Digital Avionics:
A Computing Perspective

Elisabeth A. Strunk
John C. Knight
(Eds.)

Preface

This is a book about the use of computers in airplanes. It is intended primarily for people with a computer science background who would like to learn more about this computer-dependent application domain. It can also be useful for novices from other disciplines who would like an introduction to the domain, in order to understand the ideas and vocabulary associated with it. The book is suitable for a final-year undergraduate course or a first-year graduate course in avionics systems, or as a reference for an engineer entering the avionics field.

The term *avionics* is a contraction of *aviation electronics*, and *digital avionics* is that part of the avionics field concerned with digital, usually computerized, technology. This is an important field because modern aircraft use digital avionics extensively for a wide variety of applications. Modern autopilots, for example, are very sophisticated devices capable of reducing pilot workload dramatically. With a few rare exceptions, autopilots are completely computerized.

In practice, most cockpit functions are computerized. This is the result of a transition that has occurred in recent years in which older, electro-mechanical technology has been replaced by technology from the computer age. Displays in aircraft cockpits used to be mostly mechanical dials and mechanical graphics. These mechanical systems are being replaced quickly by what are referred to as *glass* cockpits in which displays are presented on monitors similar to those found on personal computers.

The impact of computer technology extends beyond cockpit displays; the term avionics also applies to the use of computers in the aircraft structure. In older aircraft, control of the engines and the control surfaces (e.g., the flaps and rudder) was achieved with mechanical and hydraulic links. The construction costs, maintenance costs, and operational weight of all the different mechanical elements made them targets for replacement with digital technology, and that led to the introduction of *fly-by-wire* control. The term fly-by-wire usually refers to the combination of the communication of control signals over a digital data bus and the use of those signals by computers within the aircraft structure for adjustment of the control surfaces and the engine settings.

As airplanes have thus become flying computer systems, avionics has assumed an increasingly significant role in their development and production. Construction of avionics systems requires large teams of engineers from a wide variety of disciplines, including computer engineering and software engineering. It is impossible for all those engineers to be familiar with the system's complex goals and operating principles. Still, it would be helpful if they all understood generally what avionics systems are for and how they work. It is with that in mind that the editors and authors produced this book.

We have organized this book into three parts. The first provides background material on aircraft and air traffic that is necessary to understand the requirements of the computing systems discussed in the book. The second part describes a spectrum of avionics components, discussing their specific requirements and

design concerns. The third part looks at general dependability issues in avionics, from both the computer engineering and the human-computer interaction perspectives.

Although we edited the book, most of the material was written by students in a graduate seminar on digital avionics that we taught in the Spring of 2004. We are very grateful to the participants in the course for both their work in the course and their production of the material for this text. Their names appear with the chapters that they wrote. Dean Bushey also contributed material, although he did not attend the course. Finally, as well as being an author, Tony Aiello was very helpful in preparing this book. We are very grateful to all of the authors for their efforts.

Elisabeth Strunk
John Knight
Charlottesville, VA 2006

avionics@cs.virginia.edu
Dependability Research Group
Dept. of Computer Science
University of Virginia

Authors

M. Anthony Aiello
Master of Computer Science
University of Virginia
August 2005

2nd Lt. J. Graham Alsbrooks
Master of Science, Mechanical Engineering
University of Virginia
May 2005

Matthew Bolton
Master of Science, Systems Engineering
University of Virginia
May 2005

Lt. Col. Dean E. Bushey
Master of Science, Computer Science
Clemson University
December 1996

Maj. John C. Giordano
Master of Computer Science
University of Virginia
May 2004
Master of Public Administration
Troy State University
1999

Sinem C. Göknur
Master of Science, Systems Engineering
University of Virginia
August 2005

William S. Greenwell
Master of Science, Computer Science
University of Virginia
May 2003

Kathryn A. Klein
Master of Science, Systems Engineering
University of Virginia
August 2005

John C. Knight
Doctor of Philosophy, Computer Science
University of Newcastle Upon Tyne
May 1973

Lori Stotler
Master of Computer Science
University of Virginia
May 2004

Elisabeth A. Strunk
Doctor of Philosophy, Computer Science
University of Virginia
May 2005

Rajat Tikoo
Master of Computer Science
University of Virginia
August 2004

Sarah Waziruddin
Master of Computer Science
University of Virginia
May 2004

Table of Contents

Aircraft Dynamics and the National Airspace System

Chapter 1. Basic Aircraft Dynamics 1
1. The Four Forces and Basic Aerodynamics 1
   1.1. Airfoils and Fluid Dynamics 1
   1.2. Lift 3
   1.3. Weight and Load 3
   1.4. Thrust 4
   1.5. Drag 4
2. Aircraft Orientation 5
3. Aircraft Stability 5
4. Control Surfaces 7

Chapter 2. The National Airspace System and Air Traffic Control 11
1. Types of Airspace 11
2. Controllers and Facilities 12
3. Flight Plans 12
4. VFR and IFR Landing 13
5. Looking Ahead 13

Chapter 3. Navigation 15
1. Coordinate Frames 15
2. Navigation Systems 16
   2.1. Positioning Systems 16
   2.2. Dead-Reckoning Navigation Systems 19
3. Current Practice in Navigation 20
   3.1. Navigation System Characterization 20
   3.2. Navigation Software 20
   3.3. Design Tradeoffs 21
   3.4. System Profiles 21
4. Communications, Navigation, and Surveillance / Air Traffic Management 21
   4.1. CNS/ATM Implementations 24
   4.2. The Future of CNS/ATM 25
5. Closing Comments 25

Avionics Components

Chapter 4. Flight Control Systems 29
1. Flight Control Background 29
   1.1. Traditional Flight Control Systems 29
   1.2. Digital Flight Control Systems 31

1.3. FCS Architecture *35*
1.4. Flight Control Laws *36*
1.5. Looking Ahead *36*
2. Survey of Fly-by-Wire Flight Control Systems 37
2.1. Airbus A320 ..... *37*
2.2. Boeing 777 ..... *40*
3. Flight Control Design Philosophy 42
Chapter 5. Autopilot Flight Director Systems 45
1. AFDS Components 45
2. Modes 46
3. Mode Confusion 49
4. Conclusion 51
Chapter 6. Flight Management Systems 53
1. Short History of Flight Management Systems 53
1.1. Origins of the Flight Management System *53*
1.2. Development of the Flight Management System *54*
2. Main Functions of the FMS 54
3. FMS Components 56
3.1. Flight Management Computer System *56*
3.2. Electronic Flight Instrument System *57*
3.3. Component Interactions *59*
3.4. Brown vs. Gray FMS Systems *59*
4. Conclusion 60
Chapter 7. Mission Specific Avionics 63
1. Functional Requirements – Influence on Design 63
1.1. The UH-60 and MH-60 Variant *63*
1.2. The C-130 and AC-130 Variant *64*
2. Special Requirements for Special Operations 65
3. Common Airframes for Dissimilar Needs 66

