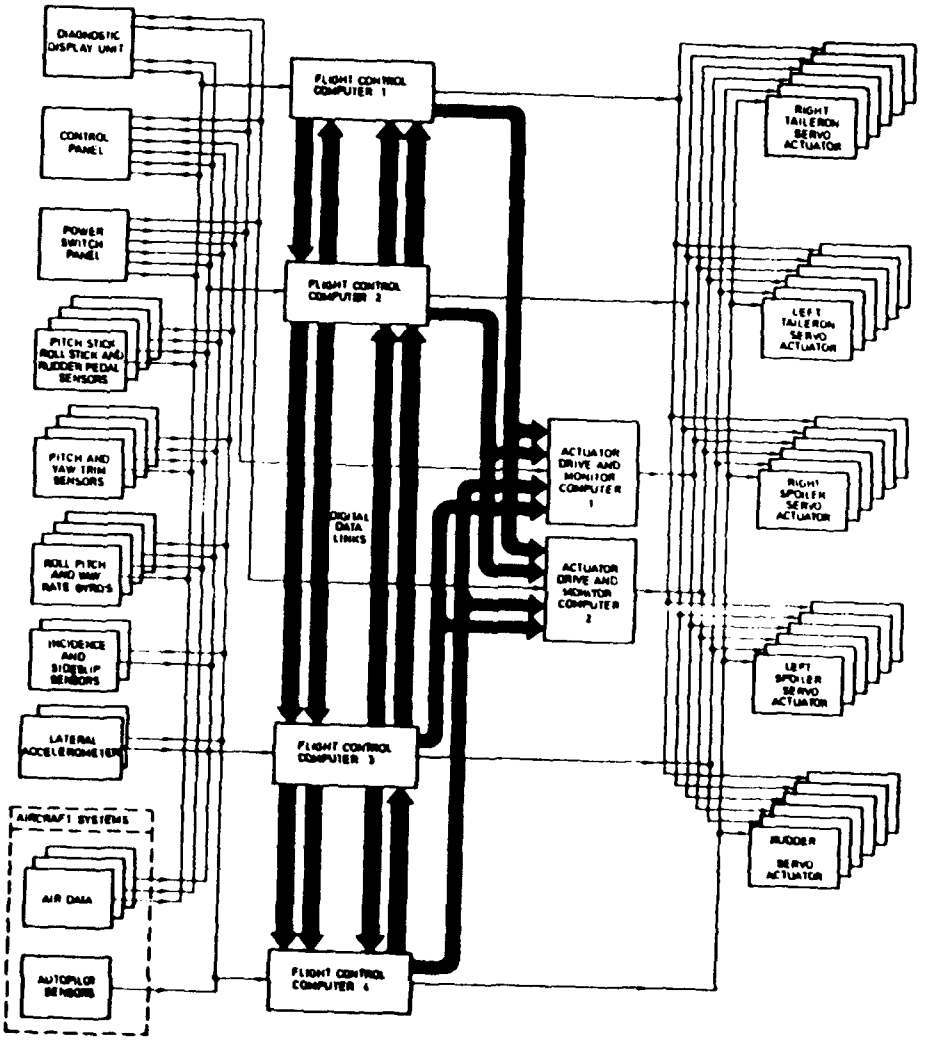


Figure 2.6 Jaguar FBW Architecture

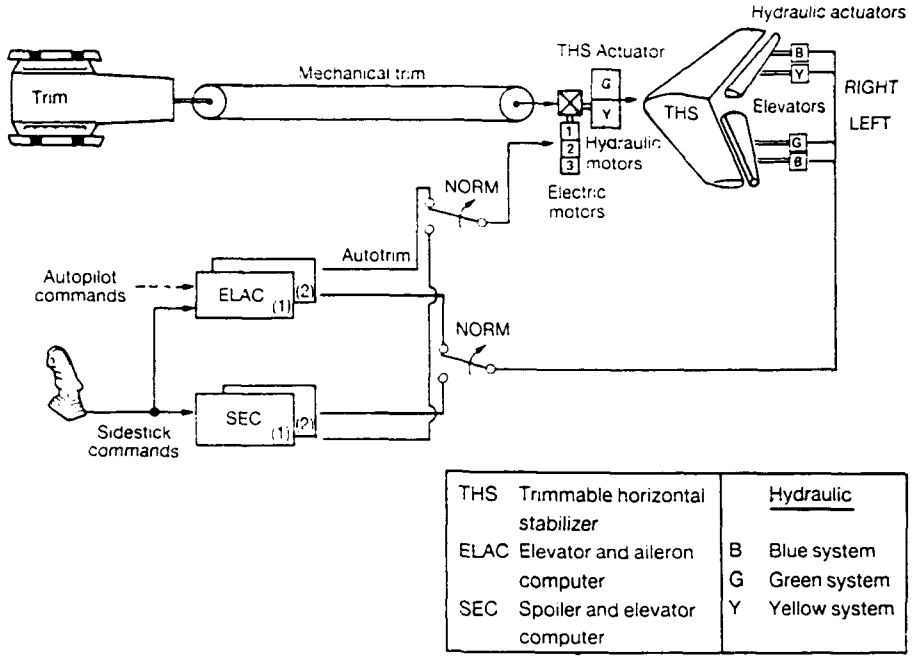


Figure 2.7 A320 Pitch Control

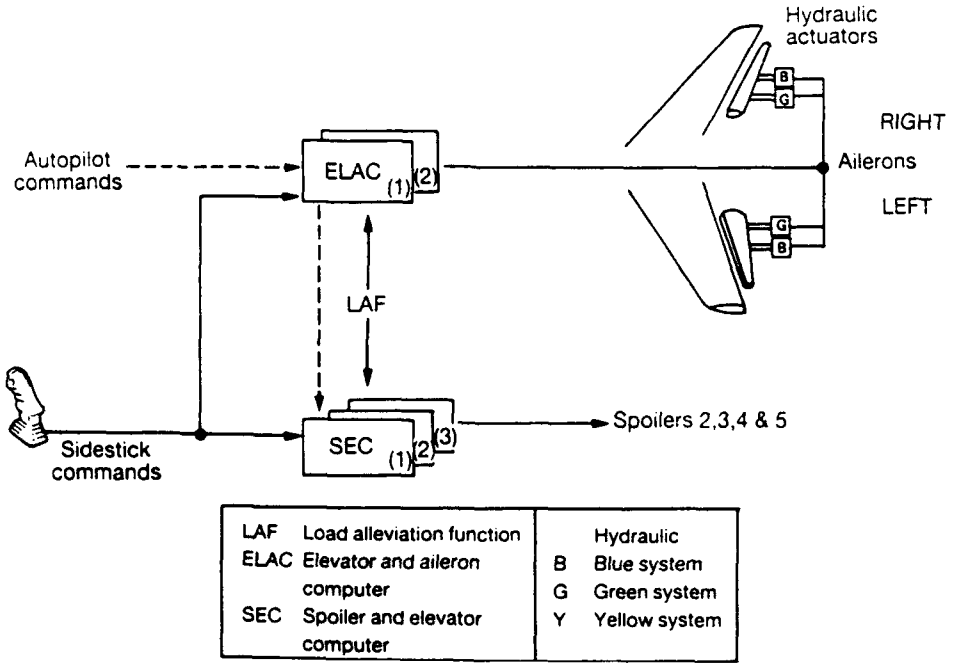
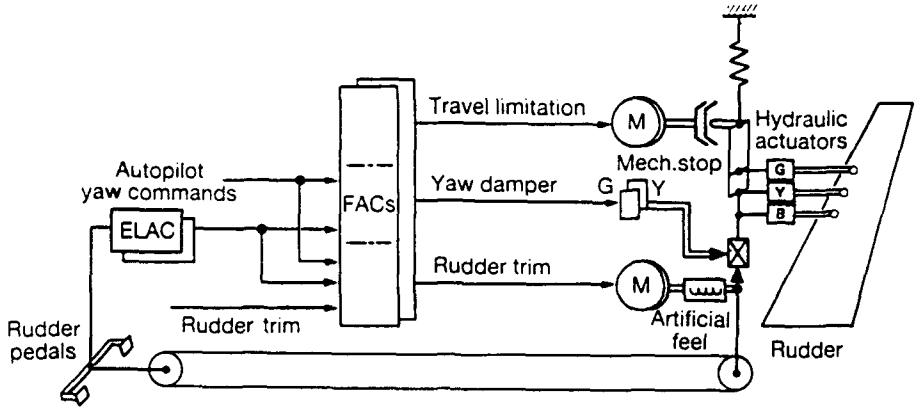


Figure 2.8 A320 Roll Control



M	Motor actuator	Hydraulic	
FAC	Flight augmentation computer	B	Blue system
		G	Green system
		Y	Yellow system

Figure 2.9 A320 Yaw Control

CHAPTER 3

SOA GENERIC DEVELOPMENT AND VALIDATION PROCESS

3.0 Introduction

The validation process is embedded in a complex series of events making up the development of the flight critical control system (FCCS), which is only part of the flight system and total airplane development process. A well organized and systematic airplane and flight system development process is a necessary foundation for a successful and efficient validation program.

The purpose of this chapter is to provide a top level description of the FCCS validation process and its relationship to the overall airplane and flight systems development cycle. It serves as a guide and background to chapter 4, which contains a very detailed description of the state-of-the-art tools, techniques, methods, and approaches used in the validation of the FCCS.

There are many ways in which systems can be developed and validated and these ways change with time. The method described in this chapter is a generic process, based on the experience of the members of the working group who have been associated with the development and validation of most of the flight critical control systems produced in Europe and USA during the past two decades.

It should be noted that in addition to the application which is the focus of this report, the methods used to validate FCCS's are sufficiently general and rigorous that they can be used to validate other flight critical systems such as those given in Table 3.1.

3.1 Relationship of FCCS Development to the Military Aircraft Life Cycle

The top level requirements for the flight control system of a military aircraft are derived from the system requirements associated with the aircraft/weapon system mission and operating requirements. The relationship between the development of the airplane and the flight control system is shown in Table 3.2, as characterized in the U.S. A more detailed description of this military aircraft life cycle model, and how the flight control system development is embedded in it, is contained in Appendix 3.1.

3.2 The Life Cycle of a FCCS

Figure 3.1 shows the phases in the life cycle of a typical FCCS[1]. The process is sequential and the top level requirements are systematically converted into detailed designs with validation providing the feedback to check against errors and omissions.

The system is validated by showing that the production system meets the Goals and Requirements defined at the start of the process. This can be done by checking that the results of each phase meets the requirements placed on it by a prior phase, e.g., the Development Specifications may be validated by checking that they meet the Systems Specifications. Alternatively they may be validated in part by checking that the design meets the top level Goals and Requirements.

Various methods are used to validate the system in a particular phase. In the earlier phases validation may be based on abstract models, but as the project matures, the models become more concrete and the results of tests on prototype and production equipment are used for validation. Figure 3.2 lists the general categories of the methods used in each of the phases (also from Ref 1). Figure 3.3 shows how the validation process can be considered as a feedback path checking the operation of the forward path[2].

A clear hierarchy in the development and documentation process is essential for a successful validation

program. The documentation must be structured so that all the requirements are explicitly stated. Furthermore, the documentation should avoid confusion by being so organized that requirements are referenced only once. One method of providing this hierarchy is shown in Figure 3.4, based on DOD-Std 2167A. The following definitions are appropriate:

a. System Specification - A document which states the technical and mission requirements for a system as an entity, allocates requirements to functional areas (or configuration items), and defines the interface between or among the functional areas.

b. Development Specification - A document applicable to an item below the system level which states performance, interface and other technical requirements in sufficient detail to permit design, engineering for service use, and evaluation.

The key point here is that there should exist a clear and unambiguous relationship between the various specifications, development steps, and documentation items to provide a traceable and continuous flowpath of activities. This top-level view will provide a structure for the entire development and validation program which can be used by all parts of the organization.

3.3 Goals and Requirements

The Goals and Requirements are the overall attributes of the system; they include the mission-related requirements as defined in highest level staff targets, or statements of need. They also include the requirement to meet the appropriate national specifications (e.g., Mil Std 8785). Overall safety requirements are included, based on national specifications (e.g., Mil Std 882) or on historical data with appropriate extrapolation. The Goals and Requirements reflect the improvements in system performance which are possible due to advances in technology.

A semi formal statement of the Goals and Requirements is a valuable aid in the initial phases of a project; it promotes discussion, reduces ambiguities and highlights omissions. It acts as a focus for the initial discussions and as a criteria against which to validate subsequent phases. The validation of the Goals and Requirements can be accomplished in part by carefully comparing them against the mission scenarios, against historical data and against similar systems.

3.4 System Specification

The System Specification is a key document in the design of a FCCS. It is a statement of the functions which the system must provide and forms the basis for the subsequent detailed design and becomes the model against which the system is validated. The system specification usually follows a standard format and includes a definition of the context in which the system is to operate, a statement of the integrity objectives. Figure 3.5 is a typical format. The validation of the Functional Specification is accomplished by mapping it out against the Goals and Requirements by analysis.

It is possible to introduce semi formal methods to improve the coverage and consistency of the system specification. Software tools are now available to help in the generation and maintenance of specifications using semi formal methods.

3.5 Development Specification

Once the Functional Specification is established, a set of Development Specifications are produced. There will be many such specifications to cover various areas. A list of typical Development Specifications is given in Figure 3.6.

Development Specifications can be validated by checking that each of the functional requirements is covered by the specifications and that each of the requirements in the development specifications is consistent with the functional requirement. Again, this process can be facilitated by software tools using semi formal methods based on a mixture of text, graphics and computer language syntax.

In a broader interpretation development specifications include all the design definitions and as such they are models of the system which can be validated by analyses such as a failure modes and effects

analysis, to check that the design does meet the System Specification. Some excellent approaches have been suggested and developed for project documentation[3].

3.6 Implementation and Prototype

Figure 3.7 shows the activities which lead to the prototype implementation. The control laws determine the handling qualities of the aircraft, and the structure of the control laws determines the levels of hardware and software redundancy necessary to meet the integrity and availability criteria. Thus, the design of the control function becomes the central activity which defines the complexity of the systems and the reliability level required for each of the facilities within the system. The hardware designer, the software designer and the redundancy management designer cooperate to implement the specified control function.

The control law validation process uses a variety of modeling/analysis techniques (see Figure 3.8) all of which are based on aerodynamic data which must be comprehensive and accurate. Rigorous testing of models and prototypes is necessary to derive this data. The validation of these aircraft models is critical to the ultimate safety and performance of the flight control system.

Control laws in modern flight control systems are implemented in digital computers using software algorithms. The task of designing hardware with the capacity to meet all the requirements consists of a whole series of activities ranging from architectural studies, through stress and tolerance analyses, to FMEA's. Designs are checked by analysis and prototype testing.

The production of software is equally complex and stringent; measures have to be taken throughout the life cycle to ensure that the integrity of the software meets the requirements. Figure 3.9 illustrates a typical set of activities associated with the software development process.

It is necessary to check the integration of the hardware and software to ensure that the development specifications are met. Various methods of testing are used as the test phase moves from simulation to laboratory models, to benches, to rigs/iron birds, and finally to prototypes. The match with the real world improves with each step.

Test benches which allow the system elements to be interconnected, powered and stimulated with simulated inputs and outputs are used to check the detailed operations of the system. Although at each successive stage the testing becomes more realistic, the granularity of the testing becomes coarser. Thus the later phases of validation rely on the earlier phases for fine grained validation.

Rigs/iron birds are the next stage in the progression to complete systems integration and validation. In many FCCS development programs, the rigs/iron birds have been the key validation tools. They have combined representative hardware including cockpits, actuators using hydraulic systems with representative geometry and loads, flight control computers and sensors. They have the capability to check out the interfaces between units and ensure compatibility. They also have capability to trigger the system to reconfigure and hence test the redundancy management.

On many rigs it has been possible to perform pilot-in-the-loop simulation and thus validate many of the aspects of the handling qualities and pilot interactions resulting from system reconfiguration.

3.7 Prototype Aircraft

There is a clear trend toward conducting a larger amount of work on hot benches and rigs, but there remains a significant amount of testing that must be carried out on the prototype aircraft to ensure truly representative conditions. This includes check of interaction between systems, effect of aircraft structure on control system resonances, electromagnetic shielding/interference effects, power transients, cooling system performance, and other elements which are sensitive to the actual hardware installation.

The production of the evidence to support the request for first flight clearance is a key stage in the validation. This is a large and complex process and inevitably deals with a vast range of predictions and judgments. Figure 3.10 shows the process used to validate the Jaguar FBW system[4].

Once the first flight is completed a whole new set of data becomes available which can be used to check

out these predictions and judgments. Flight testing of prototypes is structured to gradually expand the information available and to check that the requirements defined in earlier stages have been met.

3.8 Production System

The final stage is to ensure that the production system has and retains the same performance as the prototype(s). The major considerations are configuration control to ensure repeatability, routine checks to ensure that latent faults do not invalidate system integrity during the long periods of risk experienced on production systems, and maintenance to ensure that the system is as designed/built.

To validate the fielded system it is necessary to maintain records of failures, maintenance actions, performance, and usage, to show that the productions system is performing in the way that the developer predicted.

APPENDIX 3.1 : A LIFE CYCLE MODEL OF A MILITARY AIRCRAFT

The following is a description of a generic life cycle model of a typical major military system, like a new aircraft. The model is consistent with the guidelines included in the System Engineering Management Guide which was published for the U.S. Defense Systems Management College. The guide describes an integrated system engineering and management approach, including methodology and tools, for defining the requirements, configuring and sizing the system, managing its development and verifying the capabilities of the design. It covers the acquisition and development process of any major military system from inception to operational deployment and use. For the purpose of this document, the system is intended to be the entire aircraft. The FCS is a prime or critical item.

The life cycle of major DOD systems includes five phases: Mission Need Determination, Concept Exploration Phase, Demonstration and Validation Phase, Full Scale Development Phase, Production and Deployment Phase.

The first Phase, Mission Need Determination, is carried out by the government. If successful, approval to proceed with the next phases is granted.

The Concept Exploration (CE) phase defines and selects promising system concepts for further analysis. Outputs include: definition of performance envelopes, preliminary alternate designs, feasibility studies, preliminary life cycle cost estimates, and a functional baseline. The major documents issued at the end of this phase are: a) a System Concept Paper (SCP) which contains a statement of needs, alternate designs with corresponding performance estimates, and a risk assessment analysis; b) a Systems Requirements Review (SRR); c) a functional baseline which states the technical and mission requirements for the entire system as a single entity (such as the aircraft); d) a Request For Proposal (RFP) for the next phase, which contains the functional baseline (Type A specifications), management approach, and the Statement of Work (SOW) which describes the scope of the contractor effort. During this phase, major subsystems are identified, and preliminary performance requirements are developed. For example, several concepts of the FCS, engine, avionics, etc., may be developed and the performance defined. A SRR may be conducted at the end of this phase or soon after initiation of the next phase. A preliminary version of the Test and Evaluation Master Plan (TEMP) document is developed in this phase.

The Demonstration & Validation (D&V) phase main objective is to determine whether to proceed with the Full Scale Development (FSD). Several trade studies are conducted for evaluating the relative merits of competing concepts which were defined in the previous phase. The most promising concept is selected and the prime and critical Configuration Items (CI) are defined. A major output of this phase is the definition of the "allocated baseline" which satisfies the mission requirements established by the functional baseline by allocating specific requirements to the identified CI's. As an example, the FCS may be partitioned in the FCC, software and hardware, actuation system, sensor suite, pilot interfaces,

etc. Some of them may also be defined as a CI, based on their complexity and criticality. The functional baseline is finally allocated to each CI. Assessment of achievable performance, life cycle cost and technical risks are performed. High risk items and all items for which a proof of principle and component demonstration test is beneficial, may be prototyped. A System Design Review is conducted at the end of this phase which validates the allocation of requirements to the CI's. The activities performed in this phase are supported by analysis, simulations, emulations, and prototypes. The TEMP document is updated to include the requirements for all major testing equipment and facility. TEMP, however, still primarily addresses system test requirements, rather than the testing requirements of individual CI's. A detailed test plan is also developed which includes methods for validating flight critical functions and CI's.

The Full Scale Development (FSD) phase primary objective is to develop and to demonstrate the design of the system concept selected in the previous phase. During this phase a small number of prototype aircraft (usually not more than three) are built. The development process of the test facilities for the individual CI's and for the entire system is conducted concurrently with the development of the CI's consistent with the test requirements established in the previous phases. They may include the development of dedicated laboratory systems, real time simulation environments, and even an iron bird. The iron bird is a most realistic duplication of the actual aircraft environment including, but not limited to: a) physical dimensions; b) aerodynamic, mechanical, electrical and thermal loads; c) hydraulic, pneumatic and electric power; d) electrical, hydraulic and pneumatic connections.

The first step of the process of building the CI's is the development of the preliminary system design which ends with the successful completion of the Preliminary Design Review (PDR). The preliminary design defines the development specifications (Type B specifications) of the CI's so that detail design process can initiate. At the end of that process a Critical Design Review (CDR) is then conducted. The CDR encompasses all CI's, and most importantly the interfaces among CI's, and produces development type specifications. Up to, and and including, CDR the vehicle is a paper airplane only, although laboratory prototypes of many subsystems and components may have been built for supporting the evaluation process and providing proof of principle. After successful completion of CDR the Interface Control Documents (ICD) are finalized, the system design is frozen, and the development process of prototype CI's, like the FCS, start. Once developed, the CI's are tested in dedicated laboratory environments which simulate the entire range of operational conditions. The CI development process ends with successful completion of two audits: the Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA). FCA is performed to validate that the CI has achieved the required performance and functional characteristics. PCA is performed to validate that the CI "as built" conforms to the technical documentation, and it establishes the CI baseline.

As the CI's are baselined, they are integrated within the system test facilities. At the completion of System Integration Test, all CI's are successfully integrated and demonstrated, and a Flight Readiness Review is conducted which, if successful, clears the way to the Flight Test Program. The Flight Test Program is performed by flying the prototype aircraft, which have been assembled during the last phases of the FSD. After successful completion of the Flight Test Program the aircraft is qualified for operational service.

After the FSD, a Production Readiness Review (PRR) is held to verify that the system is ready to go into the next phase. A design data package is then developed which includes production drawings. An RFP is then issued which includes detail product specifications (Type C specifications), which reference or include the entire design data package.

The Production and Deployment (P&D) phase primary objective is to produce systems according to cost and schedule requirements. The CI's are "build-to-print." Typically the first production CI from mature tooling is subject to a PCA and a production baseline is established. Once a PCA has been successfully completed for all CI's, a product baseline for the system is established.

REFERENCES

[1] "Validation Methods for Fault-Tolerant Avionics and Control Systems", Working Group II, NASA Conference Publication 2130, Oct. 3-4, 1979.

[2] "Validation Methods for Fault-Tolerant Avionics and Control Systems", Working Group I, NASA Conference Publication 2114, March 12-14, 1979.

[3] The Implications of Using Integrated Software Support Environment for Design of Guidance and Control Systems Software. AGARD Advisory Report No. 229 February 1990

[4] Daley, E., and Smith, R.B., "Flight Clearance of the Jaguar Fly-by-Wire Aircraft", Proceedings of the Royal Aeronautical Society Working Group Symposium on Certification of Avionic Systems, 1982.

Fly-by-Wire Flight Control Systems
Large Authority CSAS
Large Authority Autopilots
Power Plant Control Systems
Secondary Flight Control Systems
Stores Management Systems
Terrain Following Systems

Table 3.1 Systems Which Can be Validated By the Techniques Described in This Report

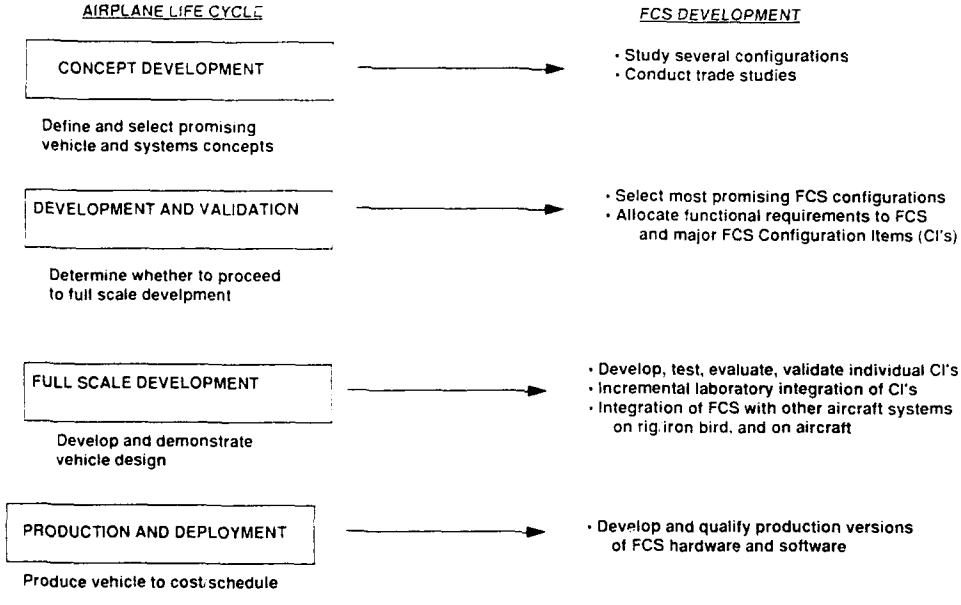


Table 3.2 Relationship of Flight Control System Development to Airplane Life Cycle

All Steps are iterated until frozen

VALIDATION ACTIVITIES

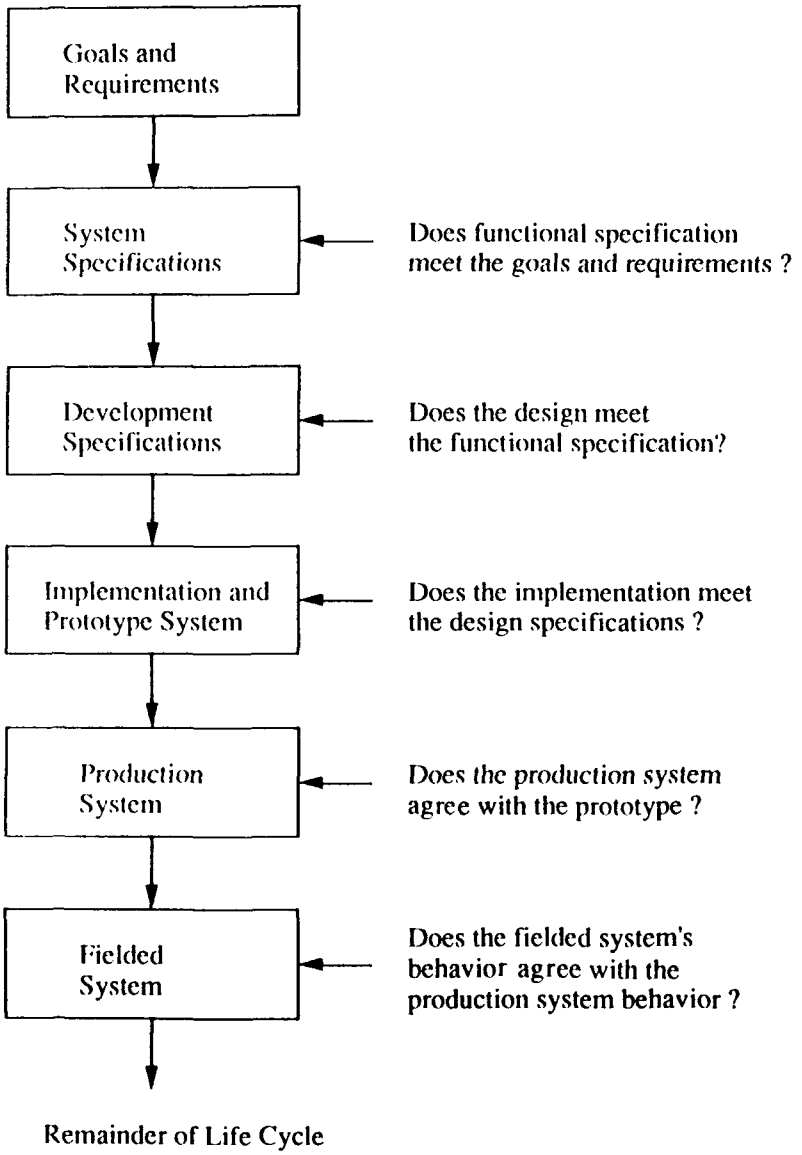


Figure 3.1 - Digital System Life Cycle

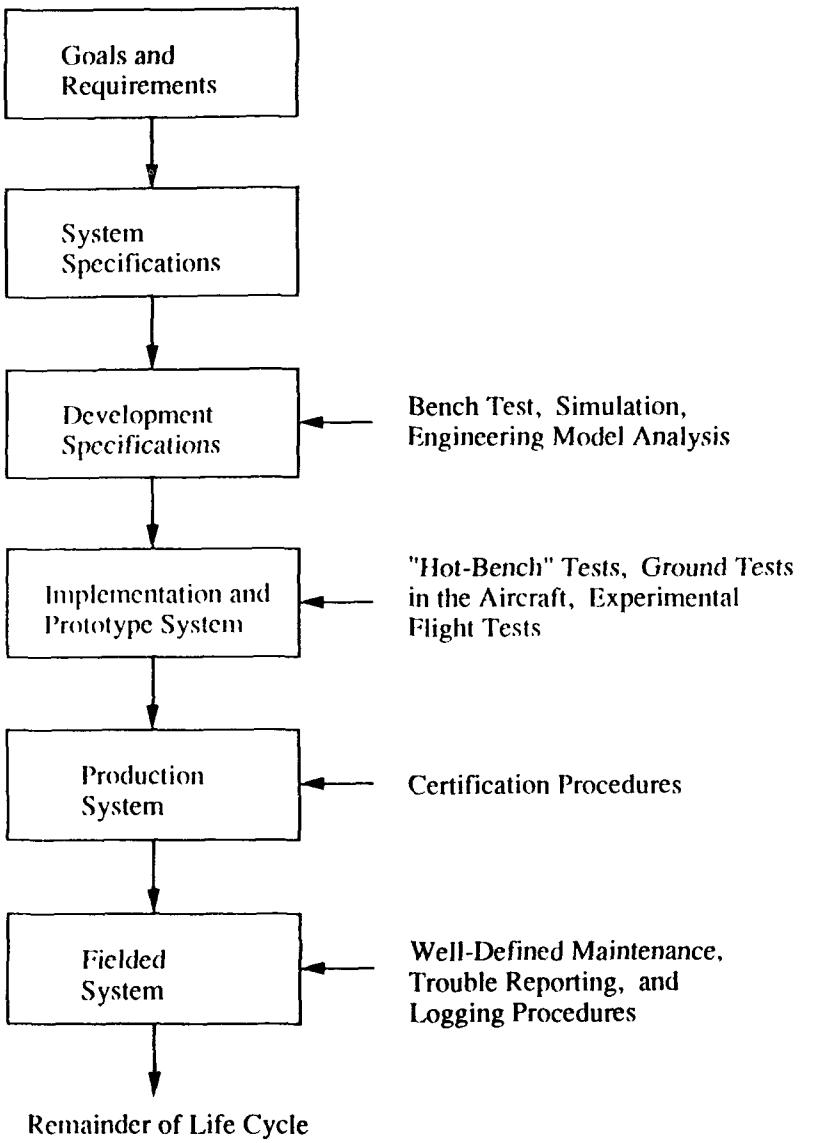


Figure 3.2 - Digital System Life Cycle Applied to Aircraft Systems

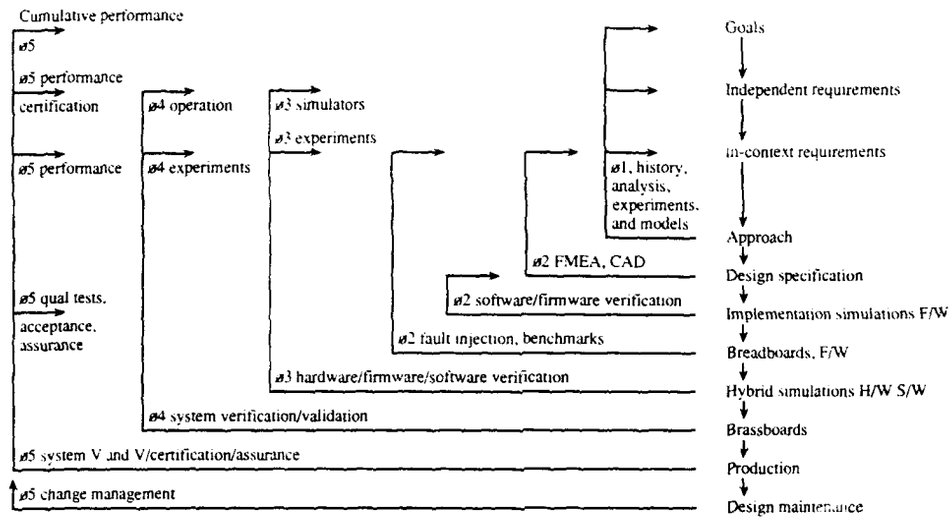


Figure 3.3 Sequential Content of Validation Process

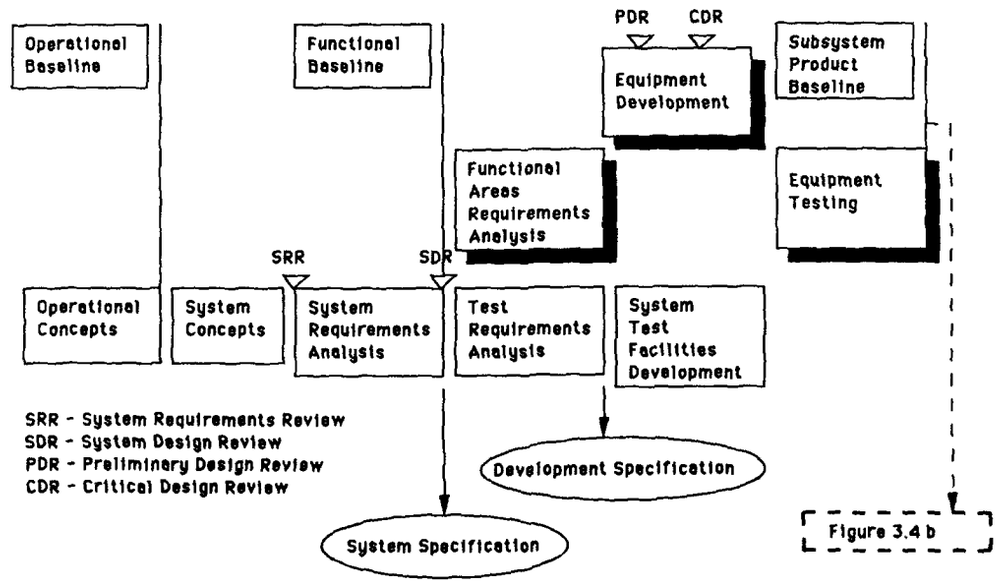


Figure 3.4 a Development and Documentation Process for a Validation Program

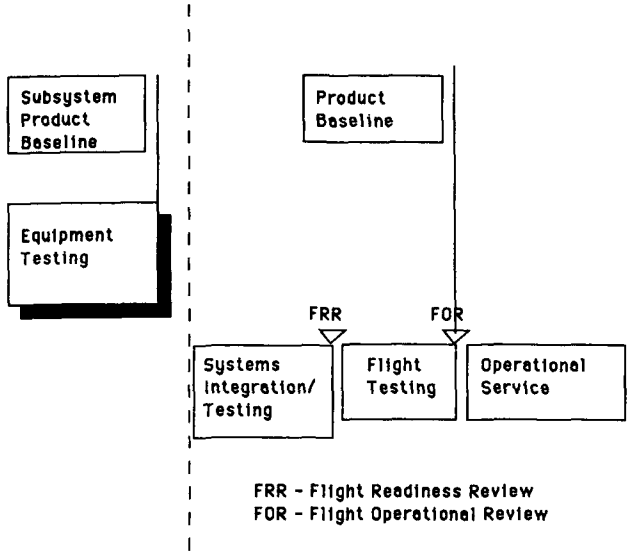


Figure 3.4 b Development and Documentation Process for a Validation Program, (continued)

Technical Requirements

1. Applicable Documents
2. Functional Characteristics
   - 2.1 Control Law Requirements
   - 2.2 Integrity/Flight Safety Requirements
   - 2.3 Architecture
   - 2.4 Reliability
   - 2.5 Maintainability
   - 2.6 Built in Test
3. Interface Definition
   - 3.1 Electrical Signal Interface
   - 3.2 Electrical Power Interface
   - 3.3 Hydraulic Interface
4. Design and Construction
5. Software
   - 5.1 General
   - 5.2 Software Development Process
   - 5.3 Documentation
   - 5.4 Development Facilities
   - 5.5 Q.A.
6. Test
   - 6.1 Development Tests
   - 6.2 Environmental Tests
   - 6.3 Qualification Tests
   - 6.4 Functional Tests
   - 6.5 Acceptance Tests
7. Environment

Figure 3.5 Functional Specification for Safety Critical Issues

1. Software Specification
2. Redundancy Management Specification
3. Integrity Requirements
4. Hardware Design
5. Reliability Requirements
6. Maintenance
7. Operational
8. Human Factors
9. Environmental Test Requirements

Figure 3.6 List of Typical Design Requirements

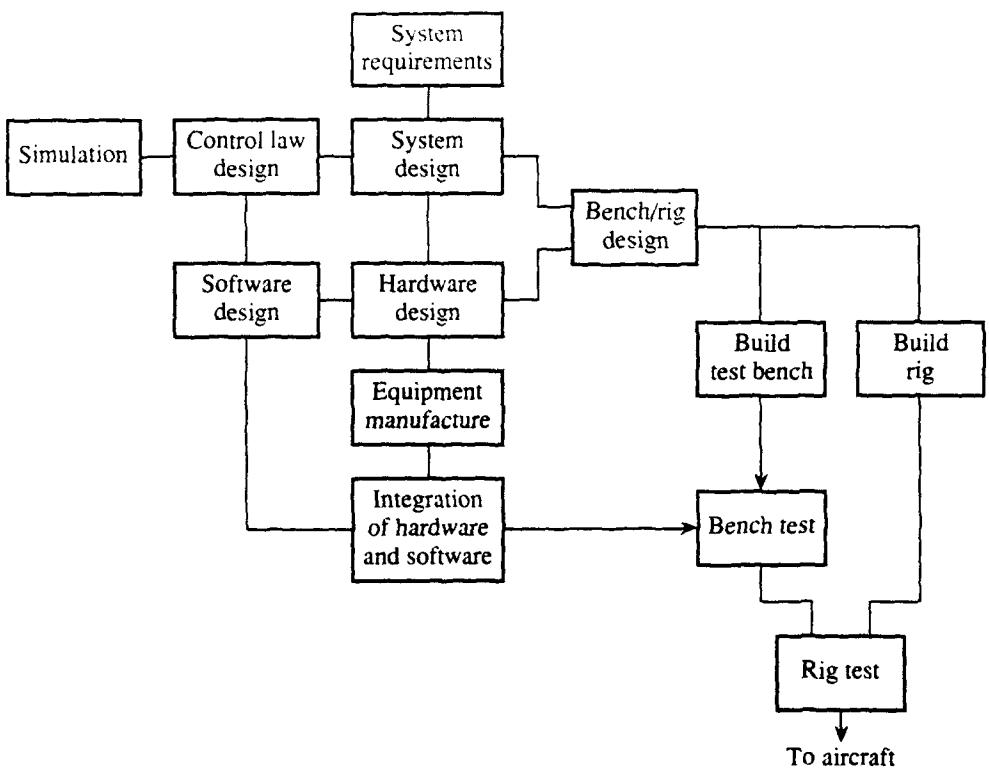


Figure 3.7 Activities Leading to Prototype Implementation

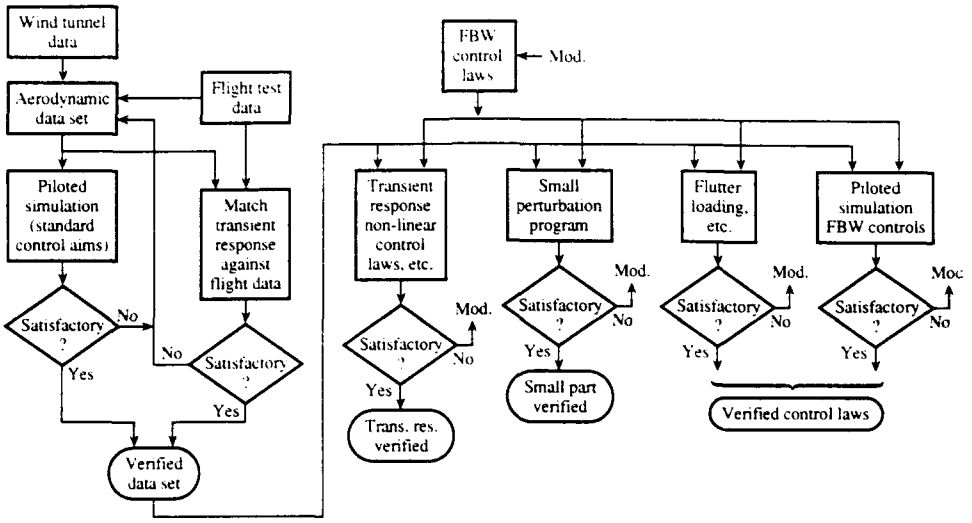


Figure 3.8 Control Law Preparation and Validation

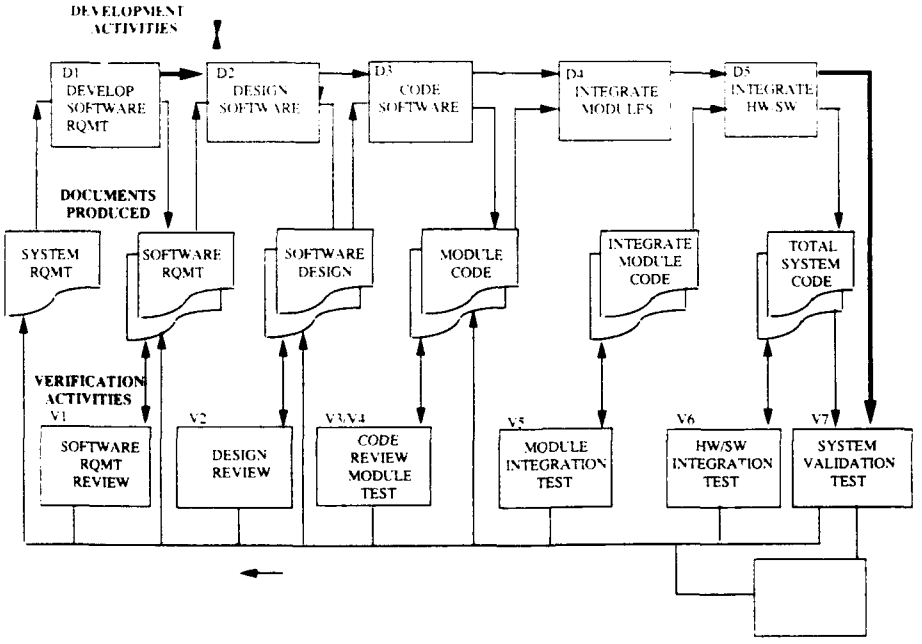


Figure 3.9 Software Development and Verification Activities

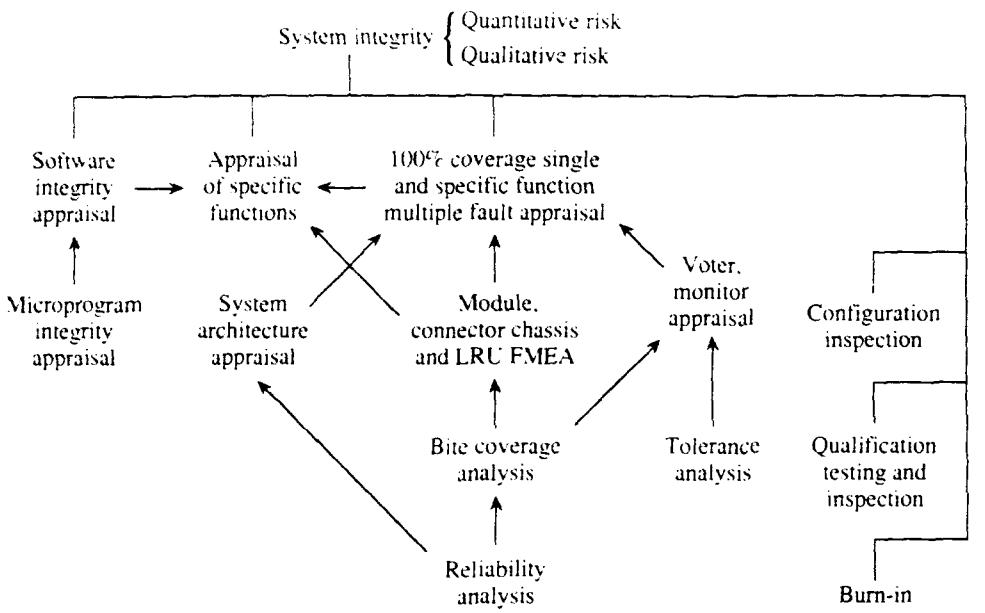


Figure 3.10 Ensuring System Integrity

CHAPTER 4

CURRENT METHODOLOGIES AND TECHNIQUES

4.1 Introduction

The purpose of this chapter is to present current methodologies and techniques used to validate a flight critical system. Assessments of individual validation methodologies are also included. In Chapter 5, an assessment is made of the overall state of the art of validation of flight critical flight control systems.

Validation begins in the system requirements analysis phase and continues through the development phase and culminates in the demonstration that the final system complies with the system-level requirements defined prior to start of development. Figures 4.1 to 4.5 show a flight control system development cycle based on European and U.S. practice

There are four basic types of validation activities:

Inspection - used for determining if the product (software, hardware, or integrated software/hardware) as built, conforms to the applicable documentation, such as engineering drawings, flow diagrams, computer listings, user requirements, system specifications, etc. Inspection typically involves visual/physical examination or simple measurements.

Test - used to establish that the product functional characteristics conform to operational and technical requirements. The process has a high technical content, and usually requires specialized test equipment and formalized procedures. The product under test is stimulated with inputs to generate controlled responses which can be compared with predictions. Data generated by test is further analyzed to determine conformance with the criteria. Test passage can either be go-no-go, or be a result which falls within criteria boundaries.

Demonstration - used to show the customer and/or legal authority that the product functions as required within the operational envelope. Test passage is usually based on go-no-go criteria established by the reviewing authority.

Analysis - used to show compliance with requirements, either to complement test, or to replace test when test is not possible or practical. Data output is from simulation and analytic models. An important, but often difficult task is the validation of the models used in the analysis.

4.2 Development of the Customer Requirements Specification

The initial step in the development of an aircraft is the formulation of the customer's requirements which define what is required of the aircraft. (The terms operational baseline, functional baseline, allocated baseline and product baseline are taken from Mil Std 2167). The user will usually have conducted a series of studies which will have established the major characteristics of the aircraft and its systems to enable the aircraft to meet its operational goals. Normally these characteristics will be contained within the statements of need/staff targets, and later in the Program Management Directives/staff requirements which are the formal statements of the Operational Baseline. They are top-level requirements and do not prejudge how the system will be configured to achieve the requirement.

The Operational Baseline for the aircraft is usually written in plain language and often is based on previously defined operational requirements of the user. An assessment is made of the reasonableness of

the Operational Baseline, based on previous systems and experience, as well as on knowledge of the state of the art of the various technologies to be incorporated in the new vehicle and its systems.

The operational baseline for the aircraft will include the following aspects which are directly relevant to the flight control system.

a. the requirements for the handling qualities, ride qualities, and control characteristics

b. the definition of the flight envelope

c. the modes of the FCS required

d. the reliability, safety, and availability requirements

e. the maintainability and testability requirements

f. the growth potential required

g. the methods and standards which must be applied.

Since the Operational Baseline is a list of the customer's requirements it is necessary to ensure that all these requirements are captured and combined with requirements from other viewpoints (e.g. design, manufacture, and certification), and then checked for completeness, consistency and traceability. Methods and computer-based tools are emerging to carry out these tasks in a systematic way. They model the requirements, the interactions between them, and maintain traceability with subsequent design stages.

4.3 Development of the Weapon System Specification

The next stage in the process is the development of the Weapons System Specification. The customer's requirements are studied and combined with candidate solutions in a series of concept development studies, analyses and reviews.

While the emphasis in this phase is on the requirements, in some areas the requirements will be matched with a candidate design, and it is inevitable and desirable that the requirements and design be iterated to produce a system design that is feasible and meets customer requirements.

The designers will study various configurations of aircraft and systems to determine how best to meet the requirements. Extensive studies involving all aspects of the system will be made and may require development programs and prototyping.

These studies will include a preliminary assessment of the FCS functions and the major characteristics of the FCS, e.g. FBW or mechanical controls, authority levels, safety, reliability, availability, interface with other sub-systems.

The result of this phase is the material for the Systems Requirement Review which will validate that the requirements in the Operational Baseline have been transformed and elaborated correctly in the System Requirements documents. It is crucial that 100% traceability is maintained during this process, because of the significant impact that errors at this point can have on the subsequent system design process.

Studies will include an assessment of the technical risk of the configuration and the technology required for implementation. The studies will determine what further development is required.

4.4 Development of the FCS Requirements Specification

The FCS Requirements Specification (FCS RS) is the entry to the FCS life cycle. It is formed by an analysis (e.g. functional decomposition) of the weapon system specification (WSS); it defines the functions performance needed from the FCS to meet the WSS.

The FCS Requirements Specification is one of the series of sub-systems requirements specifications,

each derived from the WSS; others include the Hydraulics Supply System Requirements, the Electrical System Requirements, etc.

During the development of the FCS RS studies will be made of the interaction between the airframe and the FCS to establish the levels of performance required, e.g. the level of instability that the FCS can compensate (see Figure 4.6)

The FCS RS is validated by checking the functions specified in the FCS RS against requirements in the WSS and in the customer requirements.

Another form of validation is made by comparing the functions specified against those used on previous projects and those developed in previous research and development programs. This limits the technical risk by reducing the technology increment. The validation activities will include a series of review meetings between system design specialists to ensure that the various subsystem requirements specifications are compatible.

4.5 Development of the FCS System Specification, the Control Law Design Specification and the System Quality Plan

One method of proceeding with the development of a system to meet the FCS requirements is to split the process into 3 major activities: Control Law Development, FCS Development and System Qualification (see Figure 4.2). Each of these activities is controlled by a document derived from the FCS Requirement Specification.

The control law development is specified in the Control Law Design Specification which groups together and amplifies what is required from the control laws and outlines the form of the design. This document is the basis for the control law development as detailed in the control law development plan.

The FCS System Specification outlines the total FCS. It defines the functions produced by the system and the units which comprise the system.

The specification is the result of past experience, research, development and studies and analyses to find architectures, technologies and equipments which will meet the multiple requirements placed on the FCS. It will be influenced by the requirements of other systems, by the performance required, by resources, time scales and perceived technical risk.

Techniques used in this phase include:

- Hazard Analysis to determine the integrity needed in the various elements.
- Reliability Analysis to determine the reliability levels needed from the individual elements to meet overall targets.
- Preliminary FMEA to determine the effect of failures of individual components.

Validation activities in this phase check that the FCS System Specification does "flow down" the requirements from the FCS RS and that the resulting design is capable of meeting the customer requirements. Independent assessment of the performance, integrity and reliability may be applied to validate the FCS System Specification.

The System Qualification Plan defines how the system will be qualified and includes the requirements that have to be satisfied, the methods used to satisfy them, and the support facilities required. This Plan will lead to the development of all the rigs, test benches, test equipment, and support environments which will be required to accept, test and integrate the equipments which comprise the FCS.

4.6 Development of the Requirements Specifications for Processing, Sensors and Actuation

Once the flight control system specification has been produced and verified, the next phase is to refine the requirements and the design.

One method of organizing this phase which has proved to be successful is to convert the FCS System Specification into the three major areas of Processing, Sensing and Actuation. This is shown in Figure 4.7.

A preliminary design is developed which has an initial hardware layout together with an allocation of functions to the three sub-systems. The characteristics of the resulting integrated system is determined, including the safety and reliability. The design is then checked to determine compliance against the FCS System Specification and the procedure is iterated until a satisfactory design/allocation is found.

As systems become more integrated there is a major risk when moving from these functional specifications to equipment specifications since the functionality is determined by several different pieces of equipment, and the effect of the combined tolerances in all the equipments has to be determined.

Reviews

It is important to note that the reviews at this stage are at a systems level and at a degree of abstraction which prevents detailed assessments or analysis. This tends to obscure some requirements and make it difficult to effectively review and critique those aspects of the system. It is important to include systems engineering staff with broad experience in these reviews, in addition to design specialists, because the system must be assessed at a level of abstraction which makes experience comparisons a key technique in validating requirements. The baseline configuration validated at these reviews will become the formal requirement for the system. It will cover all aspects including integrity, functionality, environment, performance, modes of operation, reversion capabilities, and methods of validation.

4.7 Development of the Processing Sub-Systems

One approach which has proved successful, and has become standard practice is to split the processing subsystem into the three principal activities, namely hardware, software and test equipment (see Figure 4.3). The development of the specifications for these three subsystems is based on an iterative process of design and analysis in which draft specifications are produced, outline designs are developed, and analyses/tests are made to validate the subsystems specifications and the design against the system specification.

4.7.1 Development of the Hardware

The development of hardware for flight critical systems follows normal flight system hardware design practice with extra attention being paid to the rigor and detail of the design and to the validation of performance and operation.

Hardware Specification

The major items in the hardware specifications are shown in Table 4.1. The validation of this specification will be routine in most areas; the detailed requirements will be checked for compatibility with the overall requirements as defined in the FCS system specification. However, in two areas of validation, redundancy management and functional performance, special care is needed. In these areas the functional capability required from the system depends on interlinked capability of the hardware and the software. Thus it is necessary to validate a matched set of subsystem requirements. This requirement for interlinking can be formulated in a separate specification of the overall capability at a more detailed level than the flight control system specification and then the subsystem can be validated against it.

Reducing Abstraction

While even at this stage, it is possible to leave the requirements general, and delay design until the next stage, it is often better to base the subsystem specification on a specific implementation. This improves clarity by removing abstraction and makes it easier to detect conflicts between requirements.

Validation Analyses

Validation techniques used at this stage to check the integrity of the system include FMEA, simulation, reviews, and Fault Tree Analyses (FTA). To quantify this analysis, these calculations are based on component failure data such as Mil Hdbk 217.

Test Equipment Requirement

The hardware subsystem specification should define the test injection and monitoring equipment which will be required to verify that the system is performing correctly in normal and under fault conditions.

Purpose of Hardware Validation

The validation tests and analyses check the the following items against the subsystem specification:

Performance

- correct execution of the instruction set by the processor
- correct input-output data handling
- correct operation of the RAM and ROM functions

Integrity

- functionality of the redundancy management logic features implemented in hardware
- ability of the system to contain failures as predicted in the FMEA to meet the safety, availability and maintenance requirements
- operation of the unit under normal and extreme conditions
- BITE coverage as required for integrity, as per the FMEA
- BITE coverage required for maintainability

Description of Hardware Validation

Representative hardware is tested in a controlled environment. A wide variety of tests are conducted using special test software to check the operation of the hardware in nominal and extreme environments with inputs at nominal values and at extremes of the operating range. Sensitivities are measured and tolerances checked against the subsystem specification or against a new set of tolerance criteria resulting from analyses of the integrated system components. Operating characteristics especially checked include:

- Correctness of operation
- Speed of operation
- Sensitivity to power supply variations

- Thermal profile of the design
- Mechanical integrity of the design
- Range of signal inputs
- Range of signal outputs
- Interface compatibility

Support Facilities

In order to support the development of the subsystems, specialized test and support facilities and equipment are required. Specifically, the support facilities must include:

- Hardware representative of the flight system (breadboard or brassboard)
- Software to exercise the computers with the facilities to change the software to aid diagnosis
- Test equipment capable of simulating the external environment and capable of generating simulated failures in the computer. (A hot bench)
- General test equipment (logic analyzers, emulators, bus analyzers, scopes)
- Environmental test facilities for altitude, thermal, vibration, and Electromagnetic Interference (EMI)

4.7.2 Development of the Operational Flight Program

Software Specification

The major items in a software specification are shown in Table 4.2. Software specifications may be produced using a program design language or a high order language to facilitiate the communication of design intent, and to provide a systematic set of specifications.

Procedures/Methods

Since techniques have not yet been developed to quantify the probability of occurrence of software error in a particular set of codes, it is necessary to control software reliability by defining the methods which must be used to produce it., hence the emphasis in the specifications on standards, procedures and structures. Figure 4.8 shows the method used for civil aircraft as specified in DO 178A. There is now a wide range of design methods available to support software development and the subsystem specification can be checked for compliance with the requirements of an appropriate method.

Software Testing

Testing plays a central role in the validation of software and the subsystem specification should be sufficiently detailed to ensure that the appropriate level of testing has been specified. Where the validation of the subsystem software has been well thought out in advance, entry and monitor points can be built into the software with little overhead to support subsequent testing.

Development of Software

The development of software involves the gradual refinement of the requirements within the specified structure until code is produced. That code is produced as modules which are then thoroughly tested and integrated into subprograms and finally into the total program.

Description of Software Validation Process

The two basic techniques used to validate software are extensive testing, and rigorous construction. Since 100% testing is impractical, the rigorous construction methods are a vital part of the process of developing software for safety critical systems. Most of the safety critical systems developed in the past have used a combination of rigorous construction and extensive testing . The current trends in software validation are automated testing and the application of more formal methods.

Validation by Testing

Testing is done at various stages;

- Module level (on the order of 100 lines of code)
- Subprogram level (on the order of 40 modules)
- Program level (using a simulator on a host computer)
- Hardware/Software integration (using representative hardware)
- Rig, or iron bird (Representative hardware with representative interfaces)
- Aircraft on ground
- Flight test

Module Level Test

Code reviews and module tests are used to check that the modules operate according to the module specifications. Static analysis tools can check:

- control flow
- data flow
- information flow

The verification of the modules will include:

- tests to ensure correct operation with a limited data set andanalyses to show that the set is representative.
- analyses to detect banned construction/instruction.
- analyses to ensure correct use of memory.
- analysis/testing to ensure operation within time budget.
- analysis of correct entry/exit procedures
- tests of robustness for out-of-limits data

In addition, some subsets of the software can be validated using formal proof methods.

Program/Subprogram level test

Software integration tests are used to check that the modules, when integrated into programs or subprograms, operate according to the subsystem software specification. Validation at subprogram level will check the compatibility of the modules and in particular will check the flow of data between modules

and the overall control of the modules.

The other levels of testing involve hardware. The tests are structured to ensure that the software is checked in all its modes, with a large set of representative data, as detailed in the following sections.

Program Validation

Once the whole program has been assembled it will be validated by testing and by analysis against the functional requirements as given in the subsystem specifications. These tests will include checks to ensure that the software detects faults, isolates them and reconfigures the system as required by the redundancy management.

Automatic Testing of Software

The large number of data sets which have to be produced to check out a system can be generated automatically using special test software which usually resides in the test support system. The automatic test software may also contain a model of the functional requirements and use that model to generate answers and tolerances. Automated testing facilities are used which calculate the inputs to be injected, calculate the required results and tolerances and produce test reports which highlight failures. This technique is important to reduce schedule delays and costs particularly after modification.

Testing on the Host

Initially the software will be tested remote from the flight hardware using a host facility. Typically a minicomputer is used with a specially developed test harness which allows the progr.. to check out the software modules with a thorough set of test cases, then to integrate the modules into subprograms and test them at that level. The final stage on the host is to assemble the complete program and test it.

The level at which the testing can be done depends on the size of the software. Simulators which check operation at register level are inevitably slow (typically 100 times real time) and the test data produced is immense.The combination of text values required to check a system forces the engineer to split the task into parts which have a much smaller number of inputs and outputs. He then tests those parts invidually and then tests thelinks between the parts. Thus the fine grain operations of the system can be tested. Clearly this form of testing does not replace the need to test the total system which gives an overall but coarser grained text. Hence the emphasis on testing to provide proven modules and then tests to check correct interaction.

Validation by Rigorous Construction

Many of the aids to validation are based on the premise that it is better to show correctness by ensuring that rigorous methods are used in the construction of the software.

The techniques used in this type of validation include:

- Formal specification (Defining the requirements and the sytem in annotation based on mathematical methods, and then proving the system meets the requirements)

- Structured Analysis (Breaking down the requirements using a structured method)

- Static Analysis (Examination of the code to ensure sound programming rules are followed)

- Program Description Language (Using a "formal" language to define requirements)

- Dynamic Analysis Configuration Control (testing by exercising the code)

Support Facilities

The principal support facilities in this stage of development are principally those related to the software support environment:

- editor
- linker
- loader
- assembler
- compiler
- debugger
- simulator
- test harness
- test case generators
- static test tools
- semantic checkers
- proof checkers
- test coverage analyzers

4.7.3 Development of the LRI Test Facility

The development and validation of test equipment follows closely the process for the development of flight hardware with the appropriate modification for the different environmental, cost, and integrity requirements. The validation of the test equipment consists of tests and analyses to establish that performance is correct and is representative of the real world it simulates.

Test Facility Specification

The principal items in a test equipment specification are very similar to those of the hardware specification given in Table 4.1; in addition, the test equipment may have its own software which will require specification as contained in Table 4.2

The subsystem specification will define the test injection and monitoring equipment which is required to validate system performance in normal and under fault conditions.

The validation of these specifications will be by comparison with the FCS specification and with the specifications of those units which it is simulating. The latter will require comparison of performance figures or analyses to demonstrate equivalence.

Software for the Test Equipment

The development of the software for the test equipment is validated as described in Par. 4.7.2 where appropriate. Where the test equipment software does not impact the integrity of the system, then the test analyses are restricted to those necessary to ensure test equipment performance. An emerging practice is to maintain configuration control of the test equipment and software to the same level as that done for flight systems, often using the same configuration control boards and documentation. This is done in recognition of the criticality of maintaining the integrity of the systems used to validate flight critical systems.

Simulating the Interfaces

The test equipment required for flight critical systems consists of interfaces, processing sections, monitoring facilities and test injection facilities. It has to simulate the interfaces in a sufficiently representative way to make rigorous testing possible.

One important area is the fidelity of simulation of the test rigs. Test rigs must simulate aircraft systems external to the flight control system. The accuracy of this simulation will influence the accuracy of the results of the tests. Special tests and test rigs to check the simulation may be needed, e.g., loads simulation for actuators, motion simulation for sensors, pneumatic simulation and bus traffic simulation.

4.8 Integrating the Hardware and Software

Once the hardware, software and test equipment have been proved as separate systems they are integrated to check compatibility. A series of tests are undertaken to check the characteristics listed in Table 4.3 against the system and subsystem specification.

Representative software is fitted into representative hardware. The computers are then connected to a test bench which supplies power, simulates the external environment and provides monitoring and recording equipment. A typical hot bench configuration is shown in Figure 4.9.

The operation of the LRI is then validated by tests based on the structure of the computer and by tests based on the functions required of the system.

Support Facilities Required

- Hardware representative of the flight system (breadboards, brassboards)

- Software representative of the flight system to operate the computers but with extra capability to aid diagnosis.

- Test equipment capable of simulating the external environment and capable of generating simulated failures in the computer.

- General test equipment (logic analyzers, emulators, bus analyzers, scopes)

- Software development environment

- Editor, linker, loader, assembler, compiler, debugger, simulator, test harness, etc

- Host computer with facilities to interact with the target computer

- Software analysis tools

- Facilities to change the target computer memory

Assessment

As is the case for the hardware-alone validation, the integrated hardware/software tests are a powerful method of checking that computers operate as per the design specification. The integrated hardware/software tests involve a lot of equipment, some of which may be still undergoing change. Hence it is often difficult to separate problems and double faults are much more difficult to resolve than single faults. These integrated tests may be used to provide the evidence to support some of the more difficult analyses, particularly the FMEA and FTA.

The effectiveness of the BITE may be checked by a series of representative tests, which provides support for the predicted coverage values, and also demonstrates the annunciation of detected faults.

Emulation

During the integration of the hardware and software it is often necessary to check the operation of the processor at a detailed level (e.g., internal register transfers). In many processors, access is not available to that information without adding special software. However, hardware emulators are available which simulate the operation of the processor and give the engineer visibility into the internal functioning of the processor. This is a valuable aid to testing system operation and may be used to validate hardware architecture as well as detailed software operation.

Power Transients

One of the major difficulties in fly-by-wire systems has been the problem of handling power interrupts. The basic problem is that the duration of the power interrupts which was acceptable for older generation systems is larger than the duration of the "hold up" capacity that can be built into the units. Thus special power supply systems with shorter power interrupts have been developed. However, there are situations in which power may be lost for sufficiently long periods that computer operation stops or becomes unreliable. To cope with such situations special circuits and software are designed into the systems. These mechanisms must be specifically validated during the hardware/software integration tests .

4.9 Integrating the LRI's to Form a System

The process of integrating the LRI's of the FCS is a gradual one which can be split into phases of increasing system coverage. Figure 4.10 illustrates one method and shows how LRI's are accepted, integrated with other LRI's and then tested on an iron bird. It also shows the specifications against which validation is performed and the phasing of open loop and closed loop testing.

Configuration Control

Strict configuration control must be established and maintained for the flight-qualified articles, including all software, as well as test stations and the aircraft system configuration. Identification of all components not qualified for flight but needed for testing must be traceable so that they can be tracked and replaced with safety of flight (SOF) rated components before first flight. For flight critical elements, 100% traceability of all actions taken regarding these systems is required. Discrepancy reports, test reports, change requests, change documentation, qualification test results, and clearance for flight must all be tracked precisely so that equipment or software not cleared for flight use is not able to creep into the flight system.