Digital Avionics Dependability

Chapter 8. Human Centered Design 73
1. Background 74
1.1. Current Design Practice *74*
1.2. Cognitive Considerations *75*
1.3. Considerations of Human Error *76*
1.4. Human-Centered Flight-deck Design and Philosophy *79*
2. Example Systems 79
2.1. Displays and Information Automation *79*
2.2. Advanced Display Technology *82*
2.3. Automated Pilot Assistants *83*
Chapter 9. Dependability in Avionics Systems 87
1. Dependability 87

2. Faults....88
2.1. Types of Faults....*88*
2.2. Dealing With Faults....*88*
2.3. Electromagnetic Interference....*89*
3. Major Computing Elements and Architectures....90
4. Dependability in Avionics Hardware....91
5. Bus Architectures....92
5.1. Terminology....*92*
5.2. Time-triggered Versus Event-triggered Buses....*92*
5.3. Data Bus Dependability....*93*
6. Software Systems....94
6.1. The Software Lifecycle....*95*
6.2. Aviation Standards and Regulations Governing Software....*96*
6.3. Tools and Techniques for Avionics Software Construction....*97*
7. System Failures....99

Annotated Bibliography....103
List of Acronyms....117
Glossary....121

Part I

Aircraft Dynamics and the National Airspace System

CHAPTER 1

Basic Aircraft Dynamics

Dean E. Bushey Elisabeth A. Strunk

Avionics systems can control many aspects of aircraft operation, but one of their main responsibilities is looking after the basic movement and integrity of the aircraft. The control algorithms used for this purpose can be very complex for any type of aircraft, and they are very different from those that people who drive cars, but do not pilot aircraft, are accustomed to. The control algorithms involve various concepts and definitions that are equally unfamiliar.

We begin our discussion of avionics, therefore, with a brief overview of the physical characteristics of flight and the terminology that surrounds them. We also describe various essential elements of an aircraft's mechanical system. These algorithms, characteristics, terms, and mechanical elements are important to avionics system developers because of their role in defining what the avionics system has to do and how it must behave.

1. The Four Forces and Basic Aerodynamics

Aircraft motion during flight is caused by four forces that operate on the vehicle. The forces, illustrated in Figure 1, are:

- **Thrust:** The force exerted by the engine or propeller, which pushes air backwards with the object of causing a reaction, or thrust, of the airplane in the forward direction.
- **Drag:** The resistance of the airplane to forward motion; the force acts directly opposite to thrust.
- **Lift:** The upward force created by the wings moving through the air, which sustains the airplane in flight.
- **Weight:** The downward force due to the weight of the airplane and its load.

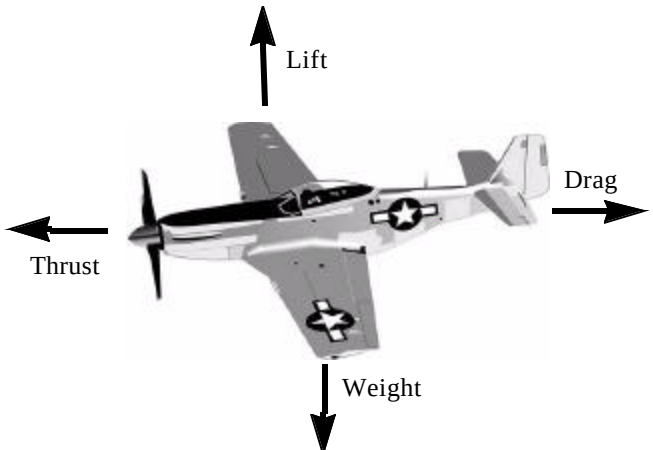

Figure 1. Thrust, Drag, Lift, and Weight

1.1. Airfoils and Fluid Dynamics

These basic four forces combine to give an overall picture of what is happening to an aircraft as it moves through air. The debate over what generates the lift on an airfoil has been going on for decades. There are two prevalent theories. One is the Bernoulli effect, which explains lift through the increase in speed over

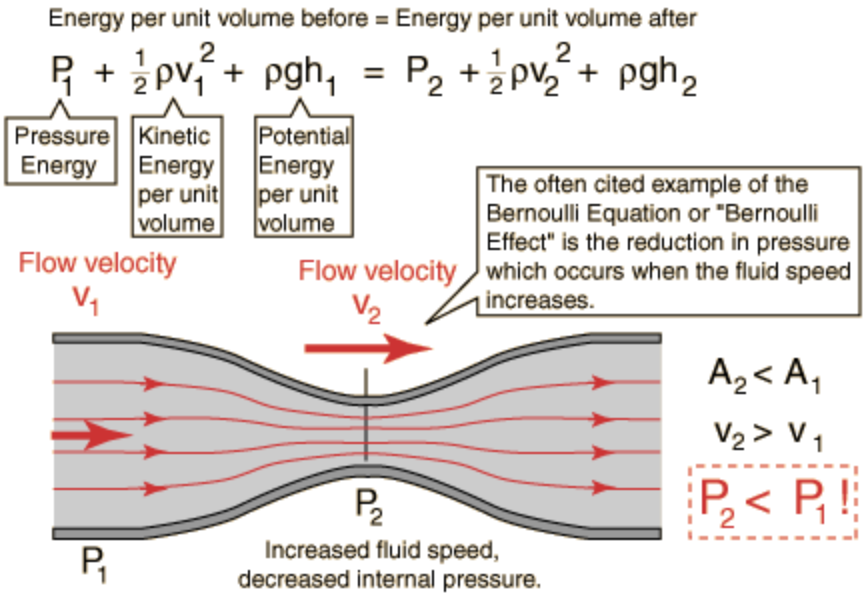

Figure 2. Depiction of the Bernoulli effect[1]

the top of the wing. The other is the Coanda effect, which explains lift using the change in direction of wind over an airfoil.

Daniel Bernoulli (1700-1782) was a Swiss mathematician and physicist who studied fluid dynamics and fluid interactions. He discovered that as fluid increases in velocity there is a corresponding decrease in pressure around the area of increased speed. Applying Newton's laws of the conservation of energy he came up with the basic law of fluids, which says that the sum of the kinetic energy, potential energy and static pressure of freely flowing fluid always remains constant. The mathematical formula that governs fluid (air) flow in and around objects is shown in Figure 2.

The next step is looking at the airflow over an airfoil (such as a wing). Figure 3 shows a cross-sectional view of airflow particles flowing over a wing. The particles in the airflow are traveling at the same speed as they approach the wing, as depicted by the first column. As the flow approaches the wing, the *camber*, or shape of the wing, changes the pattern of the airflow. As shown in the last two columns, the air flowing over the top of the wing actually travels faster than the air over the bottom of the wing.