Integration of Subsystem Interfaces

Each individual interface within the overall flight-critical system must be validated in the proper environment prior to testing the entire system. Examples of the significant interfaces that must be exercised include the computer/actuator interfaces and the interfaces between sensors, controllers, and computers.

Computer/Actuator Interface Testing

Actuator Integration Test Stations which are powered by appropriate hydraulic pressure and capacity and an electrical signal which is conditioned by and compatible with the computer is essential in risk reduction. This is to insure that the actuators and control surfaces perform as required and in conjunction with the computer, electrical, and hydraulic power prior to the actual integration of these components on the aircraft. The actuator control logic must be used in conjunction with the built-in-test development to assure that it is working before the integration begins. Figure 4.11 illustrates the actuation subsystem validation interfaces. The essential actuation system validation task is the definition of actual loads on the control surface which affect surface deflection and rates. The solid arrow lines indicate the validation boundaries which can be accomplished on the ground. The dashed arrow lines indicates the validation task that must be accomplished in flight.

Sensor/Controller/Computer Interfaces

The physical design and mounting of the sensors and controllers must assure that each of these units is rigidly supported by aircraft structure and are serviced by appropriate electrical control signals. Provisions to check the built-in-test must be worked in conjunction with the computer hardware and software logic. Developing this capability early is essential so that it can be used to diagnose problems that will be encountered during the integration process on the aircraft. Fig ure 4.12 shows the validation structure for an air data probe, and Fig 4.13 shows the structure for a rate gyro/accelerometer assembly.

In the case of a sensor subsystem, such as an air data probe, the challenge in the validation process is to

provide a representative stimulus to the system on the ground. Ground models cannot adequately represent the flow field actually encountered in flight, with current technology. The solid arrows show validation steps that can be accomplished on the ground; the dashed arrows show the validation that must be accomplished in flight.

Iron Bird development

The extent and actual design philosophy of the "iron bird" is very much system design configuration dependent. The iron bird is an integration tool that permits early resolution of certain types of problems related to hardware mechanical and structural arrangements. The iron bird typically includes dynamically faithful such control surfaces and actuation connected with control elements and hydraulic and electrical power supplies. Figure 4.14 shows a typical iron bird arrangement and its interfaces. An iron bird laboratory usually has the following features:

- interface to an avionics laboratory via a data bus
- 3-axis rate table for gyro stimulation
- aircraft or ground hydraulic pumps
- landing gear rig area
- flight control computers with test sets
- actuator test bench area
- aircraft -level electrical power supplies
- control room for iron bird set-up
- stimulation, monitoring, and recording equipment
- software support environment

Pilot-in-the-Loop Hot Bench System Simulation

This is an important simulation tool to check-out and test interactive elements that can affect parameters, such as time delays, which in turn can affect the pilot's ability to adequately control the aircraft. The term hot bench usually refers to the use of actual flight computers with embedded flight software. Modern simulation and iron bird systems usually provide for the capability to use the iron bird cockpit with flight computers and a simulated actuation system. Separate cockpits may be used with flight computer hot bench configurations as well.

4.10 On-Aircraft System Integration

One of the most time critical test phases is the on-aircraft systems integration. It is important to have already identified and resolved each problem that is likely to occur during these tests because of the time criticality and cost of fixing problems so close to first flight. Conducting tests on the aircraft is also difficult because of the limited access to internal information. Examples of important tasks during this time period include:

a. Aircraft Equipment Installation check-out of all components,

b. bore sighting and/or alignment of all sensors,

c. structural coupling testing and validation,

d. EMI/EMC validation,

e. ground/aircraft power compatibility

f. control surface rigging

g. validation of compatibility with the environmental control system

One facility, the NASA Dryden Integrated Test Facility, is under development which will allow a relatively high degree of integration of the actual aircraft and a ground test facility. Figure 4.15 shows the aircraft-facility interface in this facility.

4.11 Clearance of the FCS for the First Flight

Prior to flight, one must assure that the aircraft is safe to fly, that the instrumentation system, data transmission and processing systems, and ground monitoring equipment is ready to support the flight, and that all the flight plans have been simulated on the ground, and have been formulated to deal with any contingencies.

Special instrumentation is often designed specifically to support flight testing. It may be required to get more visibility into internal operation of various subsystems during flight test, such as near-failure declarations by the failure handling systems, so that margins on failure thresholds may be determined under actual flight conditions.

Functional Configuration Audit

It is prudent that an independent audit be made of the adequacy of the design to safely accomplish the intended mission. The audit team must be knowledgeable of the technical details of the design and must specify well in advance the tasks for the design/development team to demonstrate prior to the completion of the audit. The functional configuration audit reviews the production system baseline design by comparing performance validation test results with system requirements, and the results are documented through a validation cross-reference index.

Physical Configuration Audit

It is equally important that an independent audit be conducted to demonstrate that the specified design is implemented properly on the flight article. Each system component on the aircraft must be certified as flight worthy. The physical configuration audit reviews the production system baseline and confirms the design through traceability of documentation from the drawing to the hardware part number.

First Flight Readiness Review

The aircraft is determined to be safe to fly based on analysis of the ground test data and system analyses, the findings of a flight readiness review (FRR). The project team analyzes the technical data, the various hazards analyses, and independent analyses to establish that the aircraft is technically ready to fly. The flight readiness review is conducted by a group of independent experts and those with broader experience. This group is unassociated with the project, and takes one last look at all of the technical and management processes that have been used to arrive at the flight-ready aircraft. Their declaration of flight worthiness is required prior to release of the aircraft for first flight. Flight readiness certification documents are usually completed after this review.

The initial premise that each subsystem be already matured will most probably not be realized, and the subsystems will, in fact, continue to be refined during the entire course of this phase of the integration and development effort of the flight critical system. Changes produced by each of these overlapping tasks will require consideration by the other.

4.12 Flight Test

Background

Flight test is the culmination of all previous verification and validation testing conducted up to that point in the system's development.

Development flight testing should test to the limits of expected operational capability and should perform all tasks expected of the system operationally in a controlled test environment prior to releasing the system to the operational community. In the test environment tasks and maneuvers can be conducted in a build-up fashion under controlled conditions and monitored to optimize safety of the aircraft.

Any system is finally evaluated in the field doing the job it was designed to do. This is approximated in flight test during operational testing when operational crews conduct simulated operational missions using the test system. The tasks performed during this phase have been already accomplished in the development flight test phase. Accomplishing the operational tests integrates many tasks for the first time and enables identification of human factor and interaction problems that make operation of the system by non-test personnel difficult.

The ultimate objective of the flight test program is to determine that the integrated system will perform the mission for which it was designed and developed. It is pertinent to note that developmental flight tests usually have to be conducted prior to being able to fully validate the aircraft as a weapon system. Since safety of flight during the validation stage is insured by the testing conducted during verification, a short paragraph on verification testing is included below.

An FCF, or short series of flights, is flown with any new or modified aircraft to verify proper operation of the basic aircraft systems and assure the test community that the aircraft is safe prior to proceeding with the test flights. Preparing for the FCF is similar to preparing for the rest of the flight test program. The FCF is one flight or set of flights with emphasis on safety, and on exercising the various airframe, propulsion, and cockpit systems. Prior to flight, a test plan or profile is defined and critical data are identified for monitoring during flight.

The requirements for measurements, instrumentation, and data processing are described in detail in the flight test plan. The test plan is usually a dynamic document subject to frequent updating during testing. It describes, in detail, the objectives of the tests, the tests that are to be performed, how they are to be performed, the instrumentation parameters needed to analyze each maneuver, the analyses and displays required, and the parameters that need to be monitored in real-time to insure safe test conduct.

Initial Flight Tests

Initial flight tests are conducted to assure compliance with specifications and validation flight tests are conducted to demonstrate that the requirements for which the system was built are satisfied. A list of test objectives for critical system tests should include the following:

a. Determine that all modes operate as designed.

b. Evaluate control system/structural interaction, i.e., the aeroservoelasticity characteristics.

c. Evaluate the engage and disengage transients and mode change characteristics.

d. Evaluate flying qualities with the system engaged. Include all modes of the control system including back-up and reversionary modes.

e. Monitor preliminary system reliability, availability, and maintainability.

Detailed Flight Test Program

The detailed flight test program of a new aircraft usually involves several vehicles, with dedicated test objectives. An example of such a vehicle set is shown below:

Aircraft #1 Basic aircraft systems

Flutter clearance
Fligfht Controls/Handling qualities
Initial aircraft performance

Aircraft #2 Loads

Aircraft #3 Avionics
Integrated systems tests
Weapons tests

Envelope expansion, stability and control (S&C), handling qualities, aircraft performance, system tests, integration, loads, flutter, high angle of attack, weapon separation, and weapons accuracy tests determine compliance with specifications. They are generally tests of the integrated aircraft and subsystems and are determining indicators of how the aircraft will perform its design mission.

a. Envelope expansion testing extends the flight envelope in Mach number, angle of attack, altitude, and load factor to the design limits or until a performance limiting condition is reached.

b. Stability and control testing is the quantification of the aerodynamic forces and moments. In modern aircraft this involves both the active controls and the basic aerodynamics of the airframe and control surfaces. Stability and control derivatives are obtained using parameter identification techniques. [1]

c. Handling qualities are the subjective manifestation of stability and control as evaluated by the subjective evaluations of pilots. Quasi-quantification of the subjective ratings are obtained using the Cooper-Harper rating scale. Effective use of the Cooper-Harper scale[2] depends on a careful definition of the task, the method of accomplishing the task, and the criteria for judging the ability of the pilot to fly the task successfully. Handling qualities during tracking (HQDT) is another quantification of a special case of handling qualities. The pilot tightly closes the loop while tracking an airborne target, and data are obtained to show how closely he is able to accomplish the tracking task. A plot of deviation in pitch and yaw angles referenced to the target, graphically display the ability of the pilot to track the target.

d. Loads and flutter testing investigate the airframe's ability to withstand the flight environment both statically and dynamically. Emphasis is on the structural response to the environment.

In other integrated validation testing, the aircraft must perform to the extremes of its design envelope. High angle of attack, departure boundaries, post-departure handling qualities, and basic aircraft agility all relate to the ability of the aircraft to get the job done in the operational scenario. High duty cycle system performance, robustness, and emergency system operation must be evaluated and found to be adequate to the task of accomplishing the design mission. Mission accomplishment remains the ultimate validation.

4.13 PRODUCTION SYSTEM VALIDATION

Introduction

In this section productionisation is defined to be the transition from prototype to production aircraft and includes the completion of the qualification test process. During the development phase there will be modifications due to specification evolution or changes to correct initial design errors. When modifications have been made then, prior to production, the system will have to be revalidated by revised analysis and revised testing.

For the production phase it is also necessary to have definitive acceptance test procedures which have to be carried out on each item of equipment, each system and each aircraft.

Quality Plan for Production

This plan defines the process to be followed in productionisation, in particular:

- The configuration management

- The modification management

It also defines the documents to be updated following modification.

Configuration Management

During productionisation the configuration of the production standard is frozen. For each item a definitive list of components is necessary which defines totally the final product. The rules and procedures define how a configuration may change, who is allowed to initiate modifications, who is allowed to accept modification requests, which forms must be used, and other protocols. Both suppliers and customers are involved in this activity with the final approval being given by the regulatory authorities.

Modification Management

The modification plan describes all the documents/tests/analyses which must be updated to demonstrate that the requirements are still fulfilled.

The most significant items are:

a. Qualification Test Program/Report

Every modification must be analyzed to determine its effect on the validity of the qualification tests. From a safety point of view it must be understood that a series of small modifications may lead to a product which is significantly different from the original; in such cases requalification will be required.

b. FMEA or System Safety Analysis (SSA)

For any modification to a flight critical system a systematic review of the FMEA or SSA is necessary to identify the repercussions of this modification on the safety criteria. Any impact on the FMEA or SSA has to be addressed in the change notice to demonstrate clearly that safety aspects are not degraded.

c. Acceptance Test Procedure (ATP)

As with the FMEA, the impact on the ATP must be considered when any modification is proposed.

d. Declaration of Design and Performance (DDP)

For European civil aircraft, the DDP is the document presented to the regulators which declares all the relevant evidence submitted to support certification of an item or system (e.g., specifications, configuration index, accomplishment summary, qualification test report, acceptance test procedure.) It also lists the restrictions on usage. The DDP has to be reissued for any modification.

e. Accomplishment Summary

For software in civil aircraft, the current practice is to produce an "accomplishment summary." (See DO 178A) This document describes, briefly, the major milestones of the project, the development environment and the development process. When modifications have been made, the Accomplishment Summary summarizes the modification and the problem reports resolved by the new release. It describes any remaining problems and the major features of the new release.

f. Change Notice

To be approved, any modification must be identified by a change notice. This notice will define when the modification will be introduced into production andwhether it is mandatory or can be introduced on repair or at retrofit. To provide the information to enable the modification to be assessed, documents are included to show that tests have been performed which demonstrate that the modified system is fully acceptable. This demonstration may be made by the evidence in a test report which gives the procedures used and the test results. Most of the time, the test procedures to be applied are a subset of the qualification program. For some special aspects such as electromagnetic compatibility or lightning, specific agreements may be necessary to define the rules to be followed to validate correct operation of the system after any modification, even a minor one.

g. Service Bulletins (SB) or Service Information Letter (SIL)

These documents are used in civil aviation to inform the operators of the modifications. Retrospective actions are described. When a major change is involved, in which several types of equipment have to be modified at the same time, the system authority will issue an SB to introduce the individual SB's and to describe the allowable configurations.

Advisory Circular/Equivalent Documents

The U.S. civil aviation regulations define Federal Aviation Regulations (FAR) to control the approval of aircraft and their systems. European regulations have similar rules called Joint Airworthiness Requirements (JAR's). Table 4.4 lists some examples of these regulations which are applicable to FCS's.

In addition, the civil aviation industry has produced documents to define various development processes. Examples of these documents are given in Table 4.5 The manufacturers must demonstrate compliance with these standards.

Acceptance Test Procedures

These procedures define how an equipment or subsystem shall be finally checked prior to acceptance. They must be approved by the customer and by the regulators (e.g., LBA in West Germany, VERITAS in France for civil aviation, SIAR in France for military equipment).

Some customers require that the procedures are in "clear language" while others require the procedures to be in ATLAS (ATLAS is a test language generally requested in civil aviation).
The tests for the acceptance of the hardware and the software and all the data needed for adjustments must be provided.

Environmental Testing

Environmental testing is an essential part of the validation and reliability testing of flight critical systems (see Table 4.6).

All-weather testing continues the validation and reliability testing of flight critical systems. Such systems have already been tested in the laboratory to extremes of heat, cold, humidity, vibration, and in extensive ground and flight test by the time climatic testing is performed. The nature of flight critical systems is such that any impact on them by any source, including climate and weather, is a safety issue as well as an operational issue. The system must operate over the entire operational envelope of the air vehicle, and it

must degrade in an acceptable manner when forced outside of that envelope.

The major objective of all-weather testing is to determine to what extent a weapon system, including its essential support equipment and attendant crews, can accomplish the design mission in the required climatic extremes. Historically, all-weather testing has revealed design deficiencies that impact the operational capability of the air vehicle and require system modification to meet operational needs.

Four extreme environments provide the boundaries of air vehicle operation--arctic, desert, tropic, and adverse. The arctic environment represents cold (-29 degrees C/-20 degrees F and below) and wind. In the desert environment, heat and sunlight (43 degrees C/110 degrees F and above) are the primary factors along with blowing sand, dust, and dryness. The tropic environment is dominated by humidity as well as precipitation and fungus (27 degrees C/86 degrees F with relative humidity of 75 percent and above). Adverse environments include weather (snow, rain, sleet, hail, slush, icing, turbulence, and IFR conditions) and corrosion (salt spray and pollutants).

Three stages of testing are involved in all-weather qualification.

a. The first stage is in a controlled environment such as a laboratory or manufacturing test facility. Here conditions can be controlled individually and can be taken to extremes for failure and reliability studies. Much of this testing is at the component and subsystem level, but at facilities like the McKinley Climatic Laboratory at Eglin AFB, Florida, USA, entire aircraft can be tested.

b. The second phase of testing environmental extremes involves deploying test aircraft, support equipment, and test teams to locations that have the desired climatic conditions. This kind of test is usually part of the certification process for civil aircraft (e.g., certifying a head-up display in Europe requires 30 landings with blind cockpit among which a defined number have to be performed in rain conditions, or with severe lateral wind in order to demonstrate that the trajectory deviations when touching down are within +/- six meters). The useful season may be limited and weather patterns may alter schedules, but when the prescribed weather is available, ground and flight operations are unrestricted beyond the requirements for safety, data processing, and limitations found in the system being tested.

c. The third stage of all-weather testinginvolves actual operational use. Since malfunctions are more probable during extreme conditions when systems are being operated closer to their environmental design limits, redundancy management and backup systems are more likely to be called into use. Therefore, reversion modes and backup systems must perform to the limits of normal operation and must be tested as thoroughly as primary systems at extreme conditions.

The present trend is necessarily toward more reliable and more self-contained aircraft to reduce the support equipment and personnel required. For flight critical systems, this means more electronics, more redundancy, more backup systems, more interfacing to mechanical, hydraulic, electrical, and pneumatic subsystems, as well as more communication with other systems.

All-weather testing is an important part of ensuring that the resulting air vehicle can perform its intended mission--anywhere, anytime.

4.14 Validation of the Fielded System

Even when a system has entered operational service there is a need to validate that the system is meeting the customers requirements and where necessary to make modification.

Figure 4.16 shows one way in which the validation of the fielded system can be organized. Information from operational use may take the form of defect reports, flight incident reports and reports on the suitability of the system for its operational role. Where these reports, after analysis, indicate shortcomings against the specification of a revised operational requirement then modification will be requested and developed.

To retain the integrity of the system it is necessary to ensure that the modifications do not invalidate the testing or analyses on which the system is certified. Thus it will be necessary to consider the impact of

all phases of development and validation and to repeat where necessary.

Rigorous configuration control will be needed when introducing the modification into service, particularly where the interfaces between functions and/or equipments are changed.

4.15 Special Topics

4.15.1 Traceability

As the subsystem specifications are developed from the system specification it is important to maintain traceability. Software tools are available to help this process. They use databases to store the multiple levels of specification and keywords to relate the lower level requirements or design statements back to the higher level requirements. One such tool is EPOS (12), developed by GPP in Germany. This form of traceability is most useful for software. It becomes the basis of a top down hierarchical structure for the software development process. In the development of hardware, the implementation is less directly linked to functional requirements because equipment is shared between functions. Hence traceability is ensured by compliance matrixes rather than hierarchical decomposition techniques.

4.15.2 Use of Formal/Semi Formal Languages

There are many areas where precise design and development languages can improve the clarity of specifications, e.g. the use of Fortran to define control laws, the use of Ada to describe procedures, the use of Boolean equations to define mode switching logic, the use of structural analysis methods such as Yourdon and Hood to define system data flows,
the definition of a particularly critical piece of hardware in ELLA(5) or VHDL(6) or the definition of software in Z(7) or VDM(8) or the definition of the whole system in structured English. The use of these formal language techniques is increasing, and computer aids are available in increasing numbers with increasing capability. They aid the designer to maintain traceability, classify system models, and define system interfaces by handling large quantities of data efficiently. They ease the task of validation by producing a rigorous definition of requirements against which to compare implementation.

4.15.3 Project/National Specifications

On a project there will be many requirements which are common and can be defined by referencing a project or national specification. Clearly the use of such specifications plays an important role in improving the quality of systems, but it is important to check that they are appropriate to safety critical systems. Many specifications were formulated prior to the development of fly-by-wire systems while others were developed for less critical applications. Where common specifications affect system integrity they should be analyzed against system requirements to validate that they are appropriate .

4.15.4 Varying Criticality

In a typical safety critical system there will be signals which have a criticality level lower than others. If such signals fail they can be removed from the control system and operation continues at a less optimum level. It is often possible to reduce the level of complexity by separating out the functions which are non-essential from those which are essential. The level of redundancy of the signal conditioning and processing for the total system may be reduced, with an attendant reduction in the degree of systems analysis and testing. However, since less critical parts of the system can take over the most critical ones and corrupt the whole system, some design precautions (mix of hardware and software protections) are needed for ensuring the integrity of the partition. As for synchronization, the choice of partitioning the system into varying criticalities is really an engineering trade-off and should be carefully examined.

4.15.5 Assessment of the Use of Test as the Principal Hardware Validation Method

Testing remains the principal mode of validation for flight critical flight control system hardware. There are some notable exceptions, eg, automated sneak circuit analysis methods which analyze hardware drawings and wiring lists. It is important to understand the nature of hardware testing in the overall validation process.

Overall Assessment

Hardware tests are a powerful method of checking that the flight computers operate according to the design specification. The digital nature of the hardware permits millions of tests to be performed quickly in a representative environment. It is relatively straightforward to check the electrical signal interfaces between various parts of the system through test. In addition, it is extremely effective to let the flight computer hardware check itself through test. On the other hand, design assurance tests are normally carried out on only a sample basis. They have to be supplemented by tolerance analyses. It is also difficult or impossible to simulate some of the internal failures of the computer.

Performance Validation

The exact design specification performance of the subsystem elements is often critical to safe operation of the overall flight control system, and must be guaranteed. Hence the equipment must be carefully and thoroughly checked for performance throughout the operating environment, and this test series becomes part of the ultimate flight clearance. Similarly, precise performance must be guaranteed over the range of component tolerances and analyses/tests must be of high integrity and precision.

Bite Coverage

Where system design relies on the detection of failures by BITE, that system must be checked for correct operation and for coverage. Special analyses and tests must be done to check that the BITE will detect, isolate and reconfigure the system as specified.

Component Validation

Many of the components used in flight control systems are complex and system operation may be dependent on their precisely correct operation. One approach to this problem has been to declare that if a system depends on the design of a component being error free for its safety, then that component must be 100% analyzed or tested. Alternatively, dissimilar components should be used to prevent system failure due to a common design error. This latter approach has also been followed for flight critical software in very high integrity systems such as civil aircraft fly-by-wire systems. Where system safety depends on component designs being error free, a formal verification of the design may be necessary. This may consist of comprehensive testing, or of specifications (formal or semi formal) of component performance and comparison of implementation against that specification or a combination of the two approaches.

Test Software

It is usual to develop and test the hardware using special software. This is done because it allows the engineers to probe the hardware more deeply, to check out critical timings, to test tolerances and to test the BITE coverage comprehensively and at a detailed level. All these aspects are difficult, if not impossible to test with operational programs. It is also useful to separate hardware and software development to reduce schedule constraints. The development of the hardware may precede the development of the software, particularly if existing hardware is being used and it is very beneficial to separate the critical paths. The test software will enable the engineer to validate the hardware performance, speed, thermal characteristics, environment, accuracy and resolution, etc .

4.15.6 Allowable Constructs in Software

Experience has shown that some instructions/constructs are error prone and should be banned; the software specification should define the software instructions/constructs which will be allowed. The

software support facilities should then contain the capability to detect violations of construct rules automatically.

4.15.7 Program Design Languages

Major sections of the software subsystem specification may be in a language format to improve rigor and facilitate traceability. This may be accomplished by using programming languages or specialized program design languages (PDL).

4.15.8 Assessment of Software Validation Methods

The volume of flight critical software is increasing, and it is important to assess the status of the state of the art methods of validation, because of the the significant cost, calendar time, and criticality of the software validation process.

Validation by Test

The positive aspects of validation by test are:

- A wide range of conditions can be checked
- The structure of the modules/programs and operation with critical data can be checked
- The task can be split between software engineers and modules/subprograms can be checked in parallel by independent testers
- Confidence can be gained by demonstrating long periods of fault free operation

The negative aspects of validation by test are

- A large number of test cases have to be generated
- Testing is very manpower intensive
- It is virtually impossible to prove that "sufficient" checks have been made.
- The testing is only as thorough as the engineers who devise the testing rules and the tests

Validation by Construction

The positive aspects of validation by construction are:

- Formal methods have the potential to prove that software is correct against a formal specification
- The structure used for construction can also be used as a framework for validation
- Software tools are available to support many of the techniques, thus reducing theconsiderable potential of human error and the drudgery involved in other methods.

The negative aspects of validation by construction are:

- Formal proof approaches are generally quite abstract and difficult to apply, even to moderately sized real time programs

- The methods are not related to the normal skills of practicing flight control system designers

- Proving the integrity of the proving algorithms is difficult

4.15.9 EMI Tests

EMI tests one of the major concerns for flight critical systems, their im his class of threat. One important problem is linked to the way these tests are defined in applicable documents. Most of the time, these documents comply with D0 160-B (dash-C is to be issued soon) or MIL-STD 461C. These standards are generally concerned with individual equipments, not with systems or even subsystems. The test descriptions provided in these standards are idealized conditions, especially for all grounding specifications. Specific tests have to be conducted on real aircraft to demonstrate compliance of the systems with the requirements.

REFERENCES

[1] "Methods for Aircraft State and Parameter Identification", AGARD-CP-172, May, 1975

[2] Cooper, G.E., and Harper, Robert P., Jr., "The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities", NASA TN-D-5153, 1969

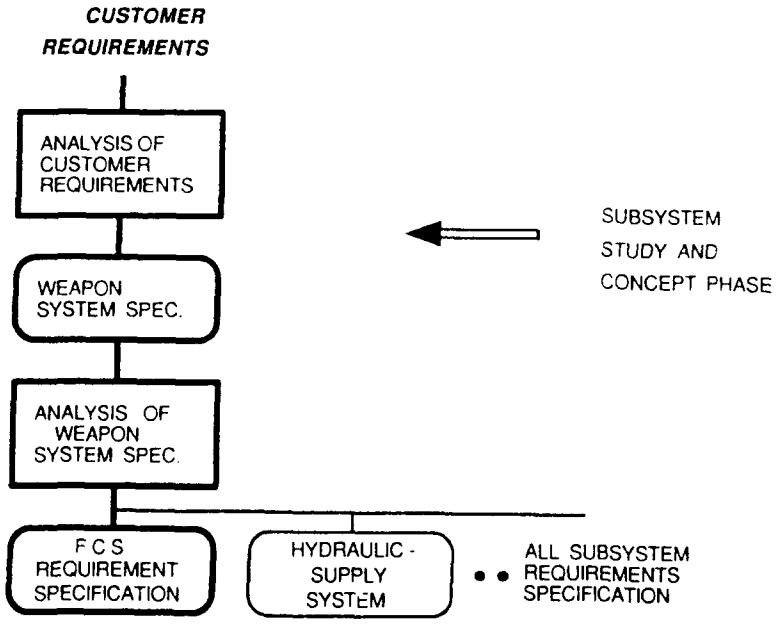


Figure 4.1 State-of-the-Art Validation Methods

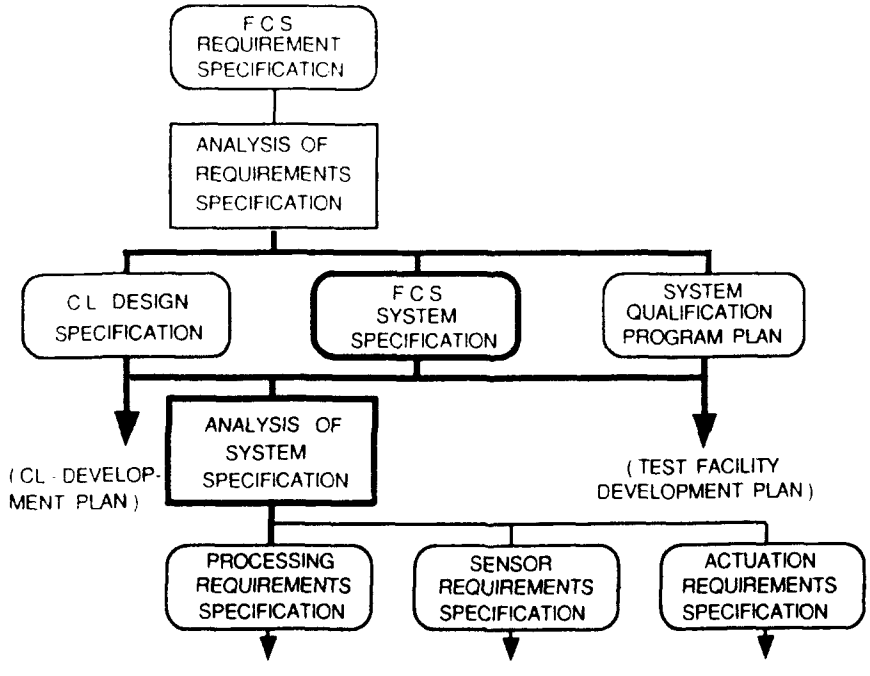


Figure 4.2 State-of-the-Art Validation Methods (continued)

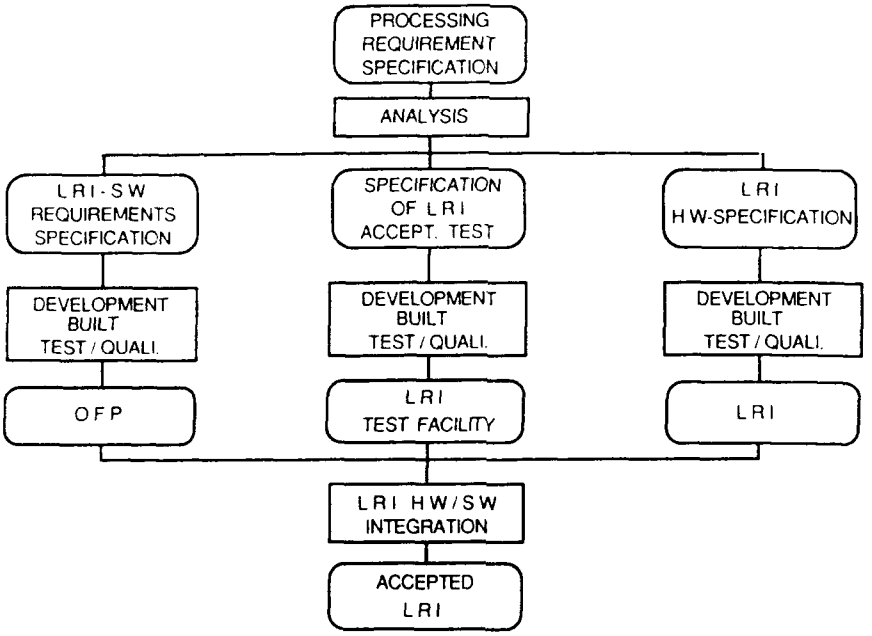


Figure 4.3 State-of-the-Art Validation Methods (continued)

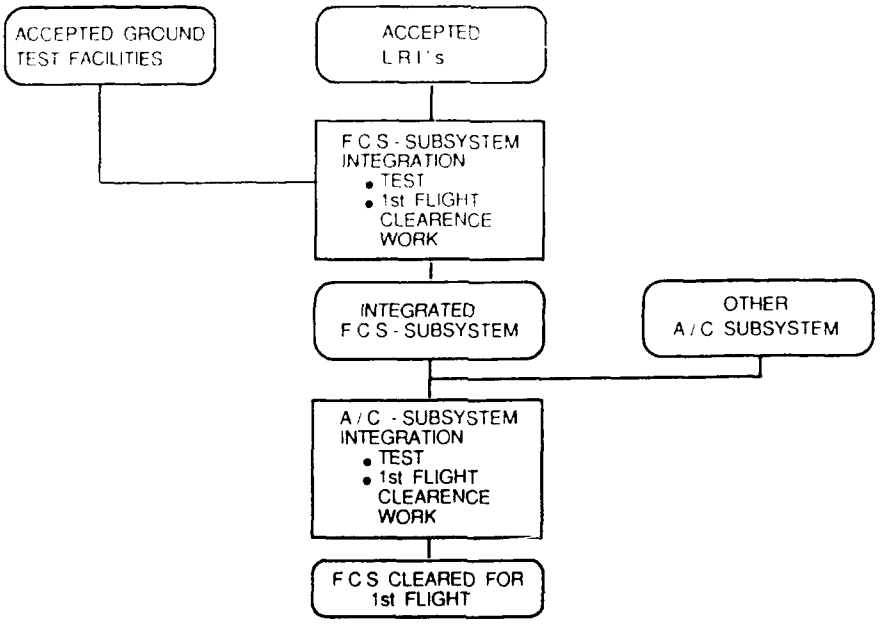


Figure 4.4 State-of-the-Art Validation Methods (continued)

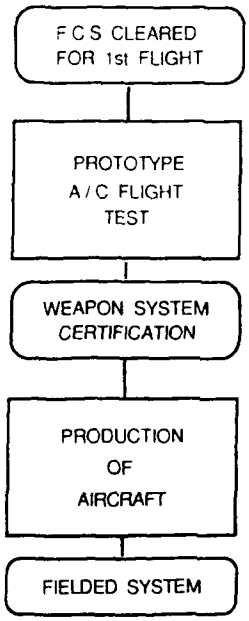


Figure 4.5 State-of-the-Art Validation Methods (completed)

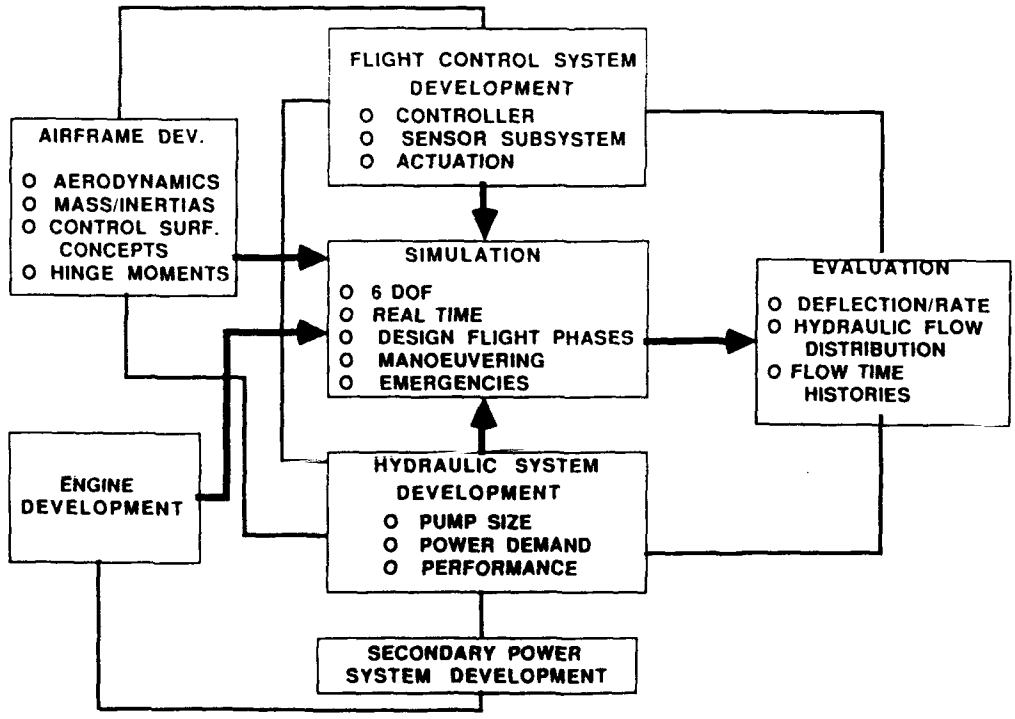


Figure 4.6 Flight Control System Development

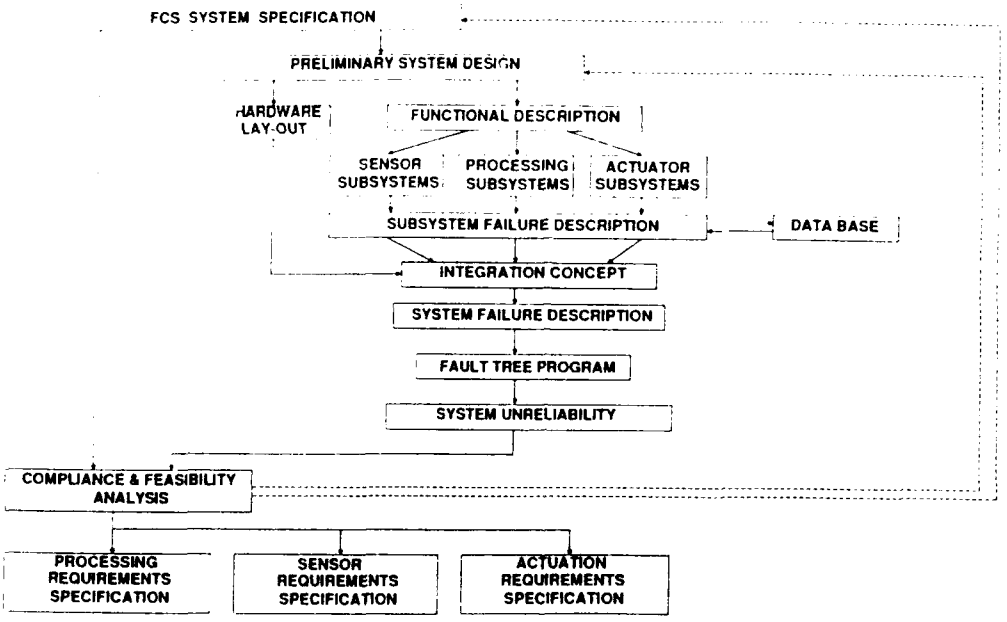


Figure 4.7 Development of Requirements Specifications for Processing, Sensors, and Actuators

1. Introduction
2. Applicable Specifications
3. Subsystem Description
4. Performance Requirements
   - 4.1 Functional Requirements
     - Control
     - Monitoring
     - Test/BITE
     - Indications/Displays
     - Power
   - 4.2 Reliability and Safety
   - 4.3 Maintainability
5. Interfaces
   - Inputs
   - Outputs
6. Weight
7. Volume
8. Validation
9. Qualification
   - Environment
   - Endurance
10. Design, Materials, Processes

Table 4.1 - Typical Subsystem Hardware Specification Document

1. Introduction
2. Applicable Documents
3. System Description
4. Software Structure
   - Functional Structure
   - Data Specifications & Data Dictionary
5. Software Engineering Methods
   - Design Methods
   - Standards
   - Language
   - Documentation
   - Configuration Control
6. System Functional Requirements
   - 6.1 Normal Operation
     - modes of control
     - performance
     - startup
     - BITE
   - 6.2 Failure Mode Operation
     - redundancy management
     - failure reporting
     - performance
   - 6.3 Components Requirements
     - inputs
     - processing
     - outputs
     - controls
     - performance
     - redundancy management
     - failure reporting
7. Test Methods
8. Software Language (including standards)
9. BITE

Table 4.2 - Typical Software Subsystem Specification Document

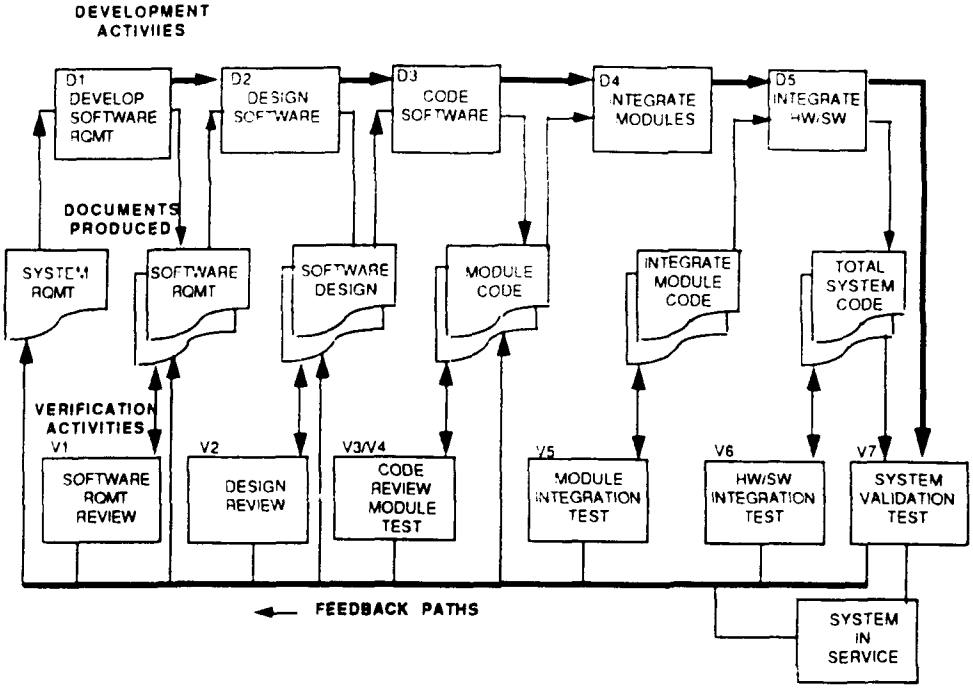


Figure 4.8 Software Development

1. Performance
   - 1.1 Noise rejection
   - 1.2 Control laws
   - 1.3 Inter-unit Compatability
   - 1.4 Bus Compatability
   - 1.5 Ability to Drive Outputs
   - 1.6 Redundancy Management
     - voter/monitor operation
     - fault detection
   - 1.7 Speed
     - iteration rates
       - foreground
       - background
     - response times
2. Tolerance to data sets
3. Tolerance to Power Supplies Variation
4. Bite Effectiveness
5. Memory usage

Table 4.3 - Aspects Verified by Test of Integrated Hardware and Software

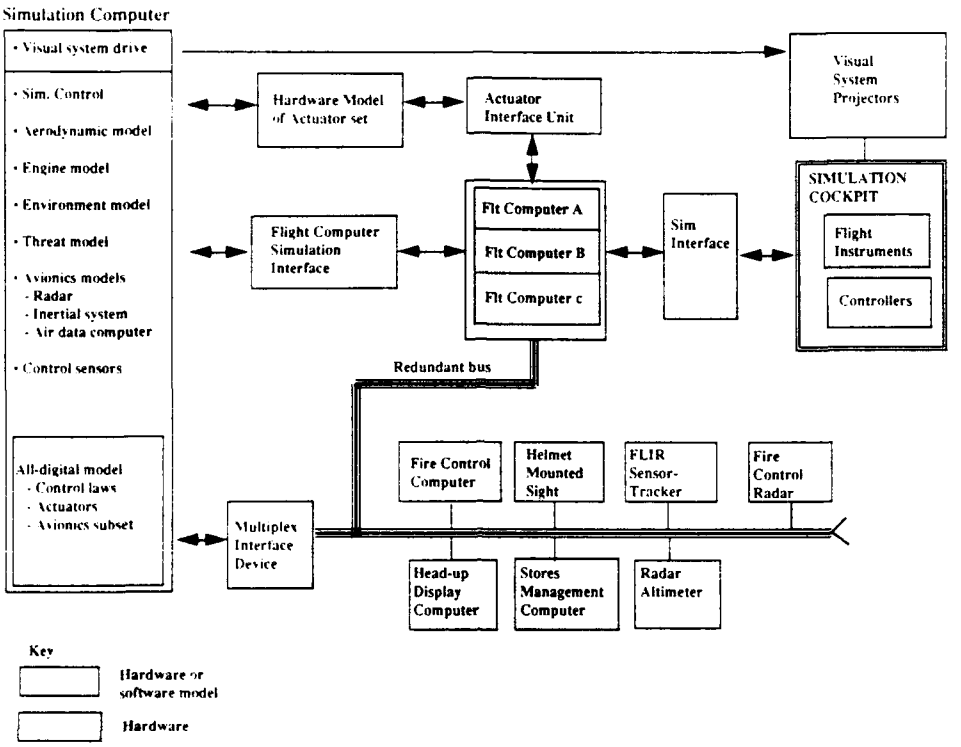


Figure 4.9 Typical Hot Bench Configuration

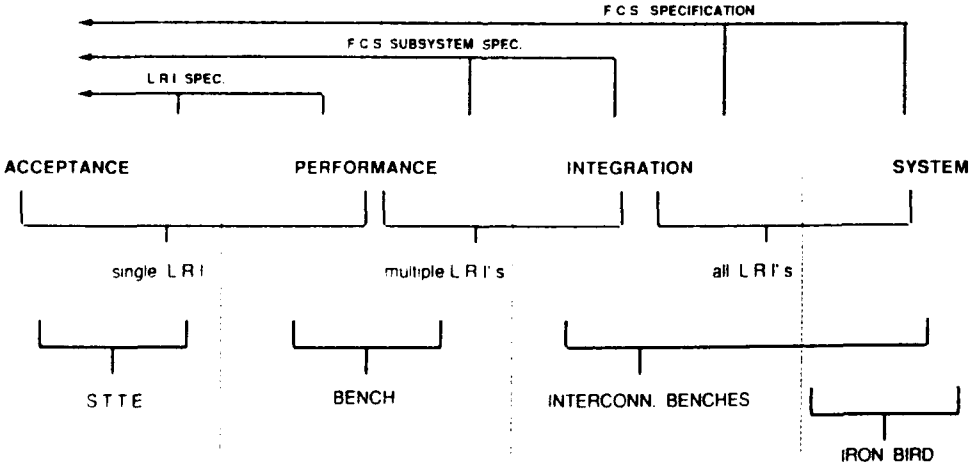


Figure 4.10 Integration Phases

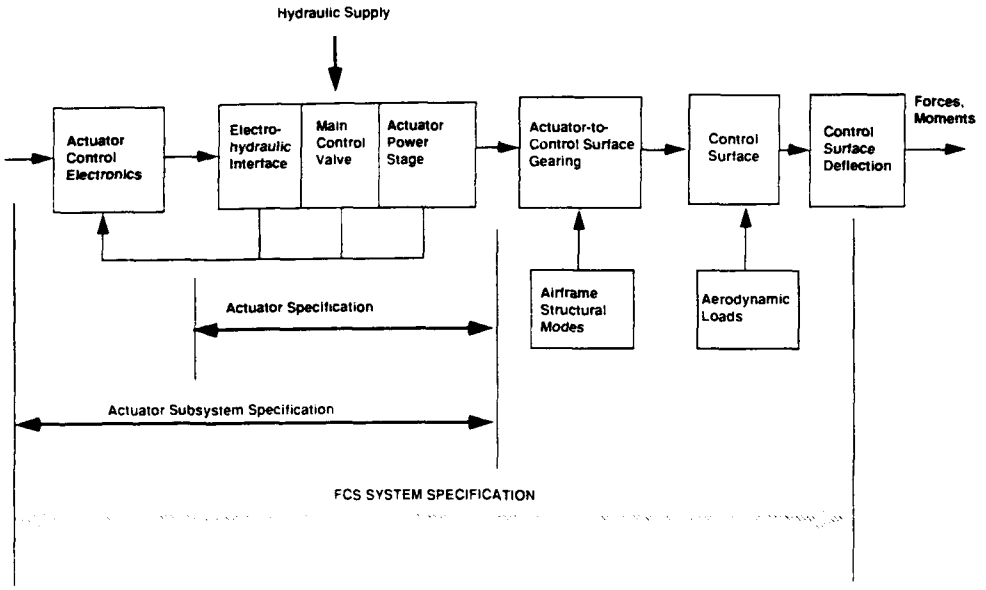


Figure 4.11 Actuation Subsystem Validation Boundaries

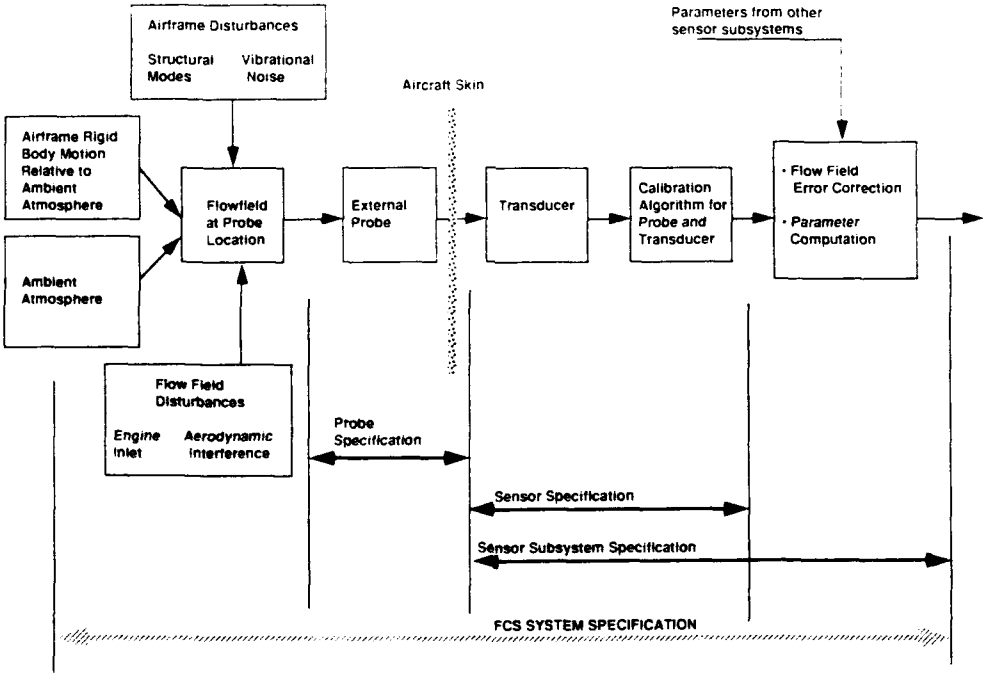


Figure 4.12 Air Data Probe Validation Boundaries

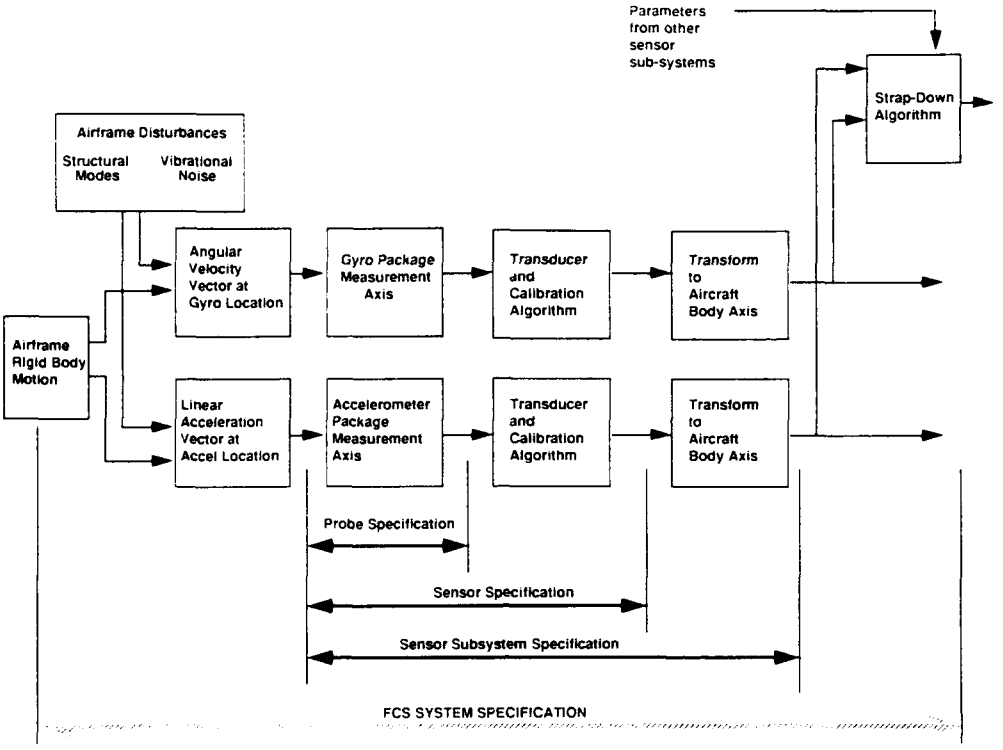


Figure 4.13 Inertial Sensor Subsystem Validation Boundaries

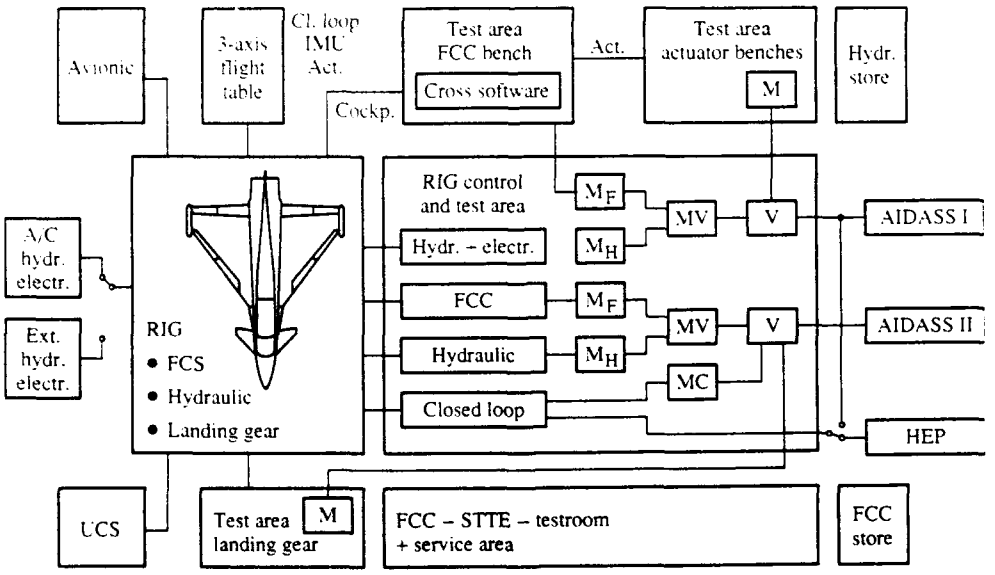


Figure 4.14 Typical Iron Bird Configuration

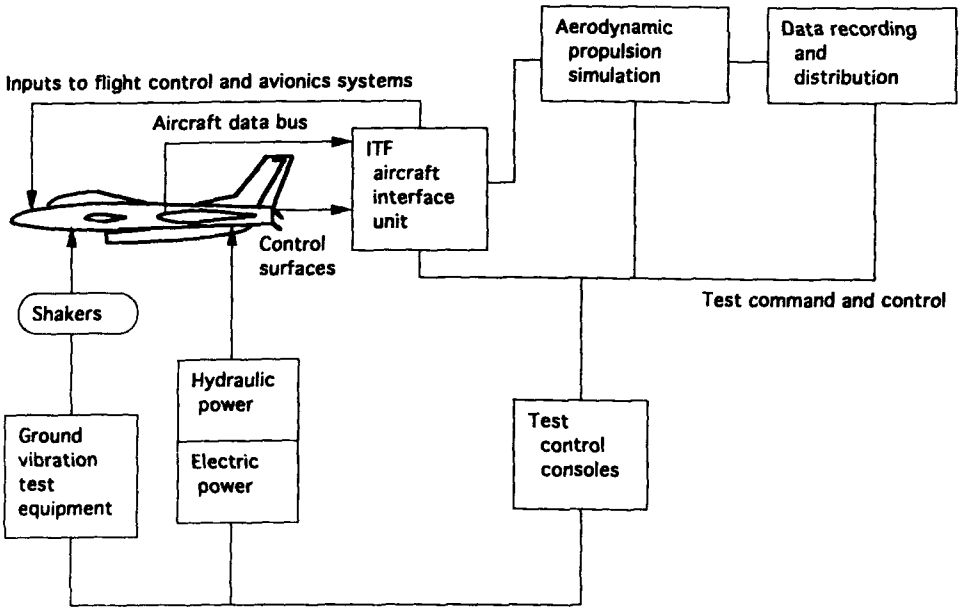


Figure 4.15 Integrated Test Facility

a) FAR 25.303	Factors of Safety
b) FAR 25.561	Emergency Landing Conditions - General
c) FAR 25.1322	Warning, Caution and Advisory Lights
d) FAR 25.1355	Flight Director Systems
e) FAR 135.335	Approval of Aircraft Simulators and Other Devices
e) FAR 135.351	Recurrent Training
Advisory Circulars, e.g.	
a) AC 20-115A	RTCA/DO-178A
b) AC 20.131	Approval of Traffic Alert and Collision Avoidance Systems
Technical Standing Orders, e.g.	
a) TSO C119	Traffic Alert and Collision Avoidance System Airborne Equipment TCAS II

Table 4.4 - Examples of Regulatory Rules Applicable to Flight Control Systems

a) Radio Technical Commission for Aeronautics Document DO-160B - Environmental Conditions and Test Procedures for Airborne Equipment (RTCA documents can be obtained fromRTCA, Washington D.C., U.S.A.)

b) Aerospace Recommended Practice (ARP) 962A, - Fault/Failure Analysis Procedures

c) ARP 1834 - Fault/Failure Analysis Guidelines for Digital Equipment. ARP documents can be obtained from SAE (Society of Automotive Engineers, Inc., Warrendale, PA., U.S.A.

Table 4.5 - Examples of Aviation Industry Standards Applicable to Flight Control Systems

Temperature Cycling
Vibration
Acoustic Noise
Humidity
Low Pressure
High Temperature
Low Temperature
Temperature Shock
Solar Radiation
Rain
Fungus
Salt Fog
Sand and Dust
Icing/Freezing Rain
Vibro-Acoustic/Temperature
Electro Magnetic Capability
- Electro Magnetic Interference
- Lightning Protection
- Electrical Grounds
- Electrical Bonding

Chemical - Biological
Nuclear
Environmental Control System Inputs
Internal Environment
Electrical Power
Logistics Environment

Table 4.6 - Environmental Conditions (from Appendix of Mil Std B72444)

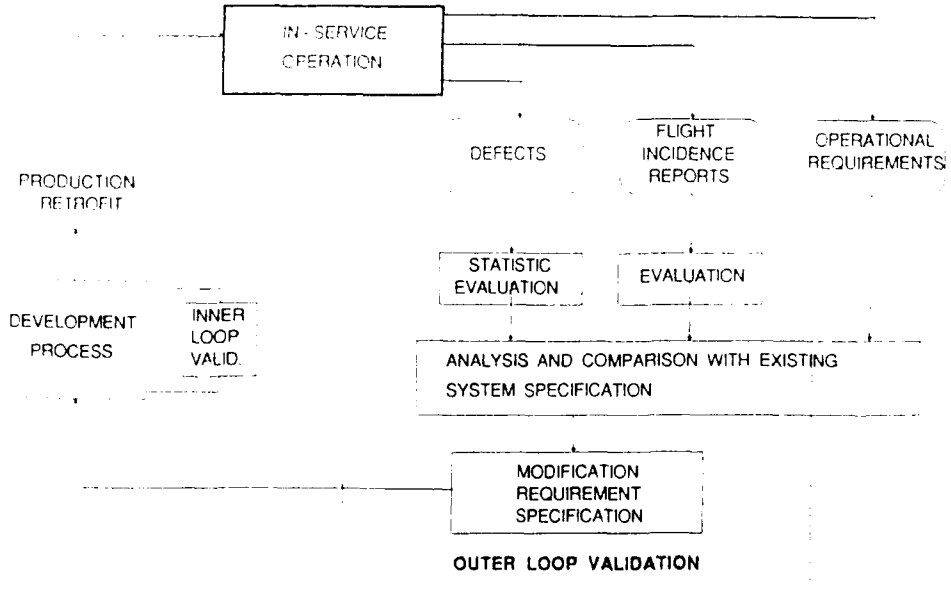


Figure 4.16 Fielded Service Validation

CHAPTER 5

ASSESSMENT OF THE STATE OF THE ART VALIDATION PROCESS

5.1 Introduction

This chapter contains an assessment of the validation issues associated with FCCS development, and follows the organization of Chapter 4, which described the state of the art of validation activities, methods and tools.

5.2 Assessment Criteria

Validation is regarded by the Working Group as an integral part of the system development process. Since the principal focus of the WG was the safety aspect of validation, this assessment having the most direct impact on two major aircraft development milestones--first flight clearance, and demonstration of safe operation throughout the full design envelope of the aircraft.

First flight clearance must be considered as the most critical point in the development process of any new flight control system. The majority of methods and tools needed for validation, with the exception of flight test instrumentation, are driven by first flight clearance requirements. Therefore, any critical assessment of the validation process must focus on this phase.

The assessment made in this chapter is limited to the case of a completely new system, where an aircraft, has to be cleared for first flight. The majority of activities needed to clear a flight critical system after a redesign or after a major modification can be regarded as a subset of first flight clearance work.

The assessments made in this chapter on made on the basis of three criteria,

- Technical
- Economics
- Schedule

These three criteria are obviously important for any system development. For FCCS, the safety-criticality feature imposes constraints on the ability to trade off easily between these three criteria. More freedom exists in non-flight-critical system development

5.2.1 Technical Criteria

The dictum and policy of any flight program, "safety first", inherently limits the flexibility to freely trade off defined steps of the validation process in order to solve resource or schedule problems elsewhere in the program. For every function which has been declared essential (MIL-Spec 9490), the safe in-flight performance must be demonstrated. The steps to doing this are usually established years before the actual ground and flight test program. This planning process also defines the amount and kind of test equipment and tools purchased and developed for the validation program.

If during the validation process leading to first flight clearance activities, a safety-of-flight problem occurs in the area of essential functions, the problem has to be cured. Concessions (waivers) are only possible in the area of non-essential functions.

During the development of new flight systems, it is common practice to carry out the first flight with a reduced functional subset of the full system (restricted envelope). This approach minimizes first flight clearance work but also eliminates most of the flexibility in the validation process because the remaining system functions are critical to safe flight.

5.2.2 Economic Criteria

Funding is allocated to the planned validation activities at an early state of a project. As a consequence, test facilities are constructed, engineers are assigned and trained, and specialized tools are developed during the time when the flight systems are being developed.

Quite often, a gap develops between the original validation plan on which the funding was based and the validation activities actually required for the FCCS. If that gap is not recognized until late in the program, it is difficult to close by an increase in funding, because of the long lead time required to provide the tools, facilities, and people needed to accomplish the task. The resources required for the validation of safety critical systems are often underestimated, hence problems frequently develop late in the program, during the peak of validation activities, when there are limited options available to correct them.

Mature methods and tools, together with experienced engineers are the major components of a successful validation program. The achievement of this state of readiness requires an early technical definition of the tools, ground test hardware, interfaces, and software support environment. Ironically, this results in a lack of flexibility to respond to unforeseen events during the critical phase prior to first flight, if different or expanded validation steps must be used.

There are indirect costs related to providing a staff of experienced engineers to carry out the validation of safety critical systems in addition to the direct labor cost. This comes about because it is costly to maintain a pool of such engineers in between programs. The experience level required by such engineers usually precludes merely hiring them just prior to the final validation phase.

5.2.3 Schedule Criteria

First flight clearance activities culminate during the period of time approximately 18 months before first flight. This is the phase of maximum technical uncertainty and numerous dependencies between equipment, deliveries, and integration activities. The overall top level project planning is usually carried out years before this phase. Historically, there is always significant pressure to minimize the duration of the final validation period because of funding and milestone commitments. Although the first-flight system is usually a subset of the final configuration, the remaining elements and functions have a high criticality level, and there are usually few areas where concessions or waivers are possible.

Meeting the major program milestones of such system development plans is a challenge common to all development work. Not meeting a major milestone where validation of a flight critical system is involved usually has major consequences:

- It may force a reevaluation and rescoping of validation activities during the peak period of validation, when it is not easy to determine all the potential impacts on total validation coverage

- It may create an imbalance between the demands for quality and thoroughness, and completion date.

5.2.4 Importance of these Criteria for the Future

In the past, there has always been much attention on the design and development of FBW flight control systems. Safe operation of such systems was generally achieved and demonstrated through the the expertise, experience, and ingenuity of small groups of flight control engineers. These same technical, economic, and schedule problems existed in the early days of FBW, but they were outweighed by the benefits provided by this emerging new technology.

It is for the current and future generations of FBW systems, that these problems are becoming dominant. FBW technology has become the standard approach for flight control systems for every new military aircraft, thus these factors and criteria are becoming discriminators between successful programs that meet cost and schedule targets, and those that do not.

Therefore, a principal factor in the success of the validation process is how it is accomplished within the normal program constraints of funding and schedule. The technical objectives of the validation process must, of course, also be of the highest quality for FCCS.

5.3 System Development Plan

The members of WG 09 concluded that there is no general or "generic" validation plan which can be directly applied to any system containing mechanical components such as sensors and actuators, computing elements with specialized software such as sensor preprocessing, and application software such as control laws. WG 09 initially attempted to directly adapt the development process for software systems as outlined in MIL-STD 2167 as a means to develop a structure for the discussion of methods and tools.

However, due to the interdisciplinary nature of a modern flight control system, this approach failed to address the complex interrelationship between four interdependent development processes involved in the development of a FCCS:

- airframe development (aerodynamic data set, mass distribution, aeroelastic behavior, structural modes, etc.)

- control law development (providing positive stability and acceptable handling qualities)

- system development (providing the functional elements for control law implementation and failure tolerancecapability including software development)

- test facility development

These processes are carried out in parallel, and rely on a very early definition of design requirements and/or parameters crossfed from each of the other processes. This burdens the validation activities in each process with important dependencies. For example, validation tests of the overall flight control system can be achieved only within the validity of the control laws, which are only valid within the confidence level of the aerodynamic data set. All these dependencies have to be covered before first flight, which represents a threat to accomplishment within planned resources and schedule.

This challenge is amplified by the fact that airframe development and control law development traditionally follow a different plan than that for flight system hardware and software and test facility development. Nevertheless, it must be recognized that the important topic of validating a safety critical system cannot be separated from discussing the means of developing a safety critical system.