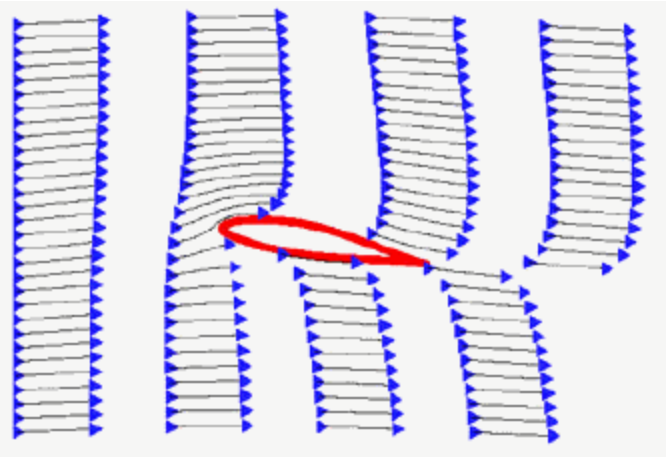

Figure 3. Aerodynamic Forces on a Wing [2]

Applying Bernoulli's theorems to this shows that a lower pressure region is created on top of the wing, relative to the bottom of the wing, and the wing is pulled upward by the difference in pressure. Increasing the angle at which the wing intersects the airflow increases the difference between the air speed over the top of the wing and the air speed over the bottom of the wing. This angle is called the *angle of attack* or *angle of incidence*, and is usually represented in calculations by the character α. It is used to alter the amount of lift

generated by the wing. Lift is also generated on the fuselage due to its similar shape. Here we discuss only the wings, but similar principles hold for the fuselage.

Figure 4. Airflow Over a Wing[2]

An alternate view of the lift created by airflow over a wing is that the change in direction, or downwash, at the trailing edge of an airfoil is what creates lift. Applying Newton's third law, the force of the downwash causes an equal but opposite force in the upward direction on the wing. The governing principle is the Coanda effect, which states that fluids tend to follow the shape of a curved surface.

Whichever theory you follow, the lifting properties of a wing are directly related to the ability of that wing to change the flow of air as it passes over the wing. An airfoil does not have to be curved on the top and/or flat on the bottom in order to work. As long as the airflow is changed, and thus the speed of the air is changed, the airfoil will produce lift. A rounded leading edge helps, but even a barn door will fly, given a positive angle of attack into the windstream.

1.2. Lift

As described in the previous section, lift is generated by the movement of the air around and over the wings, and other surfaces of the aircraft. According to Bernoulli's theory, the air moving over the top of a positively cambered wing must travel faster than air moving along the bottom edge of the wing. This creates low pressure on top of the wing relative to the bottom. This generates a lifting force, and can be mathematically modeled by the following equation:

$$\text{Lift} = C_L \times 1/2\ \rho V^2 \times S$$

where

C_L is the lift coefficient of the wing

$1/2\rho V^2$ represents the dynamic pressure of the airflow

ρ = air density

V = aircraft velocity

S = wing area

Thus to generate greater lift you can: (1) increase the angle of attack (up to a point which will be discussed later); (2) fly faster to increase velocity; (3) move to an area of greater air density; or (4) have greater wing surface area that is in the airstream.

The longitudinal shape, or camber, of the airfoil has a major impact on its aerodynamic characteristics. The more curved the upper surface of the wing, the lower the pressure above it. The curve of the wing is fixed for the most part (the control surfaces that constitute the exceptions to this are mentioned below). The pressure difference can be adjusted, however, by varying the angle of attack as the airfoil hits the windstream.

Figure 5(a) shows the typical C_L vs. angle of attack curve for light airplanes. Lift increases in direct relationship to the increase in angle of attack until a maximum angle, called the Critical Angle of Attack, is reached. Lift then decreases rapidly. This represents the onset of aerodynamic stall. A wing stalls at a predetermined angle of attack: this angle is manufactured into the wing and does not change. Figure 5(b) shows the induced drag factor that corresponds to the lift factor generated.

1.3. Weight and Load

Weight is the downward force of gravity that acts opposite to lift. It is measured at the aircraft's center of gravity; the center of gravity is usually positioned in front of the aerodynamic center for stability (see Section 3). In addition to the basic weight of an aircraft, additional forces are induced when an aircraft changes its plane of motion. These are the forces that make you feel heavier when an aircraft accelerates, banks, or

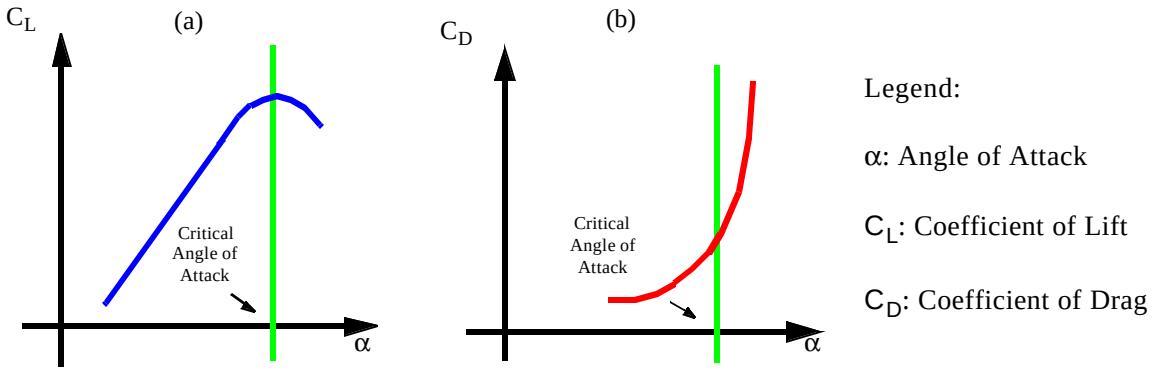

Figure 5. Graph of Angle of Attack vs. Coefficient of Lift and Coefficient of Drag[3]

begins a climb or descent. These are termed *aerodynamic loads*, and usually expressed as a *load factor*. The load factor is the magnitude of the load divided by the aircraft's weight, and is measured in g's, or multiples of the aircraft's weight.

1.4. Thrust

Thrust is the force created by the aircraft's engines that moves the aircraft forward. Thrust is measured as a vector beginning at the aircraft's aerodynamic center, pointing along the aircraft's heading, and orthogonal to the lift and weight vectors. Because it is orthogonal to these vectors, it is not necessarily parallel to the aircraft's fuselage. Increasing the angle of attack requires a corresponding increase in thrust to keep the aircraft's speed (ground speed or airspeed) constant.

1.5. Drag

There are two principal types of drag associated with an aircraft moving through air:

- Friction, or **parasitic drag**: This occurs as the aircraft is moving through the air, and is the "rubbing" resistance of the aircraft to the air through which it moves. This factor increases by the square of the speed through the air.
- **Induced drag**: This is drag induced due to the lifting force created by the airfoils (e.g., wings). It is the result of the wing's work in sustaining the airplane in flight, and is a function of the speed of the aircraft, coefficient of drag, and the density of the air.

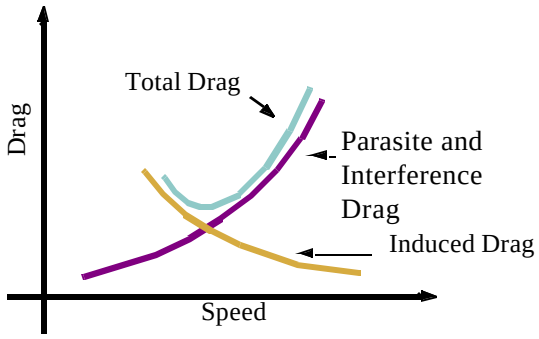

Figure 6. Drag and Speed [4]

Figure 6 shows how drag is related to speed. The point where total drag is lowest is called *Lift over Drag maximum*, or L/D max. Below this point the predominant factor in total drag is induced drag, and above this point the predominant factor in total drag is parasitic drag. Airspeeds below L/D max are an area known as the region of reverse command—that is, the slower you go, the more thrust you must add to overcome the drag.

Two other types of drag that interact with an aircraft in flight are worth noting. Wave drag is drag that comes into play as an aircraft approaches the speed of sound, or Mach speed. As the aircraft approaches the speed of sound, shock waves start to build up, generally starting around 0.75 Mach. Bernoulli's rules no

longer apply when aircraft velocity passes 1.3 Mach. The very nature of lift changes, accompanied by a corresponding change in the drag relationships. The other type of drag that influences flight is called form interference drag, and is due to the shape and smoothness of the aircraft.

2. Aircraft Orientation

Explaining an aircraft's aerodynamic characteristics requires clarification of the terminology used to refer to the aircraft's orientation in 3-dimensional space. The orientation is described using rotation about three axes, whose origins are at the center of gravity of the aircraft. The labels for the 3 axes of the aircraft, shown in Figure 7, are:

- **Pitch**: the axis roughly parallel to the wings of the aircraft. Movement about the pitch axis results in up and down movement of the nose. Pitch angle is sometimes measured as the *angle of attack*, which is angle between the wings' chord line and the relative wind.
- **Roll**: the axis from the nose to the tail of the aircraft. Movement about the roll axis leaves the nose steady but moves the wings up and down. The angle of roll is called the *bank angle*.
- **Yaw**: the axis that runs vertically through the aircraft. Movement about the yaw axis causes the nose to move left and right. The angle of yaw is called the *slip angle*.

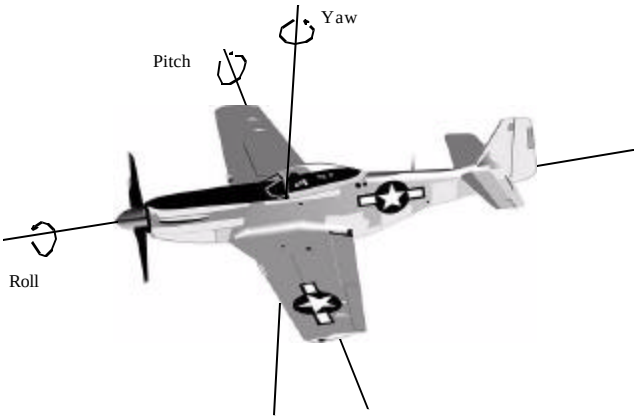

Figure 7. Aircraft Motion

An aircraft's *attitude* is its orientation to the horizon, a combination of the angle of attack and the bank angle. An aircraft is *trimmed* if its angle of attack and its bank angle are both 0.

3. Aircraft Stability

Aircraft control is heavily dependent on an aircraft's *stability* characteristics. Stability is the quality of returning to an original angle of attack, slip, or bank after a disturbance from an equilibrium. The three aircraft axes have corresponding terms to describe their stability:

- **Longitudinal stability**: stability in pitch;
- **Lateral stability**: stability in roll;
- **Directional stability**: stability in yaw.

Another aspect of aircraft stability is the static vs. dynamic characteristics of an airplane. *Static stability* is the quality of initially moving towards a trimmed state immediately after a disturbance. Figure 8 illustrates the conditions of static stability using a ball on various types of surfaces as an analogy. In the

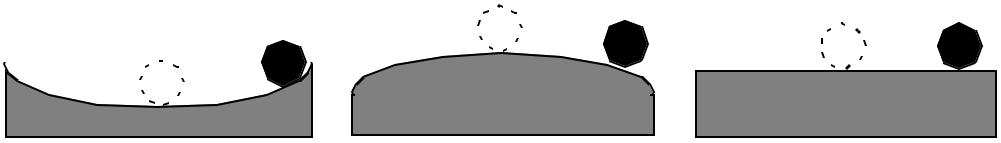

Figure 8. Conditions of Static Stability

statically stable case (a), the ball will initially attempt to return to its equilibrium position when disturbed; just as statically stable aircraft exhibit similar behavior when disturbed in the pitch, roll, or yaw axes. In the statically unstable case (b), the ball will tend to move away from its equilibrium position when disturbed as is the case with statically unstable aircraft. Statically neutral aircraft exhibit no tendency toward or away from equilibrium when disturbed as illustrated in case (c).

Dynamic stability refers to the nature of the oscillations an aircraft exhibits once it is disturbed from equilibrium. Dynamically stable aircraft exhibit *damped* oscillations, meaning that the oscillations lessen in amplitude over time and converge on stable flight. Thus, dynamically stable aircraft are necessarily statically stable. Static stability about an axis does not, however, imply dynamic stability. A statically stable aircraft's oscillations can increase in amplitude over time, averaging out to trim flight but resulting in an aircraft that is out of control.

There are four main categorizations of stability in aircraft:

- **Total stability**: A totally stable aircraft is both statically and dynamically stable, with oscillations about the trimmed position disappearing almost instantly with no need for pilot intervention.
- **Normal stability**: Normal stability is similar to total stability, but the oscillations about the trimmed position take longer to damp out.
- **Neutral stability**: After a disturbance, a statically stable but dynamically neutral aircraft will continue to oscillate about the trimmed position but will not remain trim without pilot intervention.
- **Negative stability**: A negatively stable (unstable) aircraft can be statically unstable, dynamically unstable, or both. Keeping the aircraft under control requires constant intervention, by either the pilot or a computer.

Figure 9 shows possible dynamic stability characteristics for an aircraft that is statically stable about the longitudinal axis.