Since testing is the dominant element in the validation process, the peak of activities occurs between start of equipment manufacturing and first flight. The test activities build up in a sequence which is driven by the availability of the flight system hardware and software. This sequence is usually : Module Test, Equipment Test, Subsystem Integration Test, and System Integration Test. This sequence highlights some inherent problems in a practical validation process:

- Timing Problem: System design/specification activities and the associated validation/test activities are separated in a typical development plan by a minimum of 12 months (equipment specification to equipment qualification test) and up to three years (system specification to system integration test). The most critical system integration tests begin about 12 months before first flight. There is a high schedule risk because of the dependence of the somewhat independent developments.

- Density Problem: Normally, the planning for the last 18 months before first flight contains a high density of test activities. Based on the uncertain nature of this phase, these plans have a high degree of uncertainty. The potential for unforeseen events occurring during this phase is also high, and therefore this time period is vulnerable to being used to absorb delays in systems development.

The System Development Program Plan should be constructed with this probable schedule outcome in

mind, and an attempt should be made to redistribute the validation activities. The goal should be to place a stronger emphasis on validation activities in early stages of the project.

The following ways to achieve this goal were identified by the WG:

- Start with the final specification from a previous program in order to ensure completeness of the design specification with regard to subsystem/system validation, traceability and dependencies.

- Utilize a program design language to provide automated bookkeeping and identification of dependencies. Modern relational data bases can provide powerful tools for tracking complex systems.

- Ensure that an adequate validation process exists for innovative or unconventional designs as well as for conventional designs. For unconventional or new designs approaches, a sensitivity analysis should be conducted during the early concept phase. This analysis should identify the areas requiring special `emphasis in the validation process particularly with respect to specifications where experience is lacking. In most such cases, early prototyping has to be carried out. This can be done in the laboratory environment, or it may even require investigation in a flight research program, where the laboratory environment cannot faithfully reproduce the flight conditions.

Moving validation activities from the critical phase before first flight to the specification and design phases of the development cycle can only be accomplished by a redistribution of the funding. It is an accepted engineering concept to carry out early prototyping for hardware and software. It is equally important to use the early phases of the program for validation activities which can reduce the costs and schedule conflicts later in the program.

5.4 Management of Validation Activities

The safety critical factor for FCCS's, although of paramount importance, should not unduly overshadow other issues. Validation is an activity which has to be performed for any system development, and the common objective is to create evidence that a function performs as it should. A significant amount of money has been invested in methods and tools for the development of mission avionics for military aircraft, especially for software development. The reason for this is that the vast majority of flight software is for aircraft avionics systems. FCCS software is usually a small fraction of the total amount of software written for a modern military aircraft.

Modern flight control systems also exhibit a high level of functional integration with other aircraft systems. As a consequence, an increasing amount of safety risk is associated with the interface between the FCCS and other aircraft systems. The use of standardized methods and tools across all aircraft systems would improve the capability of controlling these increasingly critical interfaces.

The development of safety critical systems should make use of general methods and tools for the development of complex systems to the greatest degree possible. It is a worthwhile activity to search for such general tools as a first step in the development of the validation process, before investing in the production of unique tools. Only when the need has been proven by a thorough analysis should these general tools be modified, or replaced by specialized tools/methods for safety critical system development and validation. The benefits of this approach are obvious:

- Shared development costs for the tools

- Tool validation through a wide application

This approach does not ignore the specific differences in system developments associated with the level of safety criticality. This difference must be addressed by the management of the validation activities. The application of general tools and methods must be governed by a detailed knowledge of their capabilities and limitations, and by enforcing rules and constraints necessary for their proper application and use..

First generation FBW flight control systems were designed to perform the basic flight control task of improving stability and handling characteristics and in of providing basic autopilot modes. This basic

task required an interdisciplinary approach (aerodynamics control laws, system design software). The software problem was contained in a well defined system (functional boundaries were identical with equipment boundaries) with "simple" and tightly controllable interfaces. The development work was carried out by a small, expert, and experienced flight control team. Each responsible engineer within such a team was able to fully understand and monitor the interdisciplinary development activities, even though being directly responsible only for one area, personally.

For example, the engineer who performed the software design could understand the mechanisms of control laws, having an engineering background in control system engineering. Design and validation was carried out by one small group. In the early developments of FBW technology, creating this small group was the most critical task of management in ensuring that a safe system would result.

For modern systems, however, the situation is considerably different.

- Increased levels of integration of modern systems has increased the complexity of the interfaces. Not only are the functional interfaces between equipments and subsystems affected, but so are the interfaces between different engineering groups.

- The function of the system data processing has increased far beyond the basic task of improving stability and handling characteristics. In the same way, the amount of software contained in the system has increased enormously.

- System boundaries are no longer unique. Equipment assigned to the flight control system is no longer identical with the functions assigned to the flight control system.

Strong management is required to handle the validation activities for this increased interface complexity within budgetary (resources) and time (planning) limits, without major technical concessions. This management process must address:

- Carefully thought-out organizational structures, including well defined responsibilities for design, program control, configuration management, documentation, technical reporting, scheduling, and decision-making.

- The rigorous use of tools and methods (common for the whole project as outlined above) to control the interfaces between the various activities

- The assignment of experienced engineers to critical tasks

The development of such complex systems as described previously requires an increasing amount of engineering labor. If this had to be provided solely by flight control specialists, it would be difficult to maintain such a large team between programs. One solution is to minimize the number of flight control specialists for design and validation, and to rely on general development teams.

As a consequence, the flight control team as described for the early days of FBW development will cease to exist in that form. The engineer who fully understands the interdisciplinary design approach still plays the important role, but now will have a supervisory and controlling function. Interdisciplinary experience and understanding of safety requirements and resultant implications for system development and validation must be used to control and monitor the detailed activities performed. The groups performing these activities will not necessarily be dedicated to safety critical tasks. This is required for the design activities as well as for the validation activities.

For example, execution of test activities for a flight control computer does not require a team of flight control engineers. It can be carried out by a team which has the capability and knowledge to test any other flight computer. The ground test plan for the flight control computer and the definition of the test requirements should be done by flight control specialists, however.

5.5 Validation Elements

Chapter 4 contained a description of a wide range of validation elements used in actual development

processes. For each system, it was proved through a safe first flight (and prototype flying) that the validation process had been performed successfully. As a consequence, it can be assumed that there are a variety of validation elements/techniques and methods available. These can be used to establish a validation process for any new system which would be close to state of the art.

It must be noted that any specific process or technique may not be valid for a specific project. An example is the need for an iron-bird. There have been projects and situations where an iron bird was not required. To come to a valid conclusion about the need for an iron bird on any specific project, one has to consider a range of specific program characteristics, such as:

- The extent to which systems used in the vehicle depart from past experience.

- Interactions between components which must be validated well before the airplane is developed.

- The validation requirements of the FCS, hydraulic system, landing gear, control surfaces, and system interconnections

- Availability of aircraft (prototypes) before first flight for system integration (as an alternative)

Decisions on specific facility requirements should be made on the basis of specific validation needs and cost/benefit analyses rather than on a purely historical basis. In the case of the iron-bird, the requirement for accurate performance evaluation during integration testing with flight qualified equipment in a realistic installation environment would most likely require the development of an iron bird.

The second general problem in discussing the validation elements in the abstract stems from the lack of a commonly accepted system development plan. The value and the importance of each validation element is dependent on its position in the development plan.

The system safety analysis is an example of this point. This analysis is a compulsory bottom up activity carried out shortly before first flight. The contribution of this kind of analysis to the safe conduct of the first flight is minimal because it usually relies on second source data, and does not approach the system from an independent point of view. To be more effective, the safety analysis should formally begin in the early design phase and should gradually build up a data base of system safety characteristics as the development of the flight system progresses.

Despite the problems of assessing validation elements in isolation, there was one approach which was judged to be universally important by the WG, and that was automated testing. This approach to validation testing carries with it technical, economic, and schedule advantages.

Many validation tests could be automated except for piloted confidence tests; piloted/manned stress tests; tests where mechanical manipulation is not practical without human or complex robotic mechanisms; or tests where a high degree of human interaction is required in analyzing intermediate outcomes. Automated test can either mean automated test execution (of a test procedure which could also be executed manually) or it could mean a test procedure utilizing the full potential of a computer driven test facility. The important difference between the two approaches is that the first one optimizes economic parameters and the second one primarily aims at a new quality of test coverage. It was felt that this second form is an area for future developments.

A cautionary note is offered-- one must ensure that unexpected outcomes in other systems or areas are visible. This requirs the recording of many unassociated parameters for a given test, and also requires human monitoring and data review. Automated testing is emerging as a principal tool because of cost, the large and increasing validation matrix size, test time availability, the desire for a high degree of repeatability, and the minimization of the impact of human error. It is the only practical way to build a 100% revalidation test capability for changes.

5.6 Validation of Piloted Simulation Systems for FCCS Validation

Piloted flight simulators are a powerful tool for designers studying the dynamic responses of pilots,

aircraft, systems and operating environments. Simulation has also proved extremely valuable for training pilots to manage their complex cockpit environment and to deal with a wide range of potential emergency conditions arising from failures and adverse operational conditions. For many years such simulations have been used successfully to predict and assess solutions to problems arising during the development of new aircraft and systems.

However, with very few exceptions, all aircraft acceptance and certification has required flight test demonstration. This situation is changing as the complexity of safety critical systems increases, presenting acceptance authorities with a very large set of potential failure modes. Also, improvements in the quality of flight simulation are increasing confidence in their ability to represent many flight situations well enough to reduce the range of conditions that require flight testing. By holding flight clearance testing within reasonable bounds, significant savings in cost and time can be realized, which is beneficial to both manufacturer and customer.

There already have been examples where piloted simulation has been used to demonstrate to acceptance authorities, a range of failure modes of a muti-channel fly-by-wire flight control system. From these simulation demonstrations, the authorities selected for flight demonstration, those situations that appeared to be most demanding and most probable. Economic and practical time limitations are going to increase the range of situations where piloted simulation will be used as a direct part of the acceptance (certification) process for both military and civil aircraft/systems.

This increasing use of piloted simulation for acceptance testing of aircraft is a natural consequence of the increasing physical understanding of aircraft and their operational environment, and confidence in the ability to extrapolate results from piloted simulation.

The confidence in the validity of modern simulators was achieved primarily from the subjective comments of experienced pilots involved in development flight testing of the aircraft. This is an important and necessary element of validation. The question is whether or not this approach is sufficient for the future. The members of the Working Group agreed that subjective comments are not sufficient to establish confidence in the validity of simulation for use in validating all aspects of flight critical control systems.

For example, it is not unusual in developing training simulators to try to compensate for motion and visual cue limitations, or computing delays, by altering the aircraft model to make it match actual closed-loop flight characteristics, based on pilot opinion. Pilots will then declare that the simulation is more 'like the aircraft' than the earlier version with the mathematically accurate aircraft model. This method of compensation for cueing deficiencies can often be acceptable for a training simulator, where the responses can be tuned to be a satisfactory representation of specific training tasks that are demonstrated on the aircraft. Although, even with training simulation there are frequently problems when users modify the training syllabus.

For validation or certification purposes, it is not acceptable to alter aircraft or system models to compensate for cueing deficiencies. Although this approach can improve the apparent validity of the simulation to the pilot in those areas experienced in flight, there can be little confidence that the simulator is presenting adequately those vital situations which are not going to be demonstrated in flight, or which dramatically affect system operation. The following issues should be addressed by the combined FCS/simulation/pilot team:

- For what sets of conditions had validity been confirmed from flight test?

- What confidence can be placed in simulation of conditions outside the validated range?

The art of simulation compensation is dependent on a wide range of factors, including physical characteristics of available cueing systems (visuals, motion, sound, control loading, g-seats, and other simulation features), computing systems (architecture, speed, capacity, etc), aircraft category, specific operational task, and many other factors. Thus acceptance authorities and manufacturers have to establish confidence in simulation results through either relevant flight test validation, or from an identification and acceptance of the validity of compensation techniques. Pilot acceptance alone is not sufficient.

Government publications seek to inform manufacturers and acceptance authorities of recent experience

with the use of simulation in both development and acceptance/certification testing. They addresse a wide range of practical issues arising in modeling, provision of cues, and in integrating the aircraft and simulation systems. They also provides guidance on validation methods and outlines issues which may need to be considered by acceptance authorities. These reports finally consider areas where simulation is likely to be a major element in acceptance testing and the impact of and need for development in simulation technology to meet a wider range of these crucial areas.

CHAPTER 6

TRENDS IN FLIGHT CONTROL SYSTEM DESIGN AND IMPACT ON VALIDATION

6.0 Introduction

Flight control system design concepts and validation test technology have advanced significantly over the past 20 years since the advent of digital systems. Validation techniques and test methodology have been influenced by experiences in the qualification of systems, and system developers have converged on several accepted methods for both test and analyses. For the most part, however, validation technology lags the advances in flight control system design. This has led to problems in the past with flight system validation. For example, multi-channel fault tolerant flight control systems were developed before suitable real-time multi-channel diagnostic/test equipment was developed to support the diagnosis and validation of sophisticated fault-tolerant architectures. In other cases, systems were developed with no convenient way of conducting multi-channel validation tests. Considerable time elapsed before software and hardware methods were developed to both stimulate the systems and instrument them to capture the system response for subsequent analysis.

Where ground support equipment and test methods were developed concurrently with the advanced systems they were to support, validation has proven to be less costly and more time efficient. Much has been learned about how to validate systems to a high degree of confidence, but advancements in flight critical control systems require a complementary and continuous updating of validation tools and techniques.

This chapter forecasts trends in flight control system design based on expected aircraft trends, and identifies the potential impact on the validation process. This should provide a basis for developing research and development programs in the area of validation techniques, and for anticipating validation requirements. Although considered mature by some standards, flight control system technology is expected to advance considerably by the year 2000[1]. The forecast in this chapter is purposely far reaching, extending into the first decade of the 21st century. It is intended that this forecast be a catalyst for developing improved validation techniques simultaneously with new flight system concepts, and for influencing flight system design decisions. It is expected that many of the potentially adverse impacts on validation of these advanced flight critical control system designs can be avoided by anticipating the validation requirements early in the design process, and by using many of the emerging structured validation tools and techniques described in the next chapter.

6.1 Aircraft Projections

6.1.1 Introduction

Aircraft configurations and specific mission capabilities are derived from statements of need from the using military organizations, and are subsequently adopted and approved by governments themselves. These needs are based on a complex and lengthy set of factors including threat assessments, obsolescence of current vehicles, budgets, timing, and national policy issues. Although it is difficult to forecast the exact path of new aircraft developments, it is possible to project trends in vehicle capability. These broad trends then provide a basis for projecting trends in flight control systems and the impact on validation.

6.1.2 Maneuverability

Aircraft maneuverability requirements are projected to continue to increase, especially in the the regime beyond maximum lift, often termed, "post-stall". Experimental aircraft are probing the utility of maneuvering at very high angles of attack, and the potential benefits of such capability have been

examined on high fidelity simulators. Thrust vectoring technology is expected to provide the control augmentation required to operate safely and effectively in this regime. Control requirements are likely to include significant propulsion/flight controls integration, as well as a high degree of integration with the fire control system.

Advances in rotorcraft maneuverability have been significant in the late 1980's, and the trend toward larger maneuvering envelopes is expected to continue. The impacts on control systems are projected to be in the areas of rotor dynamics, control integration, and man-machine integration. Expansion of the maneuvering capability will also require controls and displays integration for safe operation in nap-of-the-earth operations.

6.1.3 Survivability

The trend toward reducing the signatures or observability of aircraft is expected to continue indefinitely. Large deviations from past design outcomes are likely, due to new emphasis on signature reduction. The design approaches used to achieve lower levels of observability will produce radically new configurations affecting aerodynamics, materials and structures, controls, propulsion systems, and avionics. High levels of integration between subsystems are expected, with earlier incorporation of control system capability in the design process.

6.1.4 All Weather/Night Operational Capability

The mission benefits of all weather/night operations are so significant that this capability is expected to be introduced in much of the fixed wing and rotary wing aircraft fleets of the future. This capability implies a high degree of sensor fusion and advanced pilot-vehicle-system integration techniques. For single pilot operations, decision aiding is expected to achieve a higher level of importance. A larger set of the aircraft's systems will achieve flight-critical status, and be integrated with the flight control system.

6.1.5 Short Take-off and Landing Capability

Technology advances in propulsion thrust-to-weight, and advanced aero-propulsion concepts are expected to make short take-off and landing, or short take-off and vertical landing capability more attractive as a design option in the next decade. Integrated thrust vectoring and reversing capability suitable for take-off and landing performance enhancement as well as in-flight maneuvering could spawn a new class of aircraft. This vehicle class is known to require extraordinary levels of integration of the aerodynamic and propulsion controls, avionics, and cockpit systems.

6.1.6 Unmanned Vehicles

It is expected that there will be an increase in the use of unmanned aircraft for reconnaissance and combat applications. These unmanned vehicles are also projected to have increasing levels of autonomous operational capability, and may work in an intermetted arrangement with manned aircraft, as a "team." Ensuring safety in the overflight area as well as airborne safety will pose especially difficult challenges as the performance and autonomy of these vehicles increases.

Development of the unmanned flight crucial systems may involve the use of manned testbeds in which the unmanned vehicle systems are installed, in order to provide an adequate level of safety for initial flight tests. It is also expected that there will be strong requirements to reduce the cost of the flight control systems to enable low cost vehicles. This will have the effect of reducing flight control system redundancy, and will pose an additional challenge to the systems design to ensure ground safety and vehicle integrity.

6.1.7 Hypersonic Vehicles

Significant international research and development efforts are underway in lightweight structures, advanced heat shield concepts, and scramjets. These enabling technologies, if sufficiently advanced in the 1990's, could lead to airbreathing vehicles with hypersonic capability in the early part of the

21st century. The addition of rocket propulsion could provide access to space with such vehicle concepts. The necessary design-level integration of propulsion systems, aerodynamics, flight controls, thermal management, and cockpit systems would be unprecedented. The interactions of the various elements of the vehicle in the design, and in flight, will demand an extension of the flight control state-of-the-art to achieve hypersonic flight. The control systems will play an ever more prominent role in these vehicles for the realization of a flight vehicle, because of these strong interactions.

6.2 Systems Integration Trends and Validation Impacts

6.2.1 Introduction

Based on the vehicle projections and trends, it is clear that an ever increasing amount of integration will be required of the various subsystems of the aircraft. There is also a strong trend to integrate many of the subsystems on a vehicle through the flight control system. It is expected that this trend will continue because of mission and performance benefits. Research and development into advanced pilot-vehicle interfaces such as helmet-mounted sights/displays, virtual displays, voice command, and tactile/aural feedback suggests more highly integrated pilot-vehicle-system configurations in future aircraft. The impact on validation of these highly integrated systems of the future will be:

a) more testing required at the integrated system level, including the pilot, because of the need to include more of the system to be able to evaluate the operation of any one of the systems;

b) more difficulty in the hand-off from the vendors' and suppliers' validation programs to the integrated systems validation process usually conducted by the prime contractor;

c) a high demand for interdisciplinary expertise to design and understand the validation test requirements for such integrated systems;

d) the requirement for a larger fraction of the testing to take place with the pilot or a sophisticated pilot model in a high fidelity simulation;

e) the requirement to deal with the embedded software in these various subsystems as flight critical;

f) the requirement for more complex and complete real time environment models, such as models of the thermal conditions over the entire vehicle to be able to evaluate integrated trajectory and cooling management systems for high speed aircraft.

6.2.2 Fire/Flight Control System Integration

There has already been significant integration of the fire control and flight control systems on advanced fighter aircraft[2]. One such system configuration is shown in Figure 6.1. The further integration of terrain following, terrain avoidance, digital terrain map, threat warning systems, automated trajectory guidance and automated maneuvering and attack systems are likely to be common features of future aircraft. The avionics and flight control systems on most vehicles of the future will be integrated with redundant data buses for information transfer for cooperative functions. One impact of this integration will be the inclusion of more traditionally non-critical elements into the flight critical domain. This can compromise the desire to partition critical and non critical functions.

In rotorcraft, semi-automated nap-of-the-earth flight is expected to be a major feature of the flight control system. The system will integrate trajectory control with terrain following, terrain avoidance, obstacle sensing, and obstacle avoidance. In both fixed and rotary wing aircraft, there will be a continuous attempt to provide sufficient intelligence and integrity in the flight control and avionics systems to allow single pilot operations in complex missions.

6.2.3 Highly Integrated Flight/Propulsion Control Systems

Major advances are being made in the integration of flight and propulsion control systems for high performance aircraft. Active engine stall margin control (Figure 6.2) has been achieved in flight with major performance gains[3]. Future high performance aircraft will extend these techniques to complete inlet/engine/nozzle/flight control. Thrust vectoring capability will provide another major effector, for use in maneuvering as well as for improved take-off and landing performance under short or rough field conditions. Performance optimizing control algorithms will optimize total aircraft performance through control of vehicle trajectory while optimizing propulsion system and vehicle aerodynamic performance. It is expected that such systems will also account for installed engine differences and aging. Multi-mode control of the airframe and propulsion system will provide the ability to maximize thrust-minus-drag, thrust specific fuel consumption, or engine life.

Airframe-mounted and engine-mounted controls are expected to be separately provided by airframe and engine manufacturer, but be closely integrated functionally. The validation effort will have to involve much more activity at the system level because of the interaction of the flight and propulsion control systems. More capable ground test facilities and more extensive flight programs will be required to validate the total system functionality and safety. There will be more failure modes to contend with, and more capable analytic validation tools will be required to deal with the higher degree of system complexity.

Recent studies and preliminary designs of airbreathing hypersonic aircraft has raised the specter of the necessity of highly integrated aerodynamic, propulsion, trajectory, thermal, fuel, and power systems management. Unprecedented interactions between the vehicle aerodynamics and propulsion system are foreseen for advanced hypersonic airbreathing vehicles, where the forebody and afterbody of the vehicle also act as part of the inlet and nozzle. The complex nature of the missions, the narrow margins on trajectory, and the requirements on structural and propulsion efficiencies have led to the conclusion that all subsystems must be combined and controlled in a totally cooperative manner to achieve both safe and efficient flight. This level of systems integration will place new demands on ground facilities because of the need to include more hardware "in the loop" in simulation. Test automation is seen as a mandatory approach based on the sheer volume of testing and the complex interactions between subsystems which must be observed and considered during testing.

6.3 Emerging New Functional Capability

6.3.1 Introduction

High capacity airborne computing and the integration of subsystems will allow a continuing expansion of the number of flight control functions mechanized on future aircraft. These new functions are expected to span the spectrum of new mission capability, safety, and subsystem performance improvement. Expected functions and their impact on validation are described in this section.

6.3.2 Decision-Aiding Systems

The implementation of decision-aiding systems ("expert systems"), in a real time environment, is expected to be achieved in the 1990's. The application of these decision-aiding systems to functions today considered flight critical may be far in the future, but systems that indirectly affect safe flight and mission accomplishment are already in advanced research phases. These systems will impose severe demands on the validation process. There are today no accepted methods of validating such systems for flight crucial functions, because the domain of operation cannot be precisely defined, as for conventional systems.

It is also quite likely that there will be integrated symbolic and conventional processing in actual applications onboard an aircraft, requiring validation methods that bridge the gap between these two processing approaches, and systems. The domain of possible functional interactions and system actions will have to be understood sufficiently to provide a validation matrix which results in an acceptable level of risk for operational use. In addition, the visibility into the system required to

understand the system operation will require new approaches to implementing stimulation signals, and test monitoring functions for validation purposes.

One major new facet of such systems used onboard aircraft is the technology required to interface the crew and the decision-aiding systems. As the "apparent intelligence" of the system increases, it will be necessary to include more of the human factors interactions in the validation process. The significant challenges of establishing the credibility of a decision-aiding system in life-critical situations will be a challenge to the validation process. Techniques currently used to validate intra-crew operational procedures may have to be applied to the human-machine system validation process.

6.3.3 Self Repairing, or Reconfigurable Flight Controls

The number of control effectors is projected to increase, especially for high performance aircraft, which will employ multi-axis thrust vectoring, canards, multi-segment leading and trailing edge flaps/flaperons, and stabilators with pitch and roll capability. Forebody controls may also emerge to control forebody vortices for vehicle control or recovery purposes. Careful selection and placement of these various effectors may provide sufficient force and moment redundancy to allow substantial levels of maneuverability and operation following the loss of an actuator or other flight control system element, or following battle damage. Onboard systems would identify the loss of control and reconfigure the remaining effectors and control algorithms to provide continued mission capability (Figure 6.3).

A further extension of this design philosophy could also lead to reducing the actuator redundancy level, and hence actuation system complexity, by capitalizing on controller redundancy instead. For example, it may be possible to distribute the control redundancy between actuators and effectors to reduce the number of redundant channels per actuator, and the attendant hydraulic and electronic systems, to achieve lower system life cycle cost.

This design approach will require what is termed "self-repairing" or "reconfigurable control" technology[4]. These control systems will require some degree of on-line parameter identification and reconfiguration capability beyond that used in current systems. One impact on validation will be the requirement to identify the domain of possible configurations. This could be a very difficult task in the case of self-repairing controls where first, system identification takes place to identify the aircraft state and the control forces and moments available, and second, control reconfiguration takes place, with at least adjustments in gains, and possibly with changes in the control structure. The validation of such systems must account for the entire range of configurations possible, or validate the boundaries of operation. This problem has already been faced in advanced adaptive flight control system experiments which had a fixed set of effectors. The validation difficulty of these simpler approaches suggests much more difficulty for systems which may have to accommodate an unknown force and moment combination.

6.3.4 Total Vehicle Energy/Thermal Management

Studies and preliminary designs of hypersonic vehicles carrying cryogenic fuels show the requirement for the Vehicle Management System to manage and control the total thermal environment, including engine operation, cryo fuel conditioning, active structural cooling, and trajectory. In such cases, the environment and the interactions are not easily modeled or simulated. The validation of such systems may in fact require a complex test involving parts of the actual structure, as well as the control system, in order to test the entire closed loop system. For example, a section of actively cooled structure, with cryogenic fuel cooling systems being controlled by the actual Vehicle Management System hardware and software, would have to be subjected to heat loads in order to validate the closed loop system to a sufficiently high degree of confidence for manned flight. This would be a literal "hot bench" test.

6.3.5 High Bandwidth Flight Control System Function

Control system designs requiring higher bandwidth operation are projected to increase for future high performance aircraft, driven by advanced aircraft configurations and structural designs. For example, active vibration and load control modes are expected on advanced rotary wing aircraft, as the maneuver and speed envelopes are increased. In some cases, the flight control system for fixed

wing aircraft will also be asked to provide active damping of structural modes where aeroservoelastic problems are identified too late in the development cycle to be amenable to a structural fix.

A related situation is the case when the load-to-actuator natural frequency ratio is low. Such systems are characterized by a high degree of performance sensitivity to the actuator stabilization method, structural compliance, sensor dynamics, and structural dynamic characteristics. These systems will have a significant impact on validation technology. In order to validate such systems, a high premium will have to be placed on system and environmental models and on sensor signal simulation from sensors and ground rigs that must provide information on actual dynamic actuation loads or structural dynamics. There will be a much more critical requirement to model and faithfully reproduce the important structural modes and compliances, as well as control sensor and actuation characteristics over a wide frequency range.

6.3.6 Active Local Aerodynamic Control

The emerging ability to understand and predict local aerodynamic flow using computational fluid dynamics is expected to enable active control of local flow. For example, active circulation control may be utilized to increase lift of a V/STOL aircraft at low speeds or reduce downloads of a tiltrotor in hover. Control of forebody vortices on high performance aircraft could improve maneuvering capability at high angle of attack. The extension of this concept to actively controlled engines is also possible. In this concept, active internal engine controls are utilized to relax aerodynamic margins and design constraints. Validation of these control systems will necessarily include more in-flight testing due to the inability to duplicate the aerodynamic environment adequately in ground facilities. It may be necessary to develop integrated computational aerodynamic, and global aerodynamic simulation models for satisfactory ground testing or analysis of the total system. A new discipline of "computational controls" may be required to support the implementation of local flow control systems.

6.4 Architecture

6.4.1 Introduction

Redundant, parallel channel system architectures will continue to provide the basis for achieving fault tolerance in flight control systems of the future. Significant variations in the actual realizations of this basic approach are emerging, however, and bring with them new validation issues. There is one overriding characteristic of future systems that will have a dominant effect on architecture, namely significant increases in computer power. Just as ground-based supercomputing systems have enabled major advances in computational fluid dynamics, airborne supercomputing will likewise spawn a major expansion in the scope and character of onboard processing

The developing architectural branches of redundant system structures, and the impact of increased computational power are expected to place new demands on validation technology. There is also a clear trend to eliminate the independent back-up system for digital fly-by-wire flight control systems without compromising the levels of operational capability and safety afforded by combinations of redundant primary and back-up flight control systems in current generation systems.

6.4.2 Hardware-Implemented Fault-Tolerance

In contrast with most contemporary systems which implement fault tolerance and systems management functions almost entirely in software, the hardware-intensive system approach[5] seeks to remove from the primary flight control computers some of the systems management software functions which are not expected to undergo much change over the lifetime of the system. Redundancy management functions such as voting planes, synchronizing mechanisms, bus interfaces, and bus contention logic are instead embedded in special purpose computing hardware, isolating this software physically from the flight control function software. This approach reduces the potential of adversely unintentionally affecting critical executive executive system software functions during the change cycles usually associated with the flight control functions themselves.

For major, subsequent applications of the same architecture, it might not be necessary to revalidate

many of the systems management functions to the degree that it was validated the first time. The overall system operation may be more complex than for conventional systems, however, and the initial validation process may be more difficult than for conventional systems. With this approach approach to implementing systems management functions, it will be increasingly important to design in provisions for validation test stimuli and test monitoring, because such features may be difficult to provide late in the hardware development process.

The Working Group believed that the overall benefits of software-implemented systems management functions were so significant, that this hardware intensive approach would not be dominant in the next 15 years.

6.4.3 Dynamic Resource Allocation

It has been recognized for some time that it would be desirable to utilize unfailed components in a failed channel in other channels. This feature is shown conceptually in Figure 6.4 a and in an actual implementation in Figure 6.4 b. This approach would provide access to unfailed I/O, memory, or processors across channel boundaries, making the multi-computer system more highly fault tolerant, because a failure in one subelement in a channel does not prevent unfailed subelements of a different type in that channel from being used by the system. This architecture can be characterized as "dynamic redundancy" because system elements are not dedicated to one channel, and, may in fact, be allocated to different channels as failure sequences occur.

There could be at trend toward utilization of this architecture to meet very large mean time between unscheduled removal periods, where all possible resources must be used instead of dicarding entire channels due to a failure isolated to a particular element. The validation challenge is to ensure that dynamic resource allocation algorithms themselves are error-free, and that only unfailed elements are used again.

6.4.4 Embedded Replicated Subchannels

One major variation of classical parallel channel architecture is that of embedding, in each of the redundant channels, dual subchannels[6]. This allows within-channel failure detection to be accomplished through relatively simple hardware comparison monitoring at the output of the two subchannels (Figure 6.5). Miscomparison would result in a self-fail declaration in that channel. This approach replaces relatively complex in-line failure detection software and reduces cross-channel communication requirements with relatively simple cross-subchannel monitoring. This architecture could reduce the software burden associated with the implementation of many current redundancy management approaches, and simplify the modeling of the total system for analysis of fault survivability and coverage. Likewise, validation of the system would be simplified due to the ability to handle many faults by similarity analysis. The test burden could also be reduced by minimizing the revalidation required when new application software is introduced.

This architectural approach also enables overall flight control computing architectures to be designed to materially reduce the risk of common-mode failures as described in reference 7. Thus this approach is expected to lead to new system fault tolerance concepts, not to simply replace elements in existing architectures.

6.4.5 Dissimilar Embedded Subchannels

Another architectural variation already emerging is the use of cross-checking subchannel pairs employing dissimilar software and in some cases, dissimilar hardware[7]. In the case of the A-320, this approach is used to switch to an alternate conntroller or functionally independent aerodynamic control surface, as was described in Chapter 2.

A general arrangement of such a system is shown in Figure 6.6. This architectural approach addresses hardware failure detection in a manner similar to the previous architecture, but also seeks to provide some protection against common mode software or hardware faults[8]. In the dissimilar dual- subchannel approach, it is postulated that the likelihood of a common design or implementation error will be reduced, based on the assumption that there will be a low value of coherence of

catastrophic faults in both channels if design and implementation is carried out by independent teams, using different languages, compilers/assemblers, and then implemented in different processor hardware. If low coherence of faults and errors is actually achieved, this concept could simplify the validation process in terms of establishing that no catastrophic common mode software or hardware faults exist.

There are various means of managing failures and reconfiguring the system with this scheme, and the validation issues are somewhat dependent on the precise reconfiguration strategies used. The overall validation approach is expected to be similar to that for classical parallel architectures. If the two dissimilar channels are treated as two completely independent and separate systems in the development process, there would be a doubling of effort required to validate the two dissimilar systems, assuming they are of approximately equivalent complexity.

It is also possible that the dual implementation could provide benefits and efficiencies in the validation process which could minimize the extra validation burden. For example, This approach affords the ability to cross-compare results from two flight-quality implementations of the same specification. It is expected that this aspect of the design could enhance the quality of the validation process.