Stability is an important factor in aircraft design because it affects both how difficult the aircraft is to control and how maneuverable the aircraft is. Stable aircraft are easier to control in normal flight, requiring

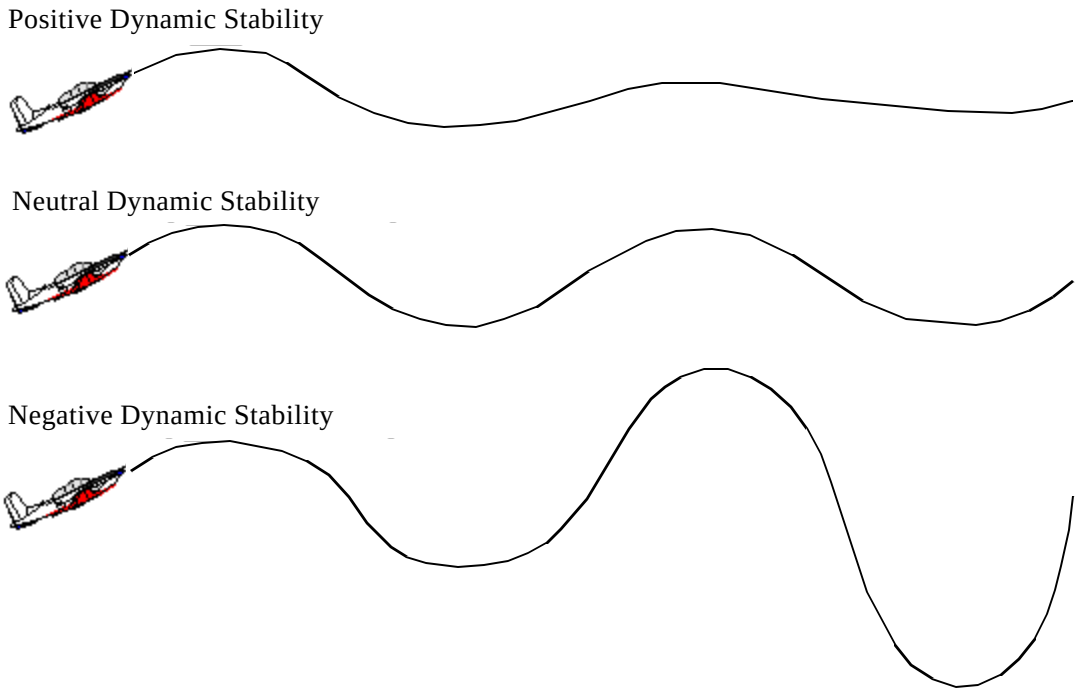

Figure 9. Positive, Neutral, and Negative Dynamic Stability [5]

the pilot to make fewer small adjustments to keep the aircraft trim when any turbulence is present. Unstable aircraft are more difficult to control in normal flight, and sometimes require computer support in order to be controlled by the pilot at all. They are, however, easier to maneuver than stable aircraft, and so they are desirable for many purposes such as air combat.

U.S. law dictates certain stability requirements for passenger-carrying aircraft. Standard commercial aircraft are designed with positive static longitudinal stability. In these aircraft, the aerodynamic center, or the center of lift, is designed to be well aft of the center of gravity. A horizontal stabilizer (see Section 4) is used to offset the nose-heavy impact of this design. Passenger-carrying aircraft are also designed with positive lateral stability through various design features, including increased wing span and position of control surfaces. Positive directional stability is also desired on commercial aircraft, and is achieved through the use of yaw dampeners (fixed attachments to the fuselage), rudder design, and fuselage design.

In contrast to the relatively stable flight characteristics desired of passenger carrying aircraft, military fighter jets are normally designed to be highly maneuverable, having high roll rates, tight bank angles, and able to change pitch attitude very rapidly. The General Dynamics F-16 Falcon provides a good example of some of the design trade-offs made between stability and maneuverability. Its center of gravity is moved forward almost to a point equal to the aerodynamic center. This was previously thought to make an aircraft uncontrollable, but with fly-by-wire technology (discussed in Chapter 4, Section 1.2), the pilot retains the agility and maneuverability desired, while the flight control computer makes continuous adjustments to keep the aircraft from departing controlled flight. This nearly neutral lateral stability, termed “relaxed stability” by designers, gives the F-16 the advantage of being able to make rapid pitch changes without having the force of gravity counteract the desired changes.

The F-16 achieves a very high roll rate (approximately 324 degrees per second through the first 90 degrees of turn) due to its low aspect ratio, swept wing design, and fly-by-wire technology. Positive lateral stability in turns and during these rolls is highly desired, and is achieved by swept wings and a computer-controlled Stability Augmentation System. Once the aircraft has been rolled, the pilot can then employ the relaxed longitudinal stability to maneuver the aircraft along its new plane of flight.

Yaw stability in modern fighter aircraft requires the precise and continuous inputs of the flight control system to compensate for the high rate of change in aircraft flight dynamics. As in commercial aircraft, positive yaw stability is highly desired to ensure coordinated flight.

4. Control Surfaces

Figure 10 shows the typical control surfaces on an aircraft. The ailerons are used for roll control, the elevators are used for pitch control and the rudder is used for yaw control.

Ailerons

A pilot cannot change the shape of the wing, but he can alter the curvature of the outboard sections of the wing by using the *ailerons*. Figure 10 shows the left and right ailerons on the outboard section of each wing. These wing extensions are hinged so that they can be deflected down or up into the airstream, effectively increasing or decreasing the camber of that section of the wing. Ailerons are usually coupled, in that an upward movement in one aileron (camber decrease) will be accompanied by a corresponding lowering of the opposite aileron (camber increase). The aileron movement then increases the lift generated by one section of a wing, while decreasing the lift from the wing on the other side. This imparts a rolling moment in the lateral plane about that axis, usually used to initiate a turn in that direction. Note that this increase in camber and lift will also cause an increase in drag, so that the wing producing a greater lift component will also be producing greater induced drag, and in turn rotate, or yaw, the aircraft’s nose in the direction of the rising wing. This is called *aileron drag* or *adverse yaw*.

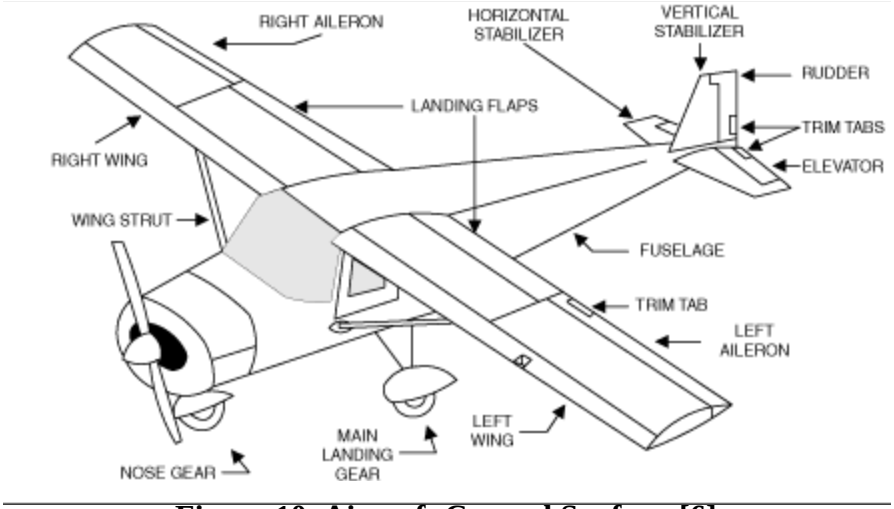

Figure 10. Aircraft Control Surfaces[6]

Elevators

The *horizontal stabilizer* (also known as a *tailplane*) forms part of the tail section of the aircraft and acts like a small wing with a cambered surface underneath to counteract moment created by wings. *Elevators* are attached along the trailing edge of the horizontal stabilizer. Like the ailerons, these are hinged, and can be deflected up or down into the airstream. When the elevators are deflected down the effect is a lifting force on the stabilizer, much like that of the wing. This lifting force produces a pitching moment about the aircraft's center of gravity. When the elevators are deflected up, which is commanded by the pilot pulling the control stick aft, the aircraft will have a decreased lifting force at the tail section, which causes the nose to pitch up. When the elevators are deflected downward the tail produces greater lift, causing the nose to pitch down.

Rudder

The other portion of the tail section is the *vertical stabilizer*, which helps to control the left and right movement of the nose. The *rudder* is hinged along the trailing edge of this vertical fin. It is used to keep the aircraft aligned with the relative airflow, to avoid slips or skids induced by adverse yaw. When the pilot steps on the left rudder pedal, the rudder is deflected left into the airstream. This produces a moment along the vertical axis, causing the nose of the aircraft to track left. Depressing the right rudder pedal causes the opposite effect.

Flaps

Flaps are hinged sections attached to the inner portion of each wing. When extended, they increase the camber of the wing, which increases the lift and drag. They are high-lift devices that are extended during slower flight conditions, such as takeoff and landing, which require greater lift due to the slower speeds. They are normally retracted during cruise flight to allow for greater speed and increased fuel efficiency.

Slats

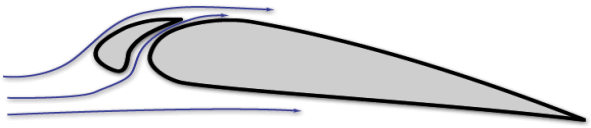

Figure 11. Wing Leading Edge Slats [7]

Some wings are built with leading edge extensions, called *slats*, which help to delay the onset of airflow separation, or stall, on the top of the wing. These devices act to keep the airflow smooth over the airfoil by accelerating the airflow between the slat extension and the leading edge. This modifies the characteristics of the air flow over the wing, so that the critical angle of attack is higher than it would normally be. This allows the aircraft to increase its angle of attack past the point where it would normally stall.

Acknowledgments

We appreciate William Greenwell's contribution of some of the material on stability in this chapter.

Sources

[1] Rod Nave, Dept. of Physics and Astronomy, Georgia State University, http://hyperphysics.phy-astr.gsu.edu/hbase/pber.html
[2] Anderson and Eberhardt, http://www.aa.washington.edu/faculty/eberhardt/lift.htm
[3] http://www.pilotsweb.com/principle/liftdrag.htm#coefficient
[4] http://pilotsweb.com/principle/liftdrag.htm
[5] Ray Preston, Selkirk College, http://142.26.194.131/aerodynamics1/Stability/Page3.html
[6] http://www.gleim.com/aviation/ltf/howtheyfly.php
[7] http://www.zenithair.com/stolch801/design/design.html

CHAPTER 2

The National Airspace System and Air Traffic Control

Elisabeth A. Strunk William S. Greenwell

We now turn to the role of avionics as part of the much larger system in which the aircraft operates—the *airspace system*. *Airspace* is the space in which aircraft are permitted to operate, and an *airspace system* refers to a region of airspace and the ground systems that support it. The airspace system overseen by the United States Federal Aviation Administration (FAA) is called the *National Airspace System* (NAS).[1]

Air traffic control (ATC) is the system that coordinates aircraft to ensure they are adequately separated to maintain safety. ATC has several different types of facilities, and some aircraft can operate without any contact with ATC. An aircraft's interaction with ATC is a significant contributor to avionics system requirements.

1. Types of Airspace

Within the NAS, airspace is classified based on proximity to airports and on *flight level*. Flight levels (FLs) are partitions of the airspace based on altitude. Altitude is primarily measured in feet *above ground level* (AGL) when discussing low altitudes or above the *mean sea level* of the Earth (MSL) when discussing higher altitudes. Flight levels are expressed in units of 100 ft., so flight level 200 is 20,000 ft. MSL.

There are six classes of airspace. The classification of a sector of airspace determines the rules for flight within that sector. Class A airspace extends from FL 180 to FL 600 and is used for *enroute* operations (those during which the aircraft is not approaching or departing an airport). Class B airspace extends from the ground to FL 100 around busy airports, and is shaped to cover the area where aircraft are approaching and departing from the airport. Class B airspace is comprised of a stack of two or more cylinders whose centers are vertically aligned and whose radii increase with altitude, similar to the shape of an upside-down wedding cake[1]. Class C airspace extends from the ground to 4,000 feet above the airport elevation. Class D airspace extends from the ground to 2,500 feet above the airport elevation. Class E airspace is airspace that is under ATC control but which does not fall into the Class A, B, C, or D designations. Class G airspace is airspace that is not under ATC control.

1. The U.S.-specific entities described here have counterparts all over the world. We use the U.S. as a specific example.

2. Controllers and Facilities

Aircraft on the ground at an airport are managed by *ramp controllers*, who work for airlines to manage transit between the gate and the taxiways, and by *ground controllers*, who work for the airport to manage transit on the taxiways to the runways.

Takeoff from a runway and landing on a runway is handled by a *local controller*. The local controller is responsible for scheduling aircraft to use runways and assigns runways in a way that will maintain sufficient separation between aircraft. Scheduling aircraft in a way that maintains required separation can be a difficult job because of the many constraints involved. Airports can have multiple runways, some parallel and some not; taxiways can cross runways so that conflicts might arise between taxiing aircraft and those using the runway; terrain near the airport may complicate approach procedures; and in some cases non-aircraft ground obstacles such as trucks or deer might occupy a runway or taxiway, making it temporarily unusable. Furthermore, the local controller must also take care to maintain separation between aircraft using the runway and flying in the vicinity of the airport. The calculation of required separation itself depends on the size of the two aircraft: a larger aircraft generally has a greater required separation distance around it than a small aircraft, although small aircraft must stay well clear of the *wake disturbance*, or disturbance in the airflow, caused by large aircraft.