This class of architecture also alters the approach taken to establish an acceptably low probability of loss of control due to a design or programing error. In conventional systems, this point is established by the results of the test program and in many cases the availability of an independent back-up system. With this new approach, immunity to a common mode catastrophic fault is predicated on the joint probability function that a similar defect has been introduced in two independent processes. This circumvents the difficult challenge of proving that common failure modes do not exist, instead, demonstrating that the configuration can survive such errors, because they will occur only in one subchannel of each channel. This approach does place a high premium on ensuring that the common specification, which drives both dissimilar implementations, does not cause two different implementations of the same catastrophic specification error.

6.4.6 High Availability Architectures

Highly fault tolerant flight crucial digital flight control systems have been developed and applied to military and civil aircraft, providing improved performance and mission capability with high levels of safety. There are, however, increasingly demanding requirements for reducing the life cycle cost of such systems, and for increasing the availability, or operational readiness of aircraft. These requirements are expected to lead to systems concepts which would have dramatically reduced rates of unscheduled maintenance actions, and which would have dispatch capability with failed flight control system components. A requirement could emerge that would relegate flight control system component removal to scheduled maintenance periods, which could be every several thousand hours. The probability of loss of control per hour, or per flight, would be required to meet the levels of current systems, which are generally required to be failure free prior to flight.

In order to provide a system which would rarely, if ever be removed from the aircraft, except for scheduled maintenance, it has been proposed[9] that computational elements or channels be replicated multiple times, and have multiple levels of redundancy within each channel. Computational paths may have dissimilar hardware and/or software (Figure 6.7). Control may also be distributed among redundant or split control surfaces, further dispersing control paths and increasing failure tolerance. One significant aspect of these systems is that they defy simplistic characterizations, such as "fail-operate", "quad", or "triplex" systems.

The challenge in validating such systems is expected to be in the modeling of the systems for failure mode and failure tolerance analysis, and in the development of test and analyses matrices which are necessary and sufficient to demonstrate the multitude of reconfiguration combinations which can exist. A high premium will also be placed on the in-flight diagnostics systems to correctly identify, log, and manage the failure history over long, unattended periods of time. Complex test hardware and software environments are another natural fallout of these systems because of the multiplicity of hardware and software environments.

6.4.7 High Throughput Architectures

Distributed processing is already commonplace in advanced aircraft flight systems as a means of meeting the increasing airborne processing requirements for avionics, flight controls, displays, actuators, fuel management systems, hydraulic management systems, and electrical power management systems. In the longer term, it is anticipated that there will be an increased need for more throughput. It is also expected that flight critical software program size will continue to increase. The validation burden of simply dealing with increasing amounts of flight critical software is in itself expected to be a significant issue.

A considerable amount of research is on-going in both the US and in Europe in the area of sensor fusion and information integration and display. Many new functions are also anticipated for advanced aircraft as was discussed previously. The advancements in mission sensors and the increase in mission complexity will create the requirement to more fully integrate the information from various sources and provide to the crew a more complete picture of the environment, the aircraft state, mission strategy options, and task advisories.

These demands will create the requirement for processing well beyond the capability of today's airborne computers. The solutions to this problem are anticipated to be in terms of multiprocessing, parallel processing, distributed processing, and massive memory capability, even though it is expected that the core stability and control functions of future flight control systems will not require this extensive level of computational capability.

If the functions implemented in these high speed processors increase in importance for the accomplishment of the mission, and for the safe operation of the aircraft, they may become flight critical. This will result in the demand for fault tolerance of these functions. Most of the software development work in parallel processing for computational fluid dynamics for example, has been in achievement of very high throughput rates, with minimal requirements for fault tolerance. Thus, the technology developed for large ground-based supercomputers may not be of direct value to the development of their airborne equivalents.

Therefore it is anticipated that new architectures will emerge from these requirements, either as derivatives of architectures now used for high speed ground processing for computational fluid dynamics, or in entirely new architectures such as fault tolerant parallel processors, massively parallel processors, or sets of processors arranged in various three dimensional arrays, all being being studied in research institutions. It is expected that there will be a reduced ability to conduct some of the validation tests on non-flight hardware emulation of these new sophisticated computer systems, because the flight hardware itself will dictate the function and operation of the software, and it will not be possible, in many cases, to achieve sufficient emulation on ground-based host computers.

There may also be a requirement for more validation emphasis at the system level. Valid qualification tests may not be possible utilizing subsets of the system because of the highly distributed and interactive nature of the software and hardware operation. The amount of software which can affect flight critical functions is expected to increase substantially, thus dictating broader application of validation processes historically reserved for flight control systems.

The application of these more exotic architectural approaches may be seen first in civil, commercial air transportation first, because of the stringent requirements for extremely low probabilities of loss of control imposed on these aircraft by civil certification authorities. The performance or life cycle cost advantages of these systems may extend their application to military aircraft.

6.4.8 Back-up Systems

The cost and complexity of developing totally independent backup flight control systems has reached the point where it is expected that future systems will rarely employ such independent control methods. This cost and schedule burden arises from the need to validate primary and back-up systems for every change that could impact the operation of vehicle. The enormous level of integration of avionics and control systems has the effect of demanding validation of both primary and back-up systems for a very large number of changes that are made elsewhere in the system.

The level of experience and development engineering discipline for reliable flight control system design without an independent back-up is expected to increase substantially during the next decade for military aircraft. Thus the independent back-up system philosophy is expected to change during this time, and reduce the perceived demand for such capability.

Where alternate control capability is determined to be necessary, reversionary software modes are expected to be implemented to provide some coverage for primary control software faults[10]. Flight experience with a primary quadruplex digital fly-by-wire system with no-backup hardware or software has demonstrated the confidence with which flight crucial digital fly-by-wire controls can be designed for fighter aircraft[11].

6.5 Flight Control System Component Trends

6.5.1 Introduction

It is expected that continuing improvement in reliability and capability will be made in critical components of flight control systems. In addition, new families of components are expected to emerge during this time period which will have a direct effect on flight control system design and validation

6.5.2 Sensors

In a manner similar to the functional integration that was described in the previous section, there is a strong trend toward the integration of sensors on advanced aircraft. For example, it is well within the state-of-the-art to integrate of navigation and flight control sensors for both the navigation and flight control functions. It is also expected that the number and complexity of sensors integrated in vehicles in order to provide increased mission capability will continue to increase. For example, advanced systems and displays proposed for future aircraft would permit the pilot to fly the safest path through a threat area. The pilot would also have the capability of real time mission replanning, based on data-linked information from ground and airborne sources. It is obvious that a massive amount of sensor fusion and processing will be required to implement this function. It is also quite likely that these systems will move from advisory functions to safety critical and mission critical functions as the systems increase in utility. There is also research and development underway to look at new ways of interneting aircraft which would exchange sensor information and coordinate their flight paths and active maneuvering through their control systems. This amounts to a spatially distributed but integrated sensor set.

The heavy demands on environment model fidelity and on the simultaneous validation of much larger systems, potentially including multiple vehicles is evident. The impact on the validation process of these sensor trends is the requirement to validate much larger systems and to validate avionics and flight control systems much less independently. In-flight testing is expected to play a much bigger role in the validation of such systems

New sensors such as optical air data sensors or multiport flush air data systems are in the research and development stage. These systems require remote processing or a considerable amount of additional air data processing within the flight control computer. An increase in the number and type of optical sensors interfacing with the flight control system is expected to increase dramatically. The validation impact is likely to be experienced in the stimulation of these sensors with optical signals for complete system-level validation.

6.5.3 Common-Modules

An emerging hardware implementation approach is expected to reduce the overall systems validation burden. The use of common hardware modules in various subsystems will allow validation by similarity, when a particular module has undergone the extensive use and validation in several different areas of the flight system. Common module central processor units, memory units, input-output units, power supplies, and bus hardware may appear in the flight control, fire control, radar processing, and offensive/defensive avionics systems. Modules may approach the character of

current generation integrated circuits in some applications, thereby easing the overall validation burden for such systems.

6.5.4 Optical Systems

In a prior section it was projected that optical sensors would increase in use for future aircraft. It follows that the use of other optical components in flight critical flight control systems will increase also over the next decade, due to demands for higher bandwidth data transmission, increased immunity to the electromagnetic environment, and the requirements to handle more optical sensors.

Optical data transmission links, position and rate sensors, and air data systems are already projected for future aircraft. As the number of optical transmission paths and sensors increases, it is expected that there will be an increased benefit of optical processing at various nodes in the system to avoid conversions to electronic signals at intermediate points. Multi-spectral optical sensors currently used in satellites for earth observation may spawn derivative devices for target identification or for terrain identification. The increase in the amount of optically formatted information will continue to increase over the next 10-15 years, suggesting the requirement for more direct processing of optical signals prior to conversion to electric signals. Optical processors are currently in the basic research phase and would require an entirely new technology for validation.

Validation tests are expected to use increasing amounts of optical stimulation in the laboratory environment. Optical laboratories will also have to be integrated with the hot bench and rig test equipment and flight sensors.

6.8 Concluding Remarks

Dramatic increases in airborne computational power, combined with the increasing level of integration will be the principal design drivers for future flight control systems. It is expected that more functions will be included in the set of those having a direct impact on safety of flight. Another dominant force in future system design will be the demand for increased reliability, maintainability, and availability of aircraft, providing increased effective force levels and reduced life cycle costs. System architectures will take on less easily characterized hybrid configurations.

The volume of software and the number of system elements which must be treated as flight critical is expected to increase continuously. There is also a clear trend to eliminate dissimilar and independent back-up flight control systems. On-line decision-aiding systems, are expected to aid the pilot/crew in complex missions and tasks. More highly integrated vehicle management systems may place the entire flight system in the "flight crucial" category for very advanced vehicle of the next century, such as airbreathing hypersonic aircraft.

For highly unstable aircraft, or vehicles implementing artificial structural mode stability, an additional burden is placed on the revalidation aspect of qualification. For such flight control functions, there is little margin for error. The high premium placed on correctly identifying all possible impacts of any design change will require a higher level of sophistication in the revalidation processes. One hundred percent regression testing may be the necessary outcome of the implementation of such functions, unless partitioning specifically to support validation is designed into the system from the beginning. This approach suggests massive test automation as the principal mitigating factor in maintaining a practical validation program. The combination of bandwidth requirements and criticality to change may in fact result in hybrid systems which physically separate and "harden the "core system" from other systems.

For some of the functions and systems previously forecasted, there may be no alternative other than to conduct some portion of the validation in an incremental fashion, in flight. This is already the case in validating the flutter margins, stability, and dynamic load capability of aircraft. Although proof load testing is accomplished on the ground, there is no ground test of a full scale vehicle capable in itself of validating the flutter margin. Thus, flight flutter clearance is an early and mandatory part of every flight test of a new aircraft. In the case of more highly integrated pilot-vehicle-systems, or advanced functions having a direct impact on safety of flight, it may be necessary to develop new validation tests that can be accomplished incrementally in flight, in a similar manner. In this kind of

validation, as in the case of flight flutter clearance, there must always be a guaranteed safe return to a previously known condition or configuration. The integration of flight control functions, such as terrain avoidance and terrain following with advanced sensors or systems, such as digital terrain maps, may require a larger portion of validation testing to be accomplished in flight due to the difficulty of adequately representing the environment in a ground-based laboratory.

The boundaries between non-flight-crucial and flight crucial systems are projected to dissolve with increasing integration of systems. The challenge will be to maintain the validation state-of-the art sufficiently high so as not to impede the introduction of advanced systems in future aircraft. To achieve this goal, the validation process must be thoroughly embedded and integrated with the system design process itself. Furthermore, the level of research in the area of validation technology itself should be commensurate with the rate of progress in systems technology forecasted.

Finally, experience with the current generation of flight control systems shows that the validation effort can be bounded through the use of judicious partitioning and protection of the most safety-critical flight control functions, such as inner loop stabilization. Design trades between level and method of functional integration, and the validation requirements will have to be more deliberate in the future. History also shows that performance and mission capability are weighted much more heavily than validation difficulty and testability. Therefore it is imperative that validation technology receive sufficient attention to permit the advanced systems of the future to be implemented with reasonable cost and safety.

References

[1] "Aeronautics Technology Possibilities for 2000: Report of a Workshop", Workshop on Aeronautical Technology: A Projection to the Year 2000, Aeronautics and Space Engineering Board, Commission on Engineering and Technical Systems, National Research Council, National Academy Press, Washington, DC, 1984.

[2] Ramage, J.K., "AFTI/F-16 Automated Maneuvering and Attack System Configuration Development and Integration",Proceedings of the IEEE 1986 National Aerospace and Electronics Conference, Dayton, OH, May 19-23, 1986, pp 538-549.

[3] Putnam, T.W., Burcham, F.W., Jr., Andries, M.G., and Kelly, J.B., "Performance Improvements of a Highly Integrated Digital Electronic Control System for an F-15 Airplane," NASA TM-86748, 1985.

[4] Urnes, J.M., Stewart, J., and Eslinger, R., "Flight Demonstration of a Self-Repairing Flight Control System in a NASA F-15 Fighter Aircraft", AGARD-CP-456, April, 1990

[5] Lala, J.H., "An Advanced Information Processing System", 6th AIAA Digital Avionics Conference, Baltimore, MD, Dec. 1984.

[6] Driscoll, K., "Multi-microprocessor Flight Control System, 1982", IEEE/AIAA 5th Digital Avionics Systems Conference, 1983

[7] Corps, S.G., "A320 Flight Controls", Minutes of the 29th Symposium of the Society of Experimental Test Pilots, September, 1985.

[8] Avizienis, A., and Lyu, M., "On the Effectiveness of Multiversion Software in Digital Avionics", AIAA Computers in Aerospace VI Conference, Oct. 1987

[9] Dennis, R.W., and Hills, A.D., "A Fault Tolerant Fly-by-Wire System for Maintenance Free Applications", AGARD-CP-456, April, 1990.

[10] Deets, D.A., Lock, W.P., and Megna, V.A., "Flight Test of a Resident Backup Software System", NASA TM-86807, January, 1986.

[11] Smith, T.D., Yeo, C.J., and Marshall, R.E.W., "Ground and Flight Testing of the Fly-by-Wire Jaguar Equipped with a Full Time Quadruplex Digital Integrated Flight Control System", AGARD 35th Guidance and Control Panel Symposium, Portugal, 1982.

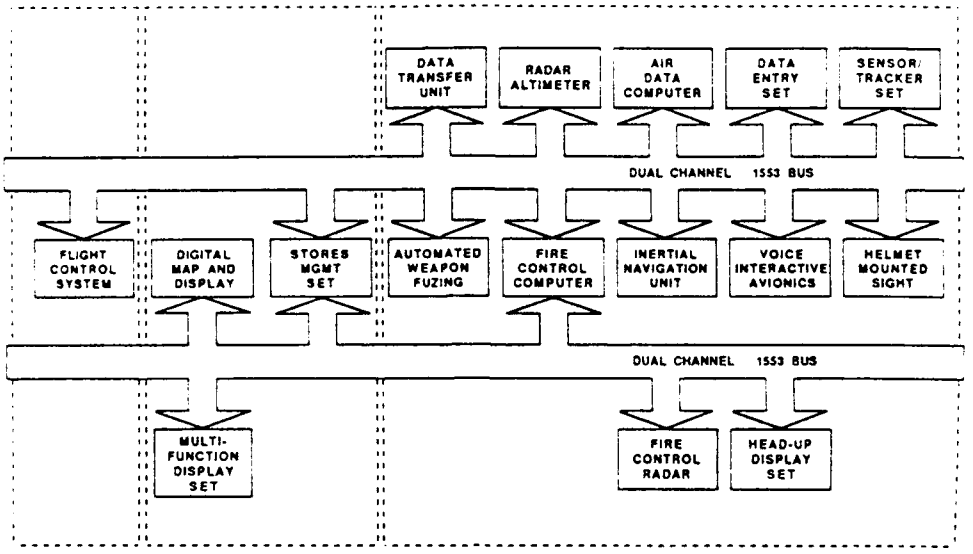


Figure 6.1 Integrated Flight Control/Avionics System

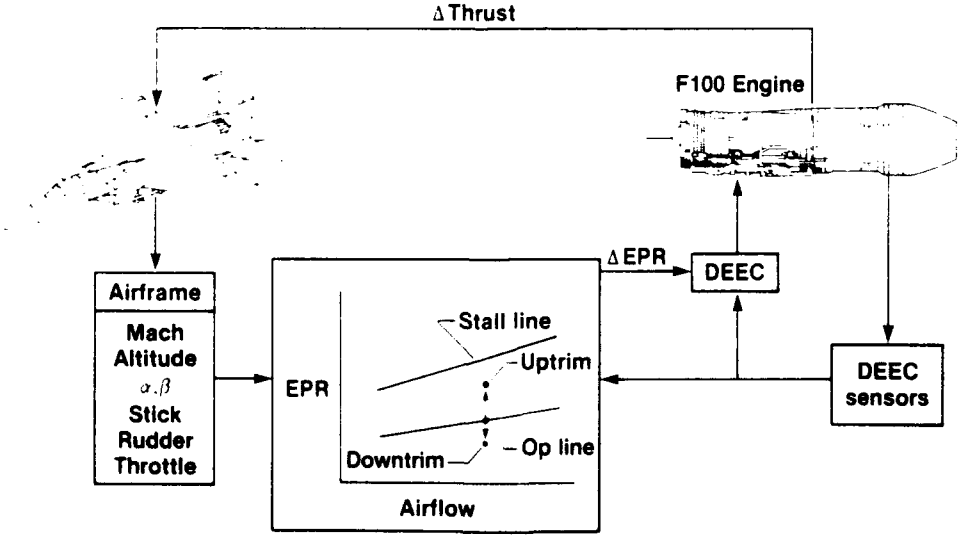


Figure 6.2 Active Engine Stall Margin Control

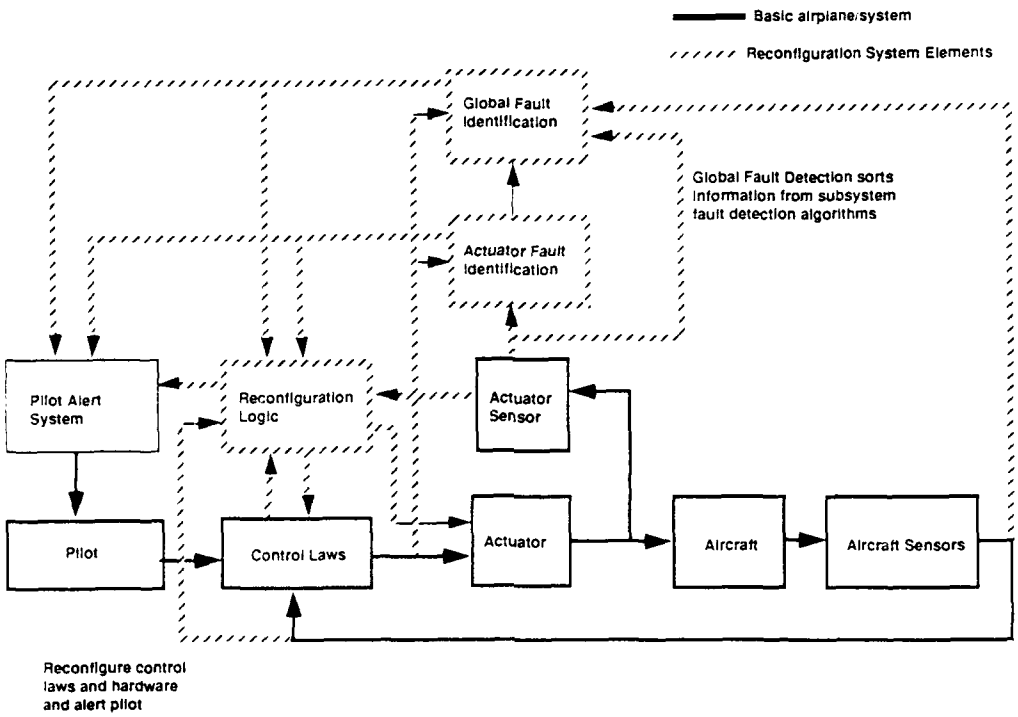
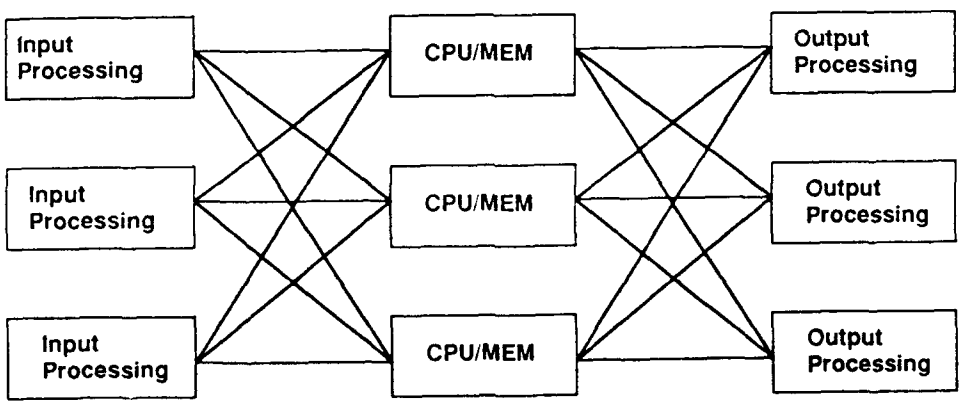


Figure 6.3 *Reconfigurable Controls Approach*



- Each channel has access to basic elements in other channels
- Following channel fail and isolated failure, unfailed elements may continue to be used

Figure 6.4 Resource Sharing in a Multi-Channel System

(a) Basic Concept

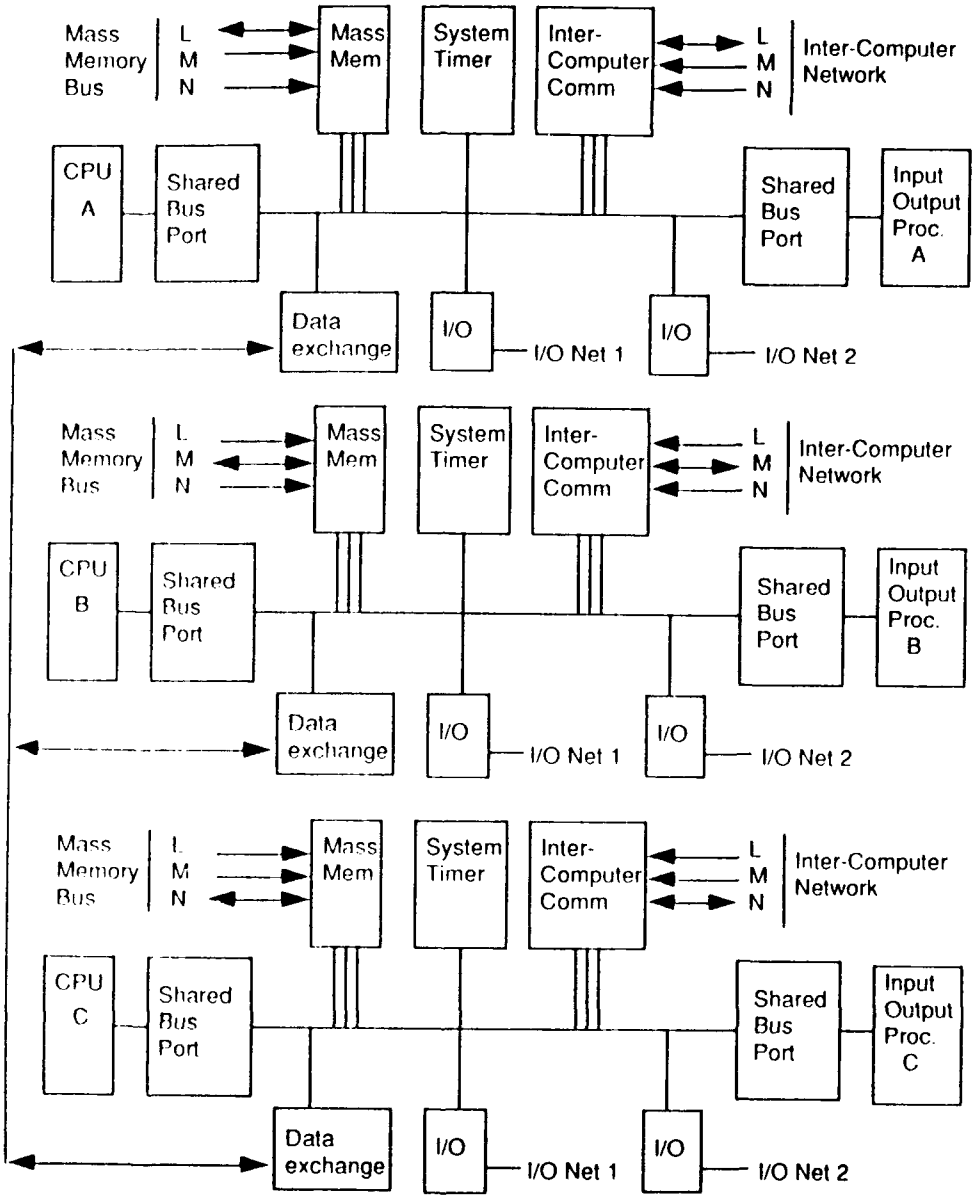


Figure 6.4 Resource Sharing in a Multi-Channel System

(b) Realistic Implementation in Fault Tolerant Processor

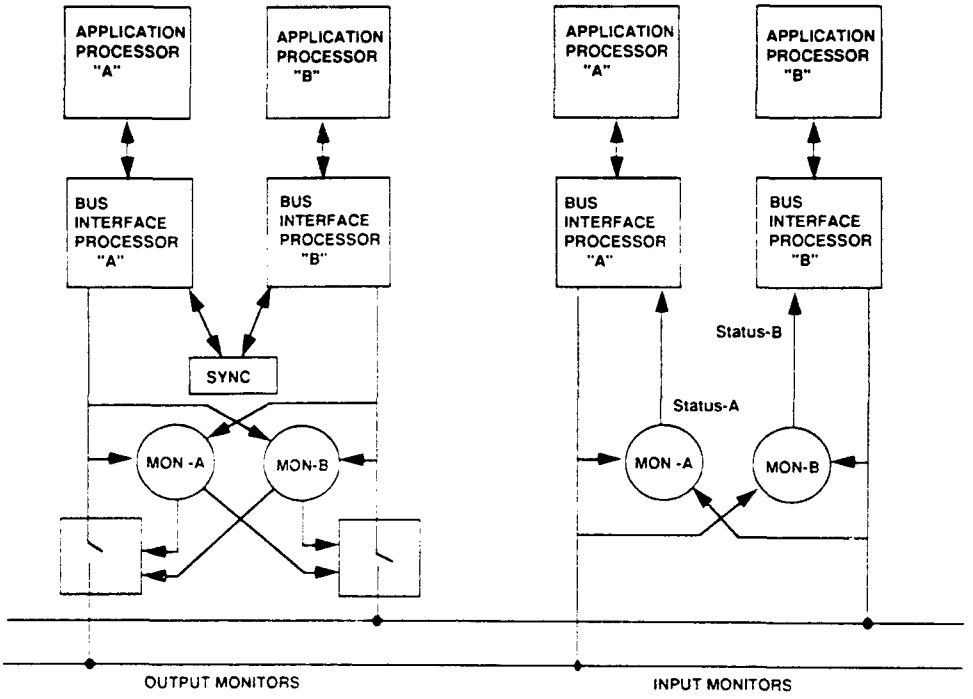


Figure 6.5 Replicated Sub-Channels

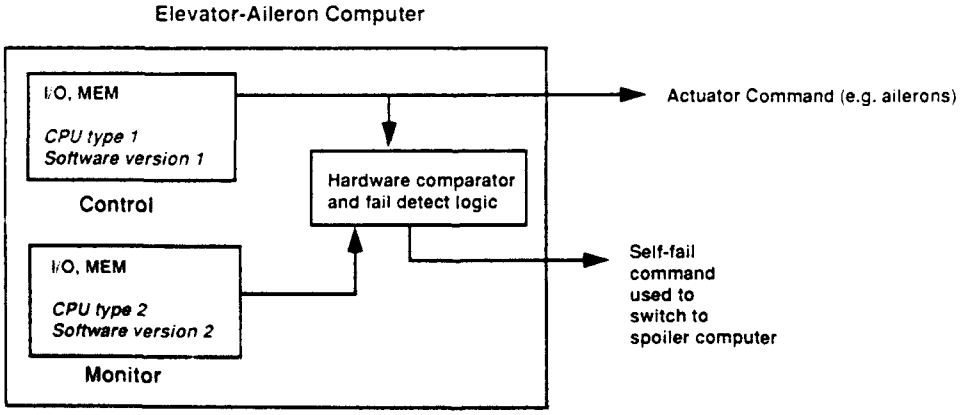


Figure 6.6 *Dissimilar Embedded Sub-Channels*

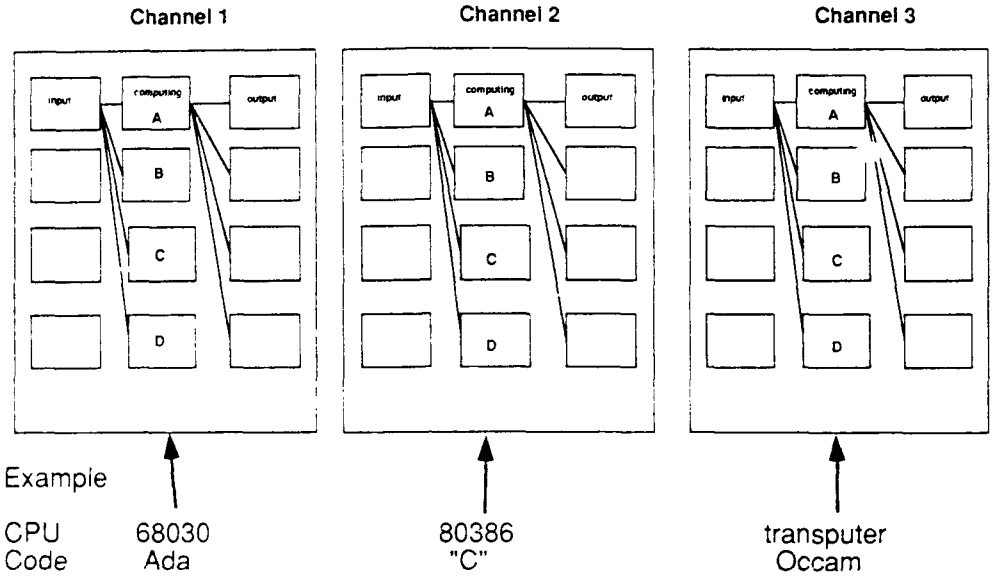


Figure 6.7 Hybrid Architecture
(not all connections shown)

CHAPTER 7

EMERGING TEST AND VALIDATION TECHNOLOGY

7.1 Introduction

Preceding sections of this document have defined a generic validation process, described the state of the art in applying that process to FCCS's and assessed future flight control system advances and their impact on validation. This chapter will describe some emerging technologies which address existing validation issues and will also indicate areas where future research activities are required. Topics will generally be presented in the order they would occur in system development. This corresponds with the sequence used in Chapter 3 and illustrated in Figure 3.2

7.2 Emerging Technologies for Specification and Design

The first activities undertaken in FCCS development are creation of system goals and requirements and development of functional and design specifications. There are a number of emerging technologies which assist in validating system development in these phases. Amongst these technologies are formal proof mechanisms and reliability assessment. The formal proof mechanisms contribute to system validation by showing that each successive step in system development can be mathematically shown to meet the intent of the specifications of the immediately prior stage. Reliability assessment, conversely, can assist in validation by showing the relationship between decisions made in one phase and the impact in a separate phase. For example, if a design decision is made to develop a quad redundant system as opposed to a triplex system with analog back-up, reliability analyses can potentially be used to assess the impact on overall system reliability for specific failure rate and coverage assumptions.

7.2.1 Formal Proof Technology

Formal proof technology involves the use of mathematical specification languages, mathematical proof, and automatic theorem provers. The basic approach is to specify the system design via a hierarchy of mathematical models. The models at the top of the hierarchy are the most abstract and represent overall system specifications. Each step down in the hierarchy contains more detail representing the results of design decisions. A mathematical proof is provided between each level to demonstrate that the "axioms" at one level can be proved as "theorems" in the level immediately below it. Automatic theorem provers are used to insure that there are no errors in the proofs. The lowest level in the hierarchy is translated into program code (or a digital circuit in the case of hardware). This approach has been used successfully in the design of operating system software[1] and in the design of hardware[2].

While progress has been made in the theory and practice of formal proof technology (e.g., program proving), much work is still needed to make this technique effective, including methods for treating numerical computations, asynchronous concurrent systems, and hardware. To be effective, this approach requires the use of hierarchical structure and formal specification. Better understanding is also needed on how to combine the methods of formal proof, simulation, and physical testing so as to achieve the greatest level of confidence in the validation process. Future systems will require the integration of independently timed and independently controlled subsystems. Faults in these systems may give rise to errors in synchronization, conflicts in control, and inconsistencies in data that are presently very difficult to analyze. New analysis techniques are needed for designs that attempt to prevent such errors.