Aircraft flying approximately 5 - 50 miles around an airport are handled by a *terminal radar control center*, or TRACON. TRACON controllers are responsible for managing aircraft in the airspace around an airport (terminal) using airport radar. In addition to their primary task of maintaining separation between aircraft, TRACON controllers assign arrival and departure routes to aircraft landing at and taking off from the terminal, respectively. Upon notification from the local controller that a runway is clear, the TRACON controller will give clearance to descend and approach the runway to the next aircraft waiting to land. Each TRACON controller monitors some of the sectors of the airspace, where the size and number of sectors are dependent on the volume of air traffic in the area (sectors are larger or a controller will monitor several sectors at night, for instance).

Enroute flight is generally restricted to specific routes, called *airways*, where aircraft are assigned to routes and flight levels by ATC. *Air route traffic control centers* (ARTCCs) divide covered airspace into sectors, with 1-2 controllers monitoring each sector to ensure adequate separation. The required separation between aircraft is generally higher than in the terminal area because of imprecision in radar coverage, faster travel speeds, and lower volumes of traffic in non-terminal airspace.

As the aircraft progresses along its route, it is handed off to other controllers as it passes from one area to the next. In some remote areas—primarily oceanic regions—no radar coverage exists, so no standard ATC coverage exists. Instead, controllers in specific coastal centers are responsible for communicating with pilots flying in those regions to determine an aircraft's position and intent.

With increasing volumes of air traffic, the time dimension of the flight plan becomes more and more important. Enroute controllers attempt to direct flights to their destination TRACONs as quickly as possible without forcing them to wait for landing clearance. In some cases, they may contact the Air Traffic Control System Command Center, which coordinates flights across the U.S.

3. Flight Plans

To assist ATC in coordinating aircraft, pilots are encouraged (and often required) to file a *flight plan.* A flight plan is a sequence of *waypoints* that the aircraft is anticipated to reach during the flight. Waypoints are typically collocated with radio navigation aids or are specified as offsets from the locations of such aids.

4. VFR and IFR Landing

Weather affects all stages of flight because of the impact it has on the airstream. Also, cold, wet weather can cause wing icing, changing the aerodynamic aspects of the wings. Weather's most significant impact, however, is on visibility during takeoff, approach, and landing. In general, pilots must be able to see the runway before they may land on it. Only specially-certified pilots flying properly-equipped aircraft may land on a runway that is obscured by clouds, and in very poor weather a runway may be taken out of service entirely. The best weather conditions are *visual meteorological conditions* (VMCs), under which *visual flight rules* (VFRs) can be followed. Any licensed pilot is permitted to fly any aircraft on which he is certified under VMCs.[1] Non-VMC conditions are known as *instrument meteorological conditions* (IMCs), and require *instrument flight rules* (IFRs) to be followed. Under IMCs, the pilot relies on instruments to land safely; thus, the aircraft must carry instruments to guide the pilot to the runway, and the pilot must be certified to use those instruments. Differences within IMCs are discussed in Chapter 3, Section 1.

An IFR approach to a runway uses a predetermined route to either land the aircraft or align the aircraft with the runway at a point where the pilot can land the aircraft visually. The beginning of an IFR approach is marked by the *initial approach fix*; the approach then continues through some sequence of maneuvers depending on airport structure and geography; and it ends with a *final approach fix*, the point at which the aircraft should be positioned in a straight line to the runway. If the pilot is unable to see the runway at the *decision height*, the point at which the decision whether to land must be made, then he must execute a *missed approach*, go around, and try again.

Some aircraft are equipped with fully automatic landing systems. Execution of a fully automated approach requires an automatic landing system on the aircraft and a runway that is authorized for such approaches.

5. Looking Ahead

Runway Incursions

With the continual increase of air traffic around major airports, *runway incursions*—incidents where an aircraft or other obstacle (e.g., trucks or deer) is on a runway at the wrong time—are becoming an increasingly significant problem. The pilot might not be able to see runway obstacles when flying in IFR conditions, and so could be unable to avoid them. This problem is on the National Transportation Safety Board's "most wanted" list of aviation safety improvements. Systems designed to aid air traffic controllers in detecting potential incursions are currently being installed in many major airports. Systems for predicting incursions and alerting the pilots whose aircraft could be in jeopardy are currently under development.

Reduced Vertical Separation Minimum

Another problem with air traffic growth is routing aircraft while they are enroute. Currently, the standard required *vertical* separation (separation in altitude) in enroute airspace is 2000 ft. Combined with horizontal restrictions on possible flight routes, the vertical separation requirements severely limit the number of aircraft that can use a volume of airspace at any given time. The Reduced Vertical Separation Minimum (RVSM) program, underway in several parts of the global airspace, aims to reduce the vertical separation requirements above flight level 290 to 1000 ft., permitting more aircraft to use the airspace at the same time. Any such reduction in the margins for safety of an engineered system must take careful account of whether the reduction will impact system safety, and the program is proceeding carefully to avoid the increased potential for accidents. The system is currently in development for many portions of the world's airspace.

1. Barring restrictions independent of weather conditions.

Free Flight

A potentially more powerful way to increase the capacity of the airspace is to reduce the role of ground systems in controlling aircraft flight paths. Free flight proposes to do away with direct control of aircraft from the ground, replacing it instead with onboard flight path planning and collision avoidance systems. This is a highly experimental concept that is still in the stage of designing basic algorithms and examining methods through which onboard software might be engineered to replace the controller with equivalent assurance of safety. A more near-term approach, CNS/ATM, is discussed in Chapter 3, Section4.

Small Aircraft Transportation System

The most problematic areas of the world's airspace are around congested airports. While it is important to increase the enroute airspace capacity, such a solution should be coupled with a solution for expanding terminal capacity in order to be truly effective. The Small Aircraft Transportation System (SATS) is a program created to allow greater utilization of the many airports that do not possess control towers by developing personal-use aircraft that could be flown between small airports by individuals with little pilot training. There are many obstacles to this program, though, particularly the question of what role automation should play in a system where pilots are expected to have far less training than current commercial—or even private—pilots.

Acknowledgments

We appreciate the help of Ellen Bass in reviewing this chapter and making helpful suggestions for its improvement.

References and Further Reading

[1] FAA's Aeronautical Information Manual, or AIM: http://www.faa.gov/ATpubs/AIM/index.htm
[2] http://virtualskies.arc.nasa.gov/main/matm.html

CHAPTER 3

Navigation

Matthew Bolton *Kathryn A. Klein* *Sinem C. Göknur*

Navigation is defined as the determination of the position and velocity of a moving vehicle. Navigational data is generally represented as a six-component state vector (described below) that contains three position components and three velocity components. Navigation sensors can be located in a variety of places: on the ground, in space, on the vehicle itself, or on other vehicles. By definition, navigation requires that the state vector be calculated onboard the vehicle monitoring it. When a particular vehicle's state vector is calculated by an outside entity, the process is called *surveillance* or *position location* [14].