7.2.1.1 Applications of Formal Proofs to Hardware

When a design team chooses a microprocessor as a component of a flight critical flight control system, they allow for the possibility that the chip may suffer a malfunction by specifying use of a redundant system. If, however, the system does not perform as anticipated, the processors in each redundant lane may show simultaneous anomalous behavior and defeat the redundancy scheme. Where possible, the existence of rigorous testing eliminates almost all of these situations. Nevertheless, since the number of possible states of the system may be extremely large, even the most careful testing may not detect all such behaviors, and more formal methods for designing microprocessors have been sought University research results, a formal hardware description language with the underlying rigor of first order logic, and modern VLSI CAD software have been combined to provide an initial basis for validated microprocessors. Devices have been produced by these novel methods and are undergoing evaluation[3]. If successful, they should be available for use in the next generation of flight critical control systems.

7.2.2 Semi formal Methods

In addition to the methods of formal proof technology which have not yet reached the maturity to be practically applied, semiformal methods have the advantage that their operational use is already feasible while research and development are still carried on. Common practice is the application of a basic configuration comprised from a subset of the methodology while extensions are still in development or objects of research.

One approach to semiformal methods to support the validation process is the implementation of a model of the development process which assumes that the first step is to define a set of general goals. The requirements and constraints imposed on the system will result from its interconnections to other technical or organizational parts[4]. For the system development a decompositional principle is assumed. One of the different approaches known is very informal and uses natural language to describe the details of the goals, requirements and constraints. The formal properties are limited to a fixed outline scheme, keywords, and references (lateral and top-down) for the goals, requirements and constraints.

The validation process is supported by automatic checking of traceability and change control from system specification down to the system design and further on even to software and hardware design. Additionally, manual inspections and walk throughs are supported by this method. Development is still in process to formalize this informal part and thus extend the formalization of this approach step by step.

In general, research activities concentrate mainly on abstract models for the description of the requirements, functions and system interconnections and interdependancies. Different approaches--mainly using state transition techniques and data flow description methods have been developed.[5] Problem areas addressed in research are the transition from one phase of the development process to the other in both directions to support the validation process top down and bottom up. These are mainly evoked by differences in the abstract models developed for the different phases of the development process.

7.2.3 Reliability Assessment

One of the first steps in validating the design of a flight critical system is to determine the approximate theoretical reliability of the system.[6] This analysis is performed to obtain assurance that the basic configuration chosen for the system has the inherent capability of meeting the reliability requirements for the functions to be performed. The analysis should take into consideration the number of redundant channels, the number of channels needed by the redundancy management scheme, the reliability of self-test, and other factors affecting the probability of total failure. The computed probability of failure is compared with the requirements for the system to determine the adequacy of the system configuration and associated hardware.

Reliability analysis performs one of its most valuable roles in the configuration tradeoff studies. It can become a major factor in determining the number of channels required, the methods used for redundancy management and failure detection, and in many other system design considerations. Once a configuration

is chosen, this type of analysis can be used to give reliability requirements for the component parts of the system. It is particularly valuable in determining the sensitivity of the probability of system failure to the reliability of critical parts. This process thus identifies where particular care must be exercised in the design and validation of the system.

Reliability analysis can only be done after the design is complete. Analyzing the system actually implemented, using credible estimates of component reliabilities, gives an updated prediction of the reliability actually expected for the system--thus providing baseline numbers against which test results and operational experience can be compared. This type of analyses gives a direct estimate of overall reliability rather than placing emphasis on indirect measures offered by fail-op, fail-op, fail-safe, or similar requirements.

Two interrelated factors--the high reliability required in flight critical systems and the complexity of redundancy schemes sometimes used to obtain this reliability--contribute to defining the comprehensive capabilities required of any applicable reliability assessment method. The increased levels of reliability required are moving systems further and further away from levels that can be analyzed and demonstrated with testing or piece part reliability prediction methods. Individual electronic units with failure rates in the range of 10^{-3} to 10^{-4} per hour can be analyzed by conventional methods such as FMEA's, using established experience for component failure rates. The predictions made by these methods can be rather accurate and can be confirmed by actual service experience. A typical production unit will accumulate hundreds of thousands of operating hours per year and will experience many failures, giving a statistically significant estimate of its actual reliability. However, to achieve the very high reliability required of a flight critical system, several units have to be integrated into a system which can tolerate faults and still operate. The reliability of this total system cannot be life-tested and thus must be assessed analytically, based on the system response to failures within the individual units. Since the reliability of individual units is an important input into this system analysis, conventional unit reliability analysis will continue to be important. Nevertheless, the methods used to combine this data to give the total system failure rate require additional capability and are still being developed. Specifically, the necessity to accurately represent and obtain credible reliability estimates for the complex architectures and sophisticated failure tolerance schemes in current use and projected for the future gives impetus to the development of advanced reliability estimation tools.

7.2.3.1 Reliability Assessment Tools

Several reliability assessment tools are emerging as potential aids in the analysis of complex systems. These analysis techniques use a variety of analysis methods and simulation procedures, and are normally implemented in large general-purpose computer programs. They are continually being involved to meet the demand posed by emerging systems and to reflect increasing understanding of the fault free and faulted behavior of fault-tolerant systems. Since there is a limit to the degree of modification that can be effected on a particular program, the evolutionary process tends to generate new programs. It has not been possible thus far to consolidate the reliability models under one mathematical representation and a proliferation of programs has resulted. One positive aspect of this situation is that many of the reliability assessment codes have overlapping capabilities thus affording the opportunity to perform comparative analyses[7]

Some approaches compute rigorous mathematical bounds on the probability of system failure for systems which satisfy fairly general mathematical requirements. The bounds are algebraic in form and are thus efficiently computed. These programs can be extremely fast and the upper and lower bounds are quite close (often within 5 percent of each other) for many systems.[8] These programs use means and variances of the system fault recovery times and thus permit use of experimental data without requiring sophisticated data fitting methods. One major advantage of this approach is that the effect on system reliability can be quickly determined for a large number of fault recovery times when the complete distribution is not known--thus providing a very useful capability in the early stages of system specification. The programs are reasonably general (e.g., it is not required that spares have the same failure rate as active components). Current manifestations of the program are limited to systems whose components experience an exponential distribution of times to failure--as is appropriate for computer systems. If the theory used to obtain the mathematical bounds could be generalized to apply to non exponential failure processes, then these

programs could be extended to cover mechanical as well as computer systems.

Other reliability assessment techniques calculate an explicit estimate of the system reliability[9]. Some of these programs use the large disparity between the low rates at which faults occur and the quickness of system detection, isolation, and recovery processes in simplifying the problem of obtaining reliability estimates through behavioral decomposition. The solution technique assumes separate fault-occurrence and fault-handling models. Numerical integration techniques are then used to obtain error coverage parameters. These results are then inserted into the fault-occurrence model to compute the system reliability.

Some of the more important features of these class of estimators are the ability to include fault/error handling situations such as latent and near coincident errors/faults and the capability to accurately capture redundancy management techniques. The models cover both transient and intermittent fault occurrences and exponential or non exponential times to failure. The fault tree model notation is used to permit description of a large state space.

While these theoretical reliability assessment programs are useful and have achieved wide distribution, they have specific limitations. Current limitations of these programs include the inability to model sets of failures where the order of failure is significant, and the inability to handle cold spares. Fourth generation reliability estimators are being developed to directly attack these limitations[10]

7.2.3.2 Reliability and Performance Analysis

Progress has been made in increasing the realism of reliability models of fault-tolerant systems, but future systems will require even more complex models. Research is needed that will increase the power, computational convenience, and fidelity in portraying the dynamic behavior of these systems. As a specific example, research is needed which will provide methods for coping with the future use of the dynamic allocation of resources as described in Chapter 6. Research is also needed on how best to combine available and proposed future methods for use on high-reliability fault-tolerant systems. There is also a clear need to integrate reliability estimation and performance assessment methods. This capability would enable performance reliability and life-cycle cost tradeoffs early in the design cycle.

7.3 Emerging Technologies for Implementation

7.3.1 Integrated Software Environments

In the analyses, design, and development of current systems, there is a need for many people to share information, for the airframer to cope with interactions between miscellaneous subsystems and to pay special attention to configuration control and documentation updating. This need will become even more pronounced with future systems. To help meet these needs, software designers and end users are turning increasingly to integrated software environments (ISE) that tend to meet three major goals:

- better mastery of the technical complexity
- optimum management of systems development and maintenance process (multiple subsystems made by multiple partners, sometimes in multiple versions along the life depending on the customer)
- easier sharing of information of any kind

The last advantage, not the least, comes from standardization (which is a natural consequence of ISE): the cost of ISE development can be spread among multiple users to reduce the initial financial impact.

ISE provides its user a homogeneous set of tools. It is integrated in the sense that all tools offer a uniform (at least similar) man-machine interface for input and output of data and a uniform (at least compatible) underlying data implementation, thus the tools can communicate together in a way transparent to the user. Some of these tools are just utilities for cross-checking communications, while the others tend to cover all the life cycle. Specific tools are:

- computer-aided formal specification

- functional test pattern generator
- simulator of formal specs
- computer-aided hierarchical design
- multi-language automatic code generator
- code complexity analyzer
- test coverage analyzer
- data base with linked statistic trying to evaluate reliability

These ISE's operation relies on varying programming languages starting from functional descriptive language (FDL) or equivalent graphics down to programming target language (PTL), via some program design language (PDL).

Because of advances in microelectronics and compiler performances, it seems definite that PTL and PDL can advantageously be merged in some high-order language (HOL) like Ada.

Some FDL's describe the behavior in terms of states of the system and transitions between states. Specified conditions enable the transitions and resulting actions. A promising research topic is the use of a more general formalism able to grasp fine problems in parallel processes such as Petri net. Specification tools based on Automation or Petri net formalisms are now close to being mature enough for production use. Note that accuracy of the subsequent analyses and validation activities using an FDL description is, of course, a function of the rigor with which the transition to formal model is made (simulator of FDL descriptions can help showing how close FDL constructs are to the required system behavior).

FDL also has features to describe the interface between a program procedure or module and its environment (such as data import/export dependency lists). Assertions (such as pre- and post-conditions) can be embedded in an FDL model for the automatic comparison of a program with its specification. These assertions enable subsequent formal proof activities. An FDL model of a program can be constructed automatically, using a translator, or by hand. Manual translation has proved acceptable in small projects where the cost of developing a special-purpose translator is not justified.

The translator from source language to FDL checks the integrity of the source code in a number of ways. In addition to checking the syntax and static semantics of a program or subprogram in a similar way to a conventional compiler, it ensures that the programmer has not used language constructs which can give rise to ambiguities or uncertainties (such as variant records in Pascal for example). The translator also performs a number of other tests, for instance to confirm that data transfers between subprograms and their environments are consistent with their import/export specifications.

Analysis of Program Flow

Once the FDL representation of the program is obtained, a number of flow analyzers can be used to check that a program is well formed in its control structure, data usage, and information flow. A representation of this analysis capability is shown in Figure 7.1. If a program is found to be defective in any of these respects, the errors or anomalies are reported.

The control-flow analyzer reports:

- "unreachable" or "dead" code,
- code from which no exits are accessible, and
- "multiple-entry loops" (whose validation and testing are intractable) and other defects in control structures.

The data-flow analyzer reports:

- use of undefined variables,
- unused variable definitions,
- loop-invariant definitions, and
- redundant tests.

It also tabulates all possible usages of each data definition in a program, and for each data usage it indicates where that data item may have been formed.

The information-flow analyzer reports:

- ineffective imported data,
- ineffective code,
- use of undefined variables in constructing exported data,
- loop stability (a form of non termination), *and*
- inconsistencies between prescribed import-export relationships and program text.

Its tabulations include a matrix showing which statements affect each exported data item and a matrix of import-export dependencies.

Program Validation

After checking the syntax and static semantics of a program procedure or module (using an appropriate translator and reader) and checking the integrity of its control, data, and information flow, a determination must be made that the program performs its required function. Tools can be provided which assist in this task. These tools can construct *"path functions" (which describe the conditions* under which particular paths through a program are executed, and the consequent transformations of variables) for manual inspection and comparison with a specification. Figure 7. 2 illustrates the process.

If a specification is provided (in the form of pre- and post-conditions and loop-invariants) the corresponding evaluation conditions can be generated directly. If these conditions are satisfied, then the program meets its specification.

Also available in some integrated development environments are tools for standardization of arithmetic and logical expressions, application of replacement and inference rules via pattern matching with a database of rules, and limited automatic deduction to fill in those details of a proof-step not provided directly by the *user.*

7.3.2 Software Fault Tolerance

Since software does not wear out or develop a fault as it is operated, it does not fail in the same manner as hardware. Software faults are present in the software from the time it was written. (Changing hardware can cause software to become "wrong". This situation is *really a total system design error but is often* labeled a software error. A particular software fault may be latent for some time since the specific circumstance that causes the execution of the faulted software to execute may not occur until the system has been in service for a number of years.

Software errors are similar to design faults in hardware and the best way to eliminate them from operational flight programs is to prevent their occurrence in the first place. Since this is not entirely possible, comprehensive validation exercises for software are performed. Several studies have been conducted to aid in the validation process by attempting to establish models which relate the time history of errors found in the debugging activity. In theory, as well as in several of these studies, the time to find each successive error is significantly increased--thus leading to the hope that a model can be constructed which will permit a decision to stop software *debugging a particular program based on the history of* software errors in that program. It will then be expected that the remaining errors cannot occur before a period of time consistent with the safety or availability objectives.

These "software reliability growth models have provided impressive results for large non-essential ground software where the number of errors are high enough to allow a reliable estimate; however their practical application to FCS is a long way off, due to the non-significant number of errors discovered during testing. Despite these procedures, it cannot be assumed that the reliability of software after testing and validation is 1.0--a situation which is manifest by the number and variety of software bugs subsequently found in systems which have been subjected to exhaustive testing. A solution to this problem in flight critical systems has been sought by providing a number of redundant channels (using dissimilar software) or provid-

ing a back-up software channel. The most widely known methods for providing software fault tolerance are known as recovery block and N-version programming methods. They are conceptually similar to hardware techniques called standby sparing and N-modular redundancy, respectively. N-version programming requires that multiple versions of software be developed by different programmers to a common set of requirements. In the application environment, these multiple versions are operationally subjected to a majority voter and the system would give an incorrect output only when the majority of its N-versions fail. Since the multi-version technique is a closely analogous to hardware redundancy for which statistically independent failures are assumed, there is a natural inclination to use the same model for software N-version redundancy. Recent experience suggests that this assumption may not be valid.[11]

Studies of the effects of multiple joint occurrences of errors on the probability of failure of N-version systems have been conducted to determine the effects of errors which are not statistically independent. In particular, it has been found that some well known implications of redundancy of hardware models do not extend to the type of model needed for redundant software. An approach to determine the effects of coincident error on overall reliability gains has been developed and a sufficient condition found under which an N-version software system is a better strategy for reducing the probability of system failure than relying on a single version of software.[12]

7.3.3 Automatic Code Generation

The automatic generation of code is a very promising and cost-effective technique that offers several advantages:

- uniform coding style and constructs (that ease readability, thus maintenance)
- use of generic low-level code modules (packages or simply libraries) that lead to reuse of software and thus enable some validation efforts to be performed only once
- less effort for validation since the unit-test step can be suppressed (at least greatly reduced)

The last point is of greatest importance with respect to this report. It means that one can take some credit from using an automatic code generator and maybe alleviate some module testing, provided the tool has been validated. The tool is then assumed to produce error-free (at least highly reliable) modules; there is still a need for integration testing and functional validation of the software within the real hardware. It should be noted however that validating a code generator can reveal much more complex than validating FCS software.

Given an arbitrary specification, automatic code generation is clearly beyond the state of the art. Nevertheless, there are subclasses of problems where this technology is feasible. In particular, programs which can generate control software from block diagrams are within the state of the art. In fact, several companies have independently developed such programs[13]

7.4 Test and Evaluation

7.4.1 Automated Testing

Automated software testing offers the opportunity to perform relatively exhaustive testing of the software while at the same time reducing the manpower required to effect the tests. To accomplish automated tests, a command system exercises the control system software through a preplanned set of conditions. High-order language models of the flight control system can be used as a basis for validating. Significant benefits can be obtained in automated testing of operational flight code if the HOL simulation used to generate the expected result is developed in a different language from that used for the flight code. The test data are automatically compared with the expected results on a point-by-point basis. If the actual and expected results differ by more than a specified tolerance, the data are then sent to a printer or plotter where they are available for visual inspection. The tests can be comprised of static checks, logic checks, and real-time checks. As noted elsewhere in this report, exhaustive automated testing offers the best answer to the question of how much software revalidation is required following code changes since the complete revalidation, which is efficiently performed using automated testing, obviates the need for decisions on how much validation is required. Still at issue is the question of how a set of tests can be identified which

assures complete validation of the software.

Experimental testing, for example, injection of faults in simulation models of fault-tolerant systems (both at the hardware and software levels) is in current practice. The method is useful for gaining insight, but it is costly, and it is difficult to assess the completeness of a set of experiments. Better theoretical understanding is needed to guide the construction of efficient test sets and to increase the power of the inferences about fault-tolerant performance that may be drawn from test results.

7.4.2 Integrated Test Environments

With the advent of increasingly complex aircraft systems, there is a need for more integrated ground test facilities to support validation test activities for new research and development aircraft. The motivation for these facilities is that a primary means for validation is the exercise of aircraft systems in as realistic a context as possible. From a systems viewpoint, this can be stated as the notion that integrated systems need integrated testing. In at least one implementation of this concept, the development aircraft is itself made a part of the test facility. The aircraft is connected to simulation models with the capability to feedback control surfaces, pilot inputs, and system commands. In this context, the aircraft becomes its own "iron bird." Overall this approach permits:[14]

- test and diagnosis of an aircraft system with all other interacting systems operating
- simultaneous structural dynamics and control systems tests
- central control of multiple redundant aircraft electrical and hydraulic supplies

While this concept has not yet been fully implemented and evaluated, it appears very promising and may preclude the need for iron birds.

7.4.3. Flight Testing

Flight test methods have shown significant improvements. In particular, significant flight test results such as stability margin determination and vehicle response validation can now be obtained in near real time. As an example, validation of the actual stability margins of the X-29 was accomplished in flight, during envelope expansion. This approach was selected as a result of the highly unstable aircraft, minimum gain/phase margin design, and uncertainty levels in the aerodynamics. This test cannot be accomplished on the ground because actual aerodynamics are not available. Stability margins were immediately overlaid on predictions for a final validation test that the control system design met stability specifications. This method is an excellent validation test which establishes the stability margin specification, the adequacy of the small perturbation aerodynamic model, the end-to-end performance of the flight control system and actuation system hardware.

In addition to the stability margins this on-line test capability has been used to overlay small perturbation time history responses on predictions of the response driven by the identical control input. This test validates the linear aerodynamic model, provides a measure of the nonlinearities in the vehicle-system and provides a high confidence validation of the design process.

7.5 General Research in Validation Methods

In Chapter 6, which discussed future trends in flight critical flight control systems, and in the earlier sections in this chapter, several areas of required research in validation methods have been mentioned. This section will not repeat those items but will add several topics of a general nature.

7.5.1 Understanding the Validation Process[15]

Models or paradigms are needed that will give comprehensive and consistent descriptions of the validation process. These models should allow clear expression of validation goals, techniques, and procedures and provide clear definition of general concepts, such as confidence, reliability, and validation. Particular research goals are:

- Models for incremental validation processes that proceed through successive levels of the system life cycle, such as statements of user needs, statements of requirements, specifications, design, integration, implementation, operation, maintenance, and modification. As a special case, models are needed for describing the validation of major changes in a system, such as redesign, or integration into new system environments.

- Clarification of the roles of formal and informal activities within the validation process.

A numerical measure of validation credibility would be very desirable and may be a legitimate goal of validation methodology. Such an overall measure would aggregate the combined impact of efforts both to prevent errors and failures from occurring and developing systems which tolerate those errors and failures which do occur. Such an overall measure has been and will continue to be elusive. In the meantime, a number of technologies are emerging which support the continuing improvement of the ability to validate flight critical control systems. In many cases emerging test and validation techniques do not replace existing techniques. They arise in order to handle the increased complexity of flight critical digital systems and are used along with the methods described earlier in this document.

7.5.2 Specification and Design

In order to achieve a high level of trust in the validation of complex fault-tolerant systems, it is necessary to have precise and understandable specifications for the functions that a given system is intended to perform. These specifications should have a high degree of mathematical rigor in order to allow formal demonstration of consistency between specifications and corresponding designs.

An important design approach is the use of fault tolerance in software and at the system level. Methods are needed to evaluate the effectiveness of this approach for structuring fault-tolerant systems.

7.5.3 Validation of Knowledge-Based Systems

As system complexity increases, a significant amount of flight code is devoted to logic and other decision-making tasks, such as pilot-decision aids, which fall in the domain of knowledge-based systems. These systems provide new challenges for validation. In particular, it is very desirable to find approaches for validating these systems which allow them to fully use their capabilities rather than to restrict application of much of their capability because appropriate validation methods are not available. In this case, as is generally true, increased efforts are required to insure that validation methods are considered concurrently with technology development rather than trailing them.

7.5.4 Design and Evaluation

The study of systems, particularly architectures, involves the determination of many characteristics of the system, some of which will be difficult to quantify. There is a growing body knowledge about flight control systems which could be categorized to form the base for an expert system to aid the design of safety critical systems. However, such techniques are still in their infancy and will require research before they can be developed into useful aids. At this time there is no alternative to the experienced designer backed by analyses of the system characteristics. In addition to methodologies for analysis, it would be beneficial to have available powerful design aids that can structure systems so as to make them easier to validate. Validation practice should have sufficient impact on design practice to assure its own feasibility. Some research on how to build design tools that enhance validation is therefore justified.

7.5.5 Data Base

Documented data from existing systems is needed for establishing the history of faults experienced, as well as the applicability and ease of use for specific validation methods.

7.5.5.1 Fault Experience[16]

Data are needed on the frequency of occurrence of all fault types, including design faults in hardware and software, physical faults, transient or intermittent faults and operational faults. Data are needed about different fault forms, such as intermittent and correlated multiple faults, and about the effects of difference environments. New means are needed to collect such data, for example, by incorporating fault monitors and recorders into operational equipment.

The value of a collection of tools for analysis and design would be greatly enhanced if they reflected a unified methodology for systems development. In the absence of complete control of the design process, tools may be needed to support a variety of design approaches. The goal of a unified methodology is nevertheless an important subject of research. Given such a methodology, the tools could comprise a powerful environment for the development of integrated avionic systems.

7.5.5.2 Validation Experience

Data are needed about the cost and effectiveness of current validation techniques, both established and experimental. These data will be important to the effective planning of large validation exercises.

Case studies (e.g., of simplified flight control systems) could be conducted to help compare competitive validation approaches, and to provide rapid evaluation for proposed modifications to an evolving approach. The approaches might include the full range of techniques for various architectural approaches. The studies could also help to evaluate the reliability enhancement of hierarchical and fault-tolerant hardware and software structures and techniques for their validation.

REFERENCES

[1] Moser, Louise; Melliar-Smith, Michael; and Schwartz, Richard: Design Verification of SIFT, NASA CR 4097, Sept. 1987

[2] Milne, G.; Subrahmanyam,P.A.: Formal Aspects of VLSI Design, North Holland, 1986

[3] Cullyer, W. J.: Hardware Integrity. Aeronautical Journal, Aug./Sept. 1985

[4] Lauber, R.J.: Development Support Systems. IEEE Computer Transactions, 1982

[5] Epple, W.K.: Rechnergestutze Spezifikation von Processautomatisierungs-Systemen (Computer Aided Specification of Real Time Automation Systems) Regelungstechnik Praxis RTP, Vol. 3 and 4 , 1984

[6] Szalai, K., et al.: Digital Fly By Wire Validation Experience. NASA TM-72860, 1978

[7] Bavuso, S.J. and Martinsen, Anna, L. : A Fourth Generation Reliability Predictor. 1988 Annual Reliability and Maintainability Symposium, 1988

[8] Butler, R.W.; and White, Allan, L.:SURE Reliability Analysis--Program and Mathematics. NASA TP 2764, March, 1988

[9] Bavuso, S.J.; A Users View of CARE III. 1984 Annual Reliability and Maintainability Symposium, San Francisco, C.A. January 1984

[10] Bavuso, S.J. and Martinsen, Anna, L. : A Fourth Generation Reliability Predictor. 1988 Annual Reliability and Maintainability Symposium, 1988

[11] Nagel, P.M.; and Skrivan, J. A.: Software Reliability: Repetitive Run Experimentation and Modelling. NASA CR-165036, 1982.

[12] Eckhardt, D. E.; and Lee, L. D.: A Theoretical Basis for the Analysis of Multiversion Software Subject to Coincident Errors. IEEE Transactions on Software Engineering. vol SE-11, no. 12, Dec. 1985

[13] Harschburger, H.E.; Glader,B.; and Hammel, J.R.: Backup Modes for the F/A-18 Digital Flight Control System. 6th Digital Avionics Systems Conference, 1984.

[14] Mackall, D.A.; Pickett, M.D.; Schilling, L.J.; and Wagner, C.A.: The Integrated Test Facility and Its Impact on Flight Research. NASA TM 1000418, May 1988

[15] Validation Methods for Fault Tolerant Avionics and Control Systems. NASA CP-2114. Working Group Meeting, March 12-14, 1978, LaRC, Hampton, VA.

[16] Meissner, C. W.; Dunham, J. R.; and Crim, G. eds: NASA-LaRC Flight-Critical Digital Systems Technology Wolrkshop. NASA CP- February 1989.

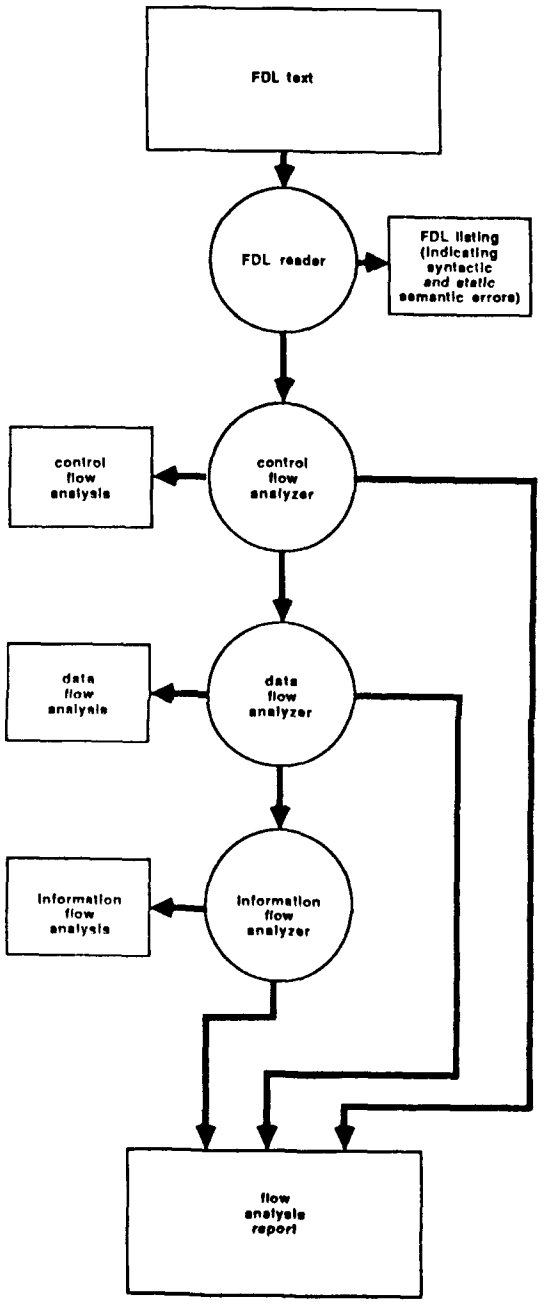


Figure 7.1 Control Flow Analyzer

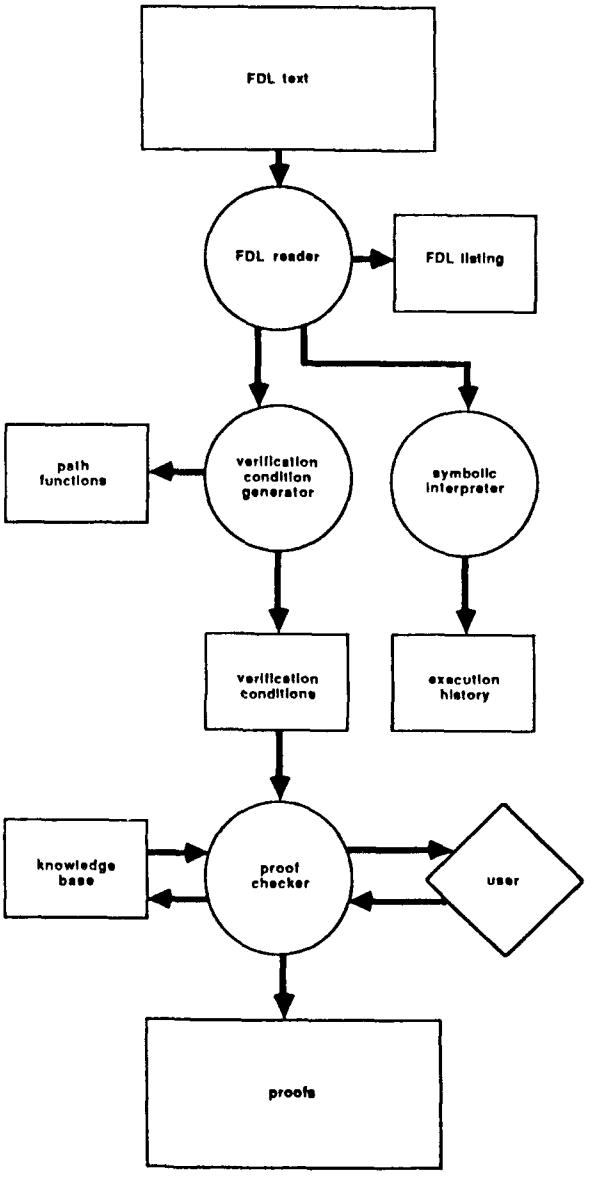


Figure 7.2 Program Verification Process

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

Working Group 09 of the AGARD Guidance and Control Panel has studied the Validation of of Flight Critical Control Systems FCCS. This effort was undertaken both to provide guidance to those concerned in the validation of FCCS by reviewing and summarizing the validation process and methods and to identify areas of research which the AGARD nations need to explore to enable validation of FCCS envisioned for the future. The following are conclusions and recommendations from the WG deliberations:

Conclusions:

1. Several FBW FCS for military and civil aircraft have been developed and flown successfully to the same safety requirements as aircraft with mechanical FCS.

2. Validation is intrinsic to the development process.

3. There is no universally accepted plan for the FCCS development process.

4. The projected increase in FCCS functionality will require significantly improved validation effectiveness.

5. A framework exists to link top level safety requirements to the FCCS design but quantitative models do not exist for all elements.

Recommendations:

1. Accelerate development of automated validation methods to predict and improve test coverage, to produce test stimuli, and to evaluate results.

2. More attention must be given to those validation activities that are required for the specification and design phases of FCCS development.

3. Research must be accelerated and expanded to develop new methods and tools for the validation of future integrated systems and aerospace vehicles.

4. Designers should trade-off unique FCCS equipment and software with standard or widely used components to gain the validation experience of those systems.

5. AGARD should continue to include advances in validation research and technology in symposia, working groups, and lecture series.

CHAPTER 9

EXECUTIVE SUMMARY

Introduction

This executive summary provides an overview of th formal report of Working Group 09 of the AGARD Guidance and Control Panel. The sections of the summary correspond to the chapters of the report for ease of reference to the material beign summarized. This executive summary may prove useful to those individuals who might not have time to read the entire report or who may wish to preview the report prior to reading it in its entirety.

The Terms of Reference for Working Group 09 were approved by the National Delegates Board of AGARD and the objectives of the working group were:

(1) To provide guidance to those concerned in the Flight Critical Control System (FCCS) validation, namely system designers and certification authorities.

(2) To identify the areas of research which need to be explored to enable validation of the next generation of FCCS.

Item (1) is addressed in Chapters 2 through 5. The report is partitioned to highlight the validation process, but it is not possible to separate validation from the design and development process itself. Therefore, the report first describes the state of the art in flight critical flight control systems in Chapter 2 to provide a common basis for understanding the validation requirements. This is done by way of four examples which illustrate the range of design variables in modern flight control systems. It is these types of systems which provide the structure for the assessment of the state of the art in validation. In Chapter 3, a top level generic development and validation sequence is described, as well as a description of the interrelationship of the vehicle development, systems development, and validation process. This serves as both a summary of the process and also as a guide to the detailed description of the processes contained in Chapter 4. In Chapter 5, a critical assessment is provided of the principal elements of the validation process.

Item (2) is addressed in Chapters 6 and 7. Chapter 6 contains the projection of trends in flight control design over the next 15 years, along with the expected impact on the validation process. Chapter 7 contains the emerging tools and techniques that will be needed to improve the validation process for the current generation of flight control systems as well as for those projected in Chapter 6.

Chapter 8 contains conclusions and recommendations which are intended to provide guidance for future flight control design, development and validation technology developments.

State-of- the -Art in Flight Critical Control Systems

Functional requirements and the continuous evolution of FCCS automatic control system architectures are provided in Chapter 2 . Limited authority, analog, stability augmentation systems were developed during the 1950's; an example is the F-104. These were followed by the development of flight critical, analog Fly-By-Wire (FBW) systems, which began during the early 1970's; examples are the F-16 and Mirage 2000. The development of digital FBW (DFBW) systems started in the early '70's, and is still evolving. Examples are: NASA F-8 DFBW, the Jaguar DFBW and the F-16C/D. The trend is clearly established towards systems which are increasingly complex and which include more flight critical functions.

Architectural complexity is increasing due to the increased functional criticality and the resulting need for satisfying stringent reliability, availability and fault tolerance requirements. Moreover, the flight critical control functions which DFBW systems are asked to provide, typically require frequent inputs to the control effecters, which cannot be effectively and consistently provided by the pilot, during some, or all flight regimes and conditions. Additional increased complexity results from the requirement of

integrating many existing and new functions for improving performance, for extending the flight envelope, and for decreasing pilot workload. Examples of functions which are being considered for integration include flight control, propulsion control, weapons control, guidance, navigation, flight management, thermal management, etc.

The design of many high performance aircraft rely on augmentation systems for providing some of the safety margins traditionally provided by inherent aerodynamic stability and the structural strength and stiffness of the basic airframe. During the design cycle of the aircraft, the availability of ACT is taken into account to relax the constraints in the aerodynamics, structures and propulsion systems, while achieving the same effective margins with the active system. The boundaries of flight critical control functions have also grown beyond classical control systems, especially in the case of military applications. Flight control functions and avionics sensory functions are integrated in common architectures to satisfy the mission requirements of advanced military aircraft.

Critical components of DFBW flight control systems include the primary sensors, the digital processors, the data distribution system, and the actuation systems of the primary control surfaces. The reliability and fault tolerance requirements of a Flight Control System (FCS) configuration can be met by using several levels of redundancy more efficiently than by applying the same level of redundancy throughout the configuration. Several examples of FCS configurations which use different levels of redundancy are discussed Chapter 2.

A large number of architectural options, which have been designed to satisfy the same or similar reliability and fault tolerance requirements, are available for use by system designers. Examples of redundant flight control computer configurations which have been designed, validated, and flight demonstrated in recent years are described in Chapter 2. Included are the General Dynamics f-16 C/D, the McDonnell Douglas F-18, the Grumman X-29, the FBW Jaguar, and the Airbus A-320.

The choice of flight control system design has a number of implications on the validation process, and these include: architectural issues, software isssues, and sensor/actuator issues. Many such considerations determine the final configuration of the control systems of an advanced aircraft. The objective is often to compromise among many different, and some time conflicting requirements. In the case of actuation systems, a major consideration is the selection of: a) a configuration which utilizes a dedicated control system for control, failure detection and isolation, and reconfiguration logic, or b) a configuration which utilizes the central flight control system, for those purposes.

SOA Generic Development and Validation Process

The validation process is embedded in a complex series of events making up the development of the flight critical control system (FCCS), which is only part of the flight system and total airplane development process. A well organized and systematic airplane and flight system development process is a necessary foundation for a successful and efficient validation program.

The purpose of Chapter 3 is to provide a top level description of the FCCS validation process and its relationship to the overall airplane and flight systems development cycle. It serves as a guide and background to Chapter 4, which contains a very detailed description of the state-of-the-art tools, techniques, methods, and approaches used in the validation of the FCCS.

There are many ways in which systems can be developed and validated and these ways change with time. The method described in this chapter is a generic process, based on the experience of the members of the working group who have been associated with the development and validation of mostof the flight critical control systems produced in Europe and USA
during the past two decades.

The elements of the generic validation processof an FCCS include, the life cycle, the goals and requrements, the funtional specification, the design specification, the implementation and prototype, the prototye aircraft, and the production system, and these are described in detail in Chapter 3.

The top level requirements for the flight control system of a military aircraft are derived from the system requirements associated with the aircraft/weapon system mission and operating requirements. The relationship between the development of the airplane and the flight control system is described in Appendix 3.1.

A Life Cycle Model of a Military Aircraft

Appendix 3.1 gives a description of a generic life cycle model of a typical major military system, like a new aircraft. The model is consistent with the guidelines included in the System Engineering Management Guide which was published for the U.S. Defense Systems Management College. The guide describes an integrated system engineering and management approach, including methodology and tools, for defining the requirements, configuring and sizing the system, managing its development and verifying the capabilities of the design. It covers the acquisition and development process of any major military system from inception to operational deployment and use. The guide is consistent with all existing U.S. military standards. For the purpose of this document, the system is intended to be the entire aircraft. The FCS is a prime or critical item.

The life cycle of major DOD systems includes five phases: Mission Need Determination, Concept Exploration Phase, Demonstration and Validation Phase, Full Scale Development Phase, Production and Deployment Phase, and these are described.

Current Methodologies and Techniques

The purpose of Chapter 4 is to present current methodologies and techniques used to validate a flight critical system. Assessments of individual validation methodologies are also included. In Chapter 5, an assessment is made of the overall state of the art of validation of flight critical flight control systems.

Validation begins in the system requirements analysis phase and continues through the development phase and culminates in the demonstration that the final system complies with the system-level requirements defined prior to start of development. A flight control system development cycle based on European and U.S. practice is illustrated in Chapter 4.

There are four basic types of validation activities:

Inspection - used for determining if the product (software, hardware, or integrated software/hardware) as built, conforms to the applicable documentation, such as engineering drawings, flow diagrams, computer listings, user requirements, system specifications, etc. Inspection typically involves visual/physical examination or simple measurements.

Test - used to establish that the product functional characteristics conform to operational and technical requirements. The process has a high technical content, and usually requires specialized test equipment and formalized procedures. The product under test is stimulated with inputs to generate controlled responses which can be compared with predictions. Data generated by test is further analyzed to determine conformance with the criteria. Test passage can either be go-no-go, or be a result which falls within criteria boundaries.

Demonstration - used to show the customer and/or legal authority that the product functions as required within the operational envelope. Test passage is usually based on go-no-go criteria established by the reviewing authority.

Analysis - used to show compliance with requirements, either to complement test, or to replace test when test is not possible or practical. Data output is from simulation and analytic models. An important, but often difficult task is the validation of the models used in the analysis.

Assessment of the SOA Validation Process

The scope of Chapter 5 is to cast a critical view on the present situation of the validation issues associated with FCCS development, and follows the organization of Chapter 4, which described the SOA validation activities, methods and tools.

Validation is regarded here as an integral part of the system development process. Since the major objective of the WG was the safety aspect, two major achievements must come out of the validation process. The first milestone is the achievement of the first flight clearance., and the second one is to demonstrate the safe operation of the system within the full performance envelope. First flight clearance must without

doubt be considered as the most critical point in the development process of any flight control system. The majority of methods and tools needed for validation, with the exception of flight test instrumentation, are driven by first flight clearance requirements. Therefore, a critical assessment should focus its attention primarily to this phase.

In this Chapter it was found helpful to limit the critical assessment to the case when a complete new system, i.e. an aircraft, has to be cleared for first flight. The majority of activities needed to clear a flight critical system after a redesign or after a major modification can be regarded as a sub-set of first flight clearance work.

As for any other system development, the criteria applied to an assessment of the validation process for flight critical systems must be grouped into three categories:

o Technical (achievements)
o Economical Aspects (resources)
o Time (planning)

In Chapter 5, it was pointed out that the details of a System Development Program Plan should aim at a relocation of this critical phase and the time scale problem. The principal solution to this is a stronger emphasis on validation activities in early stages.

Basically, three possible ways to achieve this goal have been outlined during the WG discussions:

o In order to ensure completeness of the design specification with regard to subsystem/system validation, traceability and dependencies, the following position was identified. Start with final specification for previous program. Utilization of a program design language will provide automated bookkeeping and identification of dependencies. Modern relational data bases can provide powerful tools for tracking complex systems.

o One must insure that an adequate validation process exists for innovative or unconventional designs as well as for conventional designs. For unconventional or new designs approaches, a sensitivity analysis should be conducted during the early concept phase. It should identify the areas requiring special emphasis in the validation process in particular with respect to specifications where experience is lacking. In most such cases, early prototyping has to adopted. This can either be done in the laboratory environment or it may even require investigation in a non-critical manner in a flight research program.

Taking validation activities from the critical phase before first flight to the specification phases of the development plan can only be accomplished by a redistribution of the budgetary spending. It is an attractive engineering concept to go for an early prototyping but it is an important management task to prove that spending money in this way pays off after about 3 years by a more cost-effective pre-first-flight phase.

Chapter 5 also points out that the 'safety critical' factor for FCCS's may improperly overshadow other issues. Validation is an exercise which has to be performed for any system development, and the generalized aim is to create evidence that a function performs as it should. Much money has been invested in methods and tools for the development of mission avionics for military aircraft, especially for software development. The reason for this is based on the vast amount of software in avionics systems compared to the smaller scale of FCCS's.

Modern flight control systems exhibit a high level of functional integration with other aircraft systems. As a consequence, an increasing amount of safety risk is associated with the interface between the FCCS and other aircraft systems. The use of standardized methods and tools across all aircraft systems would improve the capability of controlling the increasingly critical interfaces.

The development of safety critical systems should make use of general methods and tools for the development of complex systems. Only when the need has been proven by a thorough analysis should these tools be modified or substituted by specialized (to the defined needs of safety critical system development) tools/methods. The benefits of this approach are obvious:

- Shared development costs for the tools
- Tool validation through a wide application

A wide range of validation elements as they have been used in various development processes. For each system, it was proved through a safe first flight (and prototype flying) that the validation process had been performed successfully. As a consequence, it can be assumed that there are a variety of validation elements/techniques and methods available. These can be used to set up a validation process for any new system which would be close to state of the art.

It must be noted that any particular process or technique may not be valid for a specific project.

A typical example was the debate within WG 09 concerning the need for an iron-bird, one can find cases where an iron bird is not required. To come to a valid conclusion, one has to consider a wide range of specific program aspects such as:

- Analysis of combined test plans and test facility requirements for FCS, hydraulic system, landing gear, etc.
- Availability of aircraft (prototypes) before first flight for system integration (as an alternative)

Piloted flight simulators are a powerful tool for designers studying the dynamic responses of pilots to new aircraft, systems and operating environments, and also for training pilots to manage their complex cockpit environment and a wide range of potential emergency conditions arising from failures, adverse operational conditions. For many years such simulations have been used successfully to predict and assess solutions to problems arising during the development of new aircraft and systems. However, with the exception of a few very specialized aircraft such as the U.S. Space Shuttle, all aircraft acceptance and certification has required flight test demonstration. This situation is changing as the complexity of safety critical systems increases and presents the acceptance authority with a very large set of potential failure modes. Also improvements in the standard of flight simulation are increasing confidence in their ability to represent many flight situations well enough to limit the range of conditions that require flight testing. By holding flight clearance testing within reasonable bounds there are significant savings in cost and time, which are beneficial to both manufacturer and customer.

There already have been examples where piloted simulation has been used to 'demonstrate' to acceptance authorities a range of failure modes of, for example, a muti-channel 'fly-by-wire' flight control system. From these demonstrations, the authorities selected for flight demonstration, those situations that appeared to be most demanding and most probable. Economic and practical time limitations are going to increase the range of situations where piloted simulation will be used as a direct part of the acceptance (certification) process for both military and civil aircraft/systems.

For validation or certification purposes, it is not acceptable to alter aircraft or system models to compensate for cueing deficiencies. Although this approach can improve the apparent validity of the simulation to the pilot in those areas he has experienced in flight , there can be little confidence that the simulator is presenting adequately those vital situations which are not going to be demonstrated in flight, or which dramatically affect system operation. The following issues should be addressed by the combined FCS/simulation/pilot team:

a) For what sets of conditions had validity been confirmed from flight test?

b) What confidence can be placed in simulation of conditions outside the validated range?

The art of simulation compensation is very much dependent on a wide range of factors, including physical characteristics of available cueing systems (say visuals, motion, sound, control loading, g-seats, etc.), computing systems (architecture, speed, capacity, etc), aircraft category, specific operational task, and many other factors. Thus acceptance authorities and manufacturers have to establish confidence in simulation results in either relevant flight test validation or from an identification of and acceptance of, the validity of compensation techniques. Pilot acceptance alone is not sufficient.

Trends in Flight Critical Control System Design and Impact on Validation

FCCS design concepts and validation test technology have advanced significantly over the past 20 years since the advent of digital systems, validation techniques and test methodology have been influenced by experiences in the qualification of systems, and system developers have converged on several accepted methods for both test and analyses. For the most part, however, validation technology lags the advances in flight control system design. This has led to problems in the past with flight system validation. For example, multi-channel fault tolerant flight control systems were developed before suitable real-time multi-channel diagnostic/test equipment was developed to support the diagnosis and validation of sophisticated fault-tolerant architectures. In other cases, systems were developed with no convenient way of conducting multi-channel validation tests. Considerable time elapsed before software and hardware methods were developed to both stimulate the systems and instrument them to capture the system response for subsequent analysis.

Where ground support equipment and test methods were developed concurrently with the advanced systems they were to support, validation has proven to be less costly and more time efficient. Much has been learned about how to validate systems to a high degree of confidence, but advancements in flight critical control systems requires a complementary and continuous updating of validation tools and techniques.

Chapter 6 forecasts trends in flight control system design and identifies the potential impact on the validation process. This should provide a basis for developing research and development programs in the area of validation techniques, and for anticipating validation requirements. Although considered mature by some standard, flight control system technology is expected to advance considerably by the year 2000. The forecast in this chapter is purposely far reaching, extending into the first decade of the 21st century. It is intended that this forecast be a catalyst for developing improved validation techniques simultaneously with new flight system concepts, and for influencing flight system design decisions. It is expected that many of the potentially adverse impacts on validation of these advanced flight critical control system designs can be avoided by anticipating the validation requirements early in the design process, and by using many of the emerging structured validation tools and techniques described in the next chapter.

Dramatic increases in airborne computational power, combined with the increasing level of integration will be the principal design drivers for future flight control systems. It is expected that more functions will be included in the domain of those having a direct impact on safety of flight. Another dominant force in future system design will be the demand for increased reliability, maintainability, and availability of aircraft, all leading to increased effective force levels and reduced life cycle costs. System architectures will take on less easily characterized hybrid configurations.

The volume of software and the number of system elements which must be treated as flight critical is expected to increase continuously. There is also a clear trend to eliminate dissimilar and independent back-up flight control systems. On-line decision-aiding systems, are expected to aid the pilot/crew in complex missions and tasks. More highly integrated vehicle management systems may place the entire flight system in the "flight crucial" category for very advanced vehicle of the next century, such as airbreathing hypersonic aircraft.

For highly unstable aircraft, or vehicles implementing artificial structural mode stability, an additional burden is placed on the revalidation aspect of qualification. For such flight control functions, there is little margin for error. The high premium placed on correctly identifying all possible impacts of any design change will require a higher level of sophistication in the revalidation processes. One hundred percent regression testing may be the necessary outcome of the implementation of such functions, unless partitioning specifically to support validation is designed into the system from the beginning. This approach suggests massive test automation as the principal mitigating factor in maintaining a practical validation program. The combination of bandwidth requirements and criticality to change may in fact result in hybrid systems which physically separates and "hardens" the "core system" from other systems.

For some of the functions and systems previously forecasted, there may be no alternative other than to conduct some portion of the validation in an incremental fashion, in flight. This is already the case in validating the flutter margins, stability, and dynamic load capability of aircraft. Although proof load testing is accomplished on the ground, there is no ground test of a full scale vehicle capable in itself of validating the flutter margin. Thus, flight flutter clearance is an early and mandatory part of every flight

test of a new aircraft. In the case of more highly integrated pilot-vehicle-systems, or advanced functions having a direct impact on safety of flight, it may be necessary to develop new validation tests that can be accomplished incrementally in flight, in a similar manner. In this kind of validation, as in the case of flight flutter clearance, there must always be a guaranteed safe return to a previously known condition or configuration. The integration of flight control functions, such as terrain avoidance and terrain following with advanced sensors or systems such as digital terrain maps may require a larger portion of validation testing to be accomplished in flight due to the difficulty of adequately representing the environment in a ground-based laboratory.

The boundaries between non-flight-crucial and flight crucial systems are projected to dissolve with increasing integration of systems. The challenge will be to maintain the validation state-of-the art sufficiently high so as not to impede the introduction of advanced systems in future aircraft. To achieve this goal, the validation process must be thoroughly embedded and integrated with the system design process itself. Furthermore, the level of research in the area of validation technology itself should be commensurate with the rate of progress in systems technology forecasted.

Finally, experience with the current generation of flight control systems shows that the validation effort can be bounded through the use of judicious partitioning and protection of the most safety-critical flight control functions, such as inner loop stabilization. Design trades between level and method of functional integration, and the validation requirements will perhaps have to be more deliberate in the future. History also shows that performance and mission capability are weighted much more heavily than validation difficulty and testability. Therefore it is imperative that validation technology receive sufficient attention to permit the advanced systems of the future to be implemented with reasonable cost and safety.

Emerging Test and Validation Technology

Preceding Chapters have defined a generic validation process, described the state of the art in applying that process to FCCS's and assessed future flight control system advances and their impact on validation. This chapter will describe some emerging technologies which address existing validation issues and will also indicate areas where future research activities are required. Topics will generally be presented in the order they would occur in system development. This corresponds with the sequence used in Chapter 3.

The first activities undertaken in FCCS development are creation of system goals and requirements and development of functional and design specifications. There are a number of emerging technologies which assist in validating system development in these phases. Amongst these technologies are formal proof mechanisms and reliability assessment. The formal proof mechanisms contribute to system validation by showing that each successive step in system development can be mathematically shown to meet the intent of the specifications of the immediately prior stage. Reliability assessment, conversely, can assist in validation by showing the relationship between decisions made in one phase and the impact in a separate phase. For example, if a design decision is made to develop a quad redundant system as opposed to a triplex system with analog back-up, reliability analyses can potentially be used to assess the impact on overall system reliability for specific failure rate and coverage assumptions.

In the analyses, design, and development of current systems, there is a need for many people to share information, for the airframer to cope with interactions between miscellaneous subsystems and to pay special attention to configuration control and documentation updating. This need will become even more pronounced with future systems. To help meet these needs, software designers and end users are turning increasingly to integrated software environments (ISE) that tend to meet three major goals:

- better mastery of the technical complexity
- optimum management of systems development and maintenance process (multiple subsystems made by multiple partners, sometimes in multiple versions along the life depending on the customer)
- easier sharing of information of any kind

Automated software testing offers the opportunity to perform relatively exhaustive testing of the software while at the same time reducing the manpower required to effect the tests. To accomplish automated tests, a command system exercises the control system software through a preplanned set of conditions. High-order language models of the flight control system can be used as a basis for validating. Significant benefits can be obtained in automated testing of operational flight code if the HOL simulation

used to generate the expected result is developed in a different language from that used for the flight code. The test data are automatically compared with the expected results on a point-by-point basis. If the actual and expected results differ by more than a specified tolerance, the data are then sent to a printer or plotter where they are available for visual inspection. The tests can be comprised of static checks, logic checks, and real-time checks. As noted elsewhere in this report, exhaustive automated testing offers the best answer to the question of how much software revalidation is required following code changes since the complete revalidation, which is efficiently performed using automated testing, obviates the need for decisions on how much validation is required. Still at issue is the question of how a set of tests can be identified which assures complete validation of the software.

Experimental testing, for example, injection of faults in simulation models of fault-tolerant systems (both at the hardware and software levels) is in current practice. The method is useful for gaining insight, but it is costly, and it is difficult to assess the completeness of a set of experiments. Better theoretical understanding is needed to guide the construction of efficient test sets and to increase the power of the inferences about fault-tolerant performance that may be drawn from test results.

Models or paradigms are needed that will give comprehensive and consistent descriptions of the validation process. These models should allow clear expression of validation goals, techniques, and procedures and provide clear definition of general concepts, such as confidence, reliability, and validation.
Particular research goals are:

- Models for incremental validation processes that proceed through successive levels of the system life cycle, such as statements of user needs, statements of requirements, specifications, design, integration, implementation, operation, maintenance, and modification. As a special case, models are needed for describing the validation of major changes in a system, such as redesign, or integration into new system environments.

- Clarification of the roles of formal and informal activities within the validation process.

As stated in Reference 7-17, a numerical measure of validation credibility would be very desirable and may be a legitimate goal of validation methodology. Such an overall measure would aggregate the combined impact of efforts both to prevent errors and failures from occurring and developing systems which tolerate those errors and failures which do occur. Such an overall measure has been and will continue to be elusive. In the meantime, a number of technologies are emerging which support the continuing improvement of the ability to validate flight critical control systems. In many cases emerging test and validation techniques do not replace existing techniques. They arise in order to handle the increased complexity of flight critical digital systems and are used along with the methods described earlier in this document.

Conclusions and Recommendations

Working Group 09 of the AGARD Guidance and Control Panel has studied the Validation of of Flight Critical Control Systems FCCS. This effort was undertaken both to provide guidance to those concerned in the validation of FCCS by reviewing and summarizing the validation process and methods and to identify areas of research which the AGARD nations need to explore to enable validation of FCCS envisioned for the future.

Conclusions:

1. Several FBW FCS for military and civil aircraft have been developed and flown successfully to the same safety requirements as aircraft with mechanical FCS.

2. Validation is intrinsic to the development process.

3. There is no universally accepted plan for the FCCS development process.

4. The projected increase in FCCS functionality will require significantly improved validation effectiveness.

5. A framework exists to link top level safety requirements to the FCCS design but quantitative models

do not exist for all elements.

Recommendations:

1. Accelerate development of automated validation methods to predict and improve test coverage, to produce test stimuli, and to evaluate results.

2. More attention must be given to those validation activities that are required for the specification and design phases of FCCS development.

3. Research must be accelerated and expanded to develop new methods and tools for the validation of future integrated systems and aerospace vehicles.

4. Designers should trade-off unique FCCS equipment and software with standard or widely used components to gain the validation experience of those systems.

5. AGARD should continue to include advances in validation research and technology in symposia, working groups, and lecture series.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-AR-274	ISBN 92-835-0650-2	UNCLASSIFIED

5. Originator Advisory Group for Aerospace Research and Development
North Atlantic Treaty Organization
7 rue Ancelle, 92200 Neuilly sur Seine, France

6. Title VALIDATION OF FLIGHT CRITICAL CONTROL SYSTEMS

7. Presented at

8. Author(s)/Editor(s)	9. Date
Edited by G. Belcher, D.E. McIver and K.J. Szalai	December 1991
10. Author's/Editor's Address	**11. Pages**
Various	134

12. Distribution Statement This document is distributed in accordance with AGARD policies and regulations, which are outlined on the back covers of all AGARD publications.

13. Keywords/Descriptors

Avionics	Reliability
Flight control	Integrated systems
Flight control laws	Proving
Fault tolerant software	Validity
Airborne computers	

14. Abstract

This report summarises the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD. The objectives of the Working Group were:

(1) To provide guidance to those concerned in the Flight Critical Control System (FCCS) validation, namely system designers and certification authorities.

(2) To identify the areas of research which need to be explored to enable validation of the next generation of FCCS.

The Working Group tried to review all flight critical control system validation activities which had been completed or were under active consideration, in Europe and the United States.

AGARD Advisory Report 274 Advisory Group for Aerospace Research and Development, NATO VALIDATION OF FLIGHT CRITICAL CONTROL SYSTEMS Edited by G. Belcher, D.E. McIver and K.J. Szalai Published December 1991 134 pages This report summarises the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD. The objectives of Working Group were: (1) To provide guidance to those concerned in the Flight Critical Control System (FCCS) validation, namely system designers and certification authorities. P.T.O.	AGARD-AR-274 Avionics Flight control Flight control laws Fault tolerant software Airborne computers Reliability Integrated systems Proving Validity	AGARD Advisory Report 274 Advisory Group for Aerospace Research and Development, NATO VALIDATION OF FLIGHT CRITICAL CONTROL SYSTEMS Edited by G. Belcher, D.E. McIver and K.J. Szalai Published December 1991 134 pages This report summarises the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD. The objectives of Working Group were: (1) To provide guidance to those concerned in the Flight Critical Control System (FCCS) validation, namely system designers and certification authorities. P.T.O.	AGARD-AR-274 Avionics Flight control Flight control laws Fault tolerant software Airborne computers Reliability Integrated systems Proving Validity
AGARD Advisory Report 274 Advisory Group for Aerospace Research and Development, NATO VALIDATION OF FLIGHT CRITICAL CONTROL SYSTEMS Edited by G. Belcher, D.E. McIver and K.J. Szalai Published December 1991 134 pages This report summarises the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD. The objectives of Working Group were: (1) To provide guidance to those concerned in the Flight Critical Control System (FCCS) validation, namely system designers and certification authorities. P.T.O.	AGARD-AR-274 Avionics Flight control Flight control laws Fault tolerant software Airborne computers Reliability Integrated systems Proving Validity	AGARD Advisory Report 274 Advisory Group for Aerospace Research and Development, NATO VALIDATION OF FLIGHT CRITICAL CONTROL SYSTEMS Edited by G. Belcher, D.E. McIver and K.J. Szalai Published December 1991 134 pages This report summarises the deliberations of Working Group 09 of the Guidance and Control Panel of AGARD. The objectives of Working Group were: (1) To provide guidance to those concerned in the Flight Critical Control System (FCCS) validation, namely system designers and certification authorities. P.T.O.	AGARD-AR-274 Avionics Flight control Flight control laws Fault tolerant software Airborne computers Reliability Integrated systems Proving Validity

(2) To identify the areas of research which need to be explored to enable validation of the next generation of FCCS.

The Working Group tried to review all flight critical control system validation activities which had been completed or were under active consideration, in Europe and the United States.

ISBN 92-835-0650-2

(2) To identify the areas of research which need to be explored to enable validation of the next generation of FCCS.

The Working Group tried to review all flight critical control system validation activities which had been completed or were under active consideration, in Europe and the United States.

ISBN 92-835-0650-2

(2) To identify the areas of research which need to be explored to enable validation of the next generation of FCCS.

The Working Group tried to review all flight critical control system validation activities which had been completed or were under active consideration, in Europe and the United States.

ISBN 92-835-0650-2

(2) To identify the areas of research which need to be explored to enable validation of the next generation of FCCS.

The Working Group tried to review all flight critical control system validation activities which had been completed or were under active consideration, in Europe and the United States.

ISBN 92-835-0650-2

AGARD

NATO OTAN

7 RUE ANCELLE · 92200 NEUILLY-SUR-SEINE

FRANCE

Téléphone (1)47.38.57.00 · Télex 610 176
Télécopie (1)47.38.57.99

DIFFUSION DES PUBLICATIONS AGARD NON CLASSIFIEES

L'AGARD ne détient pas de stocks de ses publications, dans un but de distribution générale à l'adresse ci-dessus. La diffusion initiale des publications de l'AGARD est effectuée auprès des pays membres de cette organisation par l'intermédiaire des Centres Nationaux de Distribution suivants. A l'exception des Etats-Unis, ces centres disposent parfois d'exemplaires additionnels; dans les cas contraire, on peut se procurer ces exemplaires sous forme de microfiches ou de microcopies auprès des Agences de Vente dont la liste suite.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Fachinformationszentrum,
Karlsruhe
D-7514 Eggenstein-Leopoldshafen 2

BELGIQUE
Coordonnateur AGARD-VSL
Etat-Major de la Force Aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Directeur du Service des Renseignements Scientifiques
Ministère de la Défense Nationale
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Defence Research Board
Ved Idraetsparken 4
2100 Copenhagen Ø

ESPAGNE
INTA (AGARD Publications)
Pintor Rosales 34
28008 Madrid

ETATS-UNIS
National Aeronautics and Space Administration
Langley Research Center
M/S 180
Hampton, Virginia 23665

FRANCE
O.N.E.R.A. (Direction)
29, Avenue de la Division Leclerc
92320, Châtillon sous Bagneux

GRECE
Hellenic Air Force
Air War College
Scientific and Technical Library
Dekelia Air Force Base
Dekelia, Athens TGA 1010

ISLANDE
Director of Aviation
c/o Flugrad
Reykjavik

ITALIE
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PAYS-BAS
Netherlands Delegation to AGARD
National Aerospace Laboratory NLR
Kluyverweg 1
2629 HS Delft

PORTUGAL
Portuguese National Coordinator to AGARD
Gabinete de Estudos e Programas
CLAFA
Base de Alfragide
Alfragide
2700 Amadora

ROYAUME UNI
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

TURQUIE
Milli Savunma Başkanlığı (MSB)
ARGE Daire Başkanlığı (ARGE)
Ankara

LE CENTRE NATIONAL DE DISTRIBUTION DES ETATS-UNIS (NASA) NE DETIENT PAS DE STOCKS DES PUBLICATIONS AGARD ET LES DEMANDES D'EXEMPLAIRES DOIVENT ETRE ADRESSEES DIRECTEMENT AU SERVICE NATIONAL TECHNIQUE DE L'INFORMATION (NTIS) DONT L'ADRESSE SUIT.

AGENCES DE VENTE

National Technical Information Service
(NTIS)
5285 Port Royal Road
Springfield, Virginia 22161
Etats-Unis

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Division
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
Royaume Uni

Les demandes de microfiches ou de photocopies de documents AGARD (y compris les demandes faites auprès du NTIS) doivent comporter la dénomination AGARD, ainsi que le numéro de série de l'AGARD (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Veuiller noter qu'il y a lieu de spécifier AGARD-R-nnn et AGARD-AR-nnn lors de la commande de rapports AGARD et des rapports consultatifs AGARD respectivement. Des références bibliographiques complètes ainsi que des résumés des publications AGARD figurent dans les journaux suivants:

Scientifique and Technical Aerospace Reports (STAR)
publié par la NASA Scientific and Technical
Information Division
NASA Headquarters (NTT)
Washington D.C. 20546
Etats-Unis

Government Reports Announcements and Index (GRA&I)
publié par le National Technical Information Service
Springfield
Virginia 22161
Etats-Unis

(accessible également en mode interactif dans la base de données bibliographiques en ligne du NTIS, et sur CD-ROM)

Imprimé par Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ

AGARD
NATO OTAN
7 RUE ANCELLE · 92200 NEUILLY-SUR-SEINE
FRANCE
Telephone (1)47.38.57.00 · Telex 610 176
Telefax (1)47.38.57.99

DISTRIBUTION OF UNCLASSIFIED AGARD PUBLICATIONS

AGARD does NOT hold stocks of AGARD publications at the above address for general distribution. Initial distribution of AGARD publications is made to AGARD Member Nations through the following National Distribution Centres. Further copies are sometimes available from these Centres (except in the United States), but if not may be purchased in Microfiche or Photocopy form from the Sales Agencies listed below.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Coordonnateur AGARD — VSL
Etat-Major de la Force Aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

LUXEMBOURG
See Belgium

NETHERLANDS
Netherlands Delegation to AGARD
National Aerospace Laboratory, NLR

…rch Establishment

…dinator to AGARD
…grams

(MSB)
ARGE)

Dekelia, Athens TGA 1010

ICELAND
Director of Aviation
c/o Flugrad
Reykjavik

ITALY
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

UNITED KINGDOM
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

UNITED STATES
National Aeronautics and Space Administration (NASA)
Langley Research Center
M/S 180
Hampton, Virginia 23665

THE UNITED STATES NATIONAL DISTRIBUTION CENTRE (NASA) DOES NOT HOLD STOCKS OF AGARD PUBLICATIONS, AND APPLICATIONS FOR COPIES SHOULD BE MADE DIRECT TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS) AT THE ADDRESS BELOW.

SALES AGENCIES

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, Virginia 22161
United States

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
United Kingdom

Requests for microfiches or photocopies of AGARD documents (including requests to NTIS) should include the word 'AGARD' and the AGARD serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Note that AGARD Reports and Advisory Reports should be specified as AGARD-R-nnn and AGARD-AR-nnn, respectively. Full bibliographical references and abstracts of AGARD publications are given in the following journals:

Scientific and Technical Aerospace Reports (STAR)
published by NASA Scientific and Technical Information Division
NASA Headquarters (NTT)
Washington D.C. 20546
United States

Government Reports Announcements and Index (GRA&I)
published by the National Technical Information Service
Springfield
Virginia 22161
United States
(also available online in the NTIS Bibliographic Database or on CD-ROM)

Printed by Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ

ISBN 92-835-0650-2