It is also important to distinguish between navigation and *guidance*. While navigation is exclusively concerned with the onboard derivation of a vehicle's state vector, guidance is concerned with directing a vehicle to a known location with or without computation of the state vector[14].

Navigation is extremely important to aviation since, without a means to navigate, an aircraft would have little to no way of determining its location. It is also a significant application of digital systems: the digital systems are used to calculate position from instrument measurements and to combine different sources of information to produce a more accurate result. This chapter provides an overview of current navigation systems, a description of how they are implemented, and a projection of where they are headed.

1. Coordinate Frames

All navigation is done with respect to a coordinate frame. Coordinate frames can be selected and/or designed to meet the needs of a particular navigation system. For distances as short as a few hundred kilometers, *map grids* (grids overlaid on a map of the earth's surface) can be created to enable precise location with respect to particular physical entities. For example, an airport might have its own map grid, as might a runway. The map grids' origins are mapped to latitude-longitude coordinates to situate them with respect to other coordinate systems.

Long-range aircraft use *earth-bound coordinate frames*, coordinate frames that encompass the entire earth and move in space with the earth's rotation. The most common earth-bound coordinate frames are latitude-longitude-altitude and rectangular x, y, z. In rectangular coordinates (illustrated in Figure 1), x is the distance along the Greenwich Meridian, y is the distance along the 90° Meridian, and z is the distance from the plane the equator creates through the earth to the vehicle. There are also *earth-centered* coordinate frames, commonly used by spacecraft, in which the origin is at the earth's center but the coordinates of an object in space do not change as the earth rotates [13].

Precision of navigation is also important, and is critical when aircraft are flying in weather conditions where the pilot cannot see the runway until the aircraft is very close to it. The International Civil Aviation Organization (ICAO) has standardized categories of landing conditions:

- **Category I**: Landing must be aborted if the runway is not visible at 200 feet.
- **Category II**: Landing must be aborted if the runway is not visible at 100 feet.
- **Category III**: Landing must be aborted if the runway is not visible at 50 feet.

Instruments are rated for the categories under which they can be used for landing. These ratings depend on the precision with which the system can measure location.

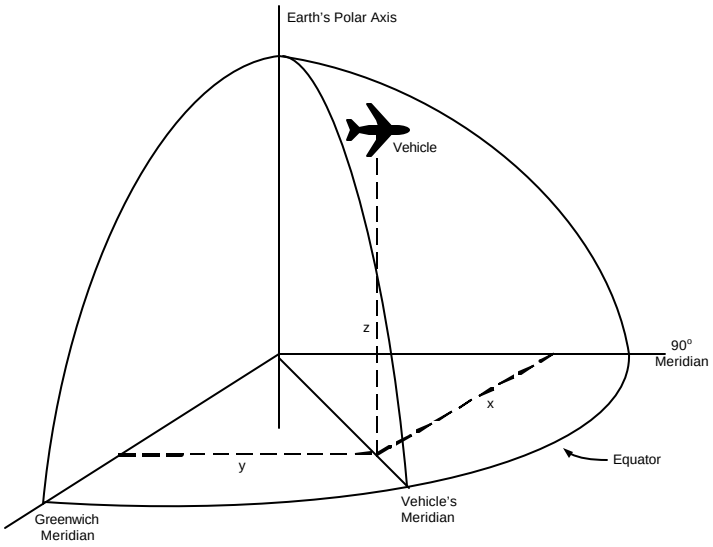

Figure 1. *x, y, z* Coordinate Frame[13]

2. Navigation Systems

There are two types of navigation systems: *positioning systems* (also known as *absolute navigation systems*) and *dead-reckoning systems*. Positioning systems calculate the aircraft state vector without considering the previous path of the vehicle. Dead-reckoning systems derive the state vector based on a series of positions relative to an initial one[14].

2.1. Positioning Systems

There are three types of positioning navigation systems: *radio systems*, *celestial systems*, and *mapping systems* [14].

Radio Navigation Systems

Radio systems are based on a network of transmitters and receivers that interact with the vehicle. Transmitters and receivers can be placed on the ground, on satellites, or on other vehicles. When an aircraft receives signals, it calculates its position relative to the known location of the transmitters. An aircraft determines its velocity by measuring the Doppler shift of transmissions or through sequential position measurements[14].

Avionics employ two types of radio navigations systems: *terrestrial* radio navigation systems and *satellite* radio navigation systems. Terrestrial systems utilize ground-based transmitters, while satellite systems have their transmitters located in satellites [14].

VORTAC

The most common type of terrestrial radio navigation comprises three components. The first is the *VHF Omnidirectional Range*, or VOR, system. VOR ground stations transmit two types of signals, and aircraft can determine their bearing by measuring the phase shifts between the two. The second component is the *Tactical Air Navigation* (TACAN) system. It provides directional information also, but is used by military aircraft while VOR is primarily for civilian aircraft. The third component is *Distance Measuring Equipment* (DME). Distance to the DME station is measured by sending a pulse to the station and then analyzing the time delay between when a the station is interrogated and when a reply is received[13].

These three systems are generally collocated at one VORTAC station. VORTAC will be the primary navigation system used for en route through non-precision approach in the national airspace until satellite radio navigation systems replace it[6].

LORAN

Another commonly used radio navigation system is *Long Range Aid to Navigation* (LORAN), which is used for non-precision airport approaches[13]. In order to be effective, a vehicle needs to be in contact with at least three LORAN stations, usually a chain which is made up of a master station and two or more secondary stations[6]. LORAN uses broadcast time differences to position a vehicle on a hyperbola between two stations. By doing this for two pairs of stations, with a single overlapping station between them, the position of the vehicle is derived by finding the intersection point between the two hyperbolas [14]. While several different types of LORAN have been used in the past (LORAN-A, LORAN-D, and LORAN-C), only LORAN-C is currently in use[14].

ILS

Another important radio navigation system is the *Instrument Landing System* (ILS). In the ILS, transmitters providing horizontal and vertical guidance are placed adjacent to the runways. The vertical descent profile transmitted by the ILS is called the *glideslope*. Using ILS, an airplane can be guided from 15 nautical miles away from the runway to 50 feet above the runway. ILS is compatible with ICAO Category I landing[13].

GPS

The most accurate radio navigation systems are those based on the *Global Positioning System* (GPS). GPS is a satellite radio navigation system that is composed of 24 satellites and 6 ground stations [17]. Using GPS, the state vector is derived from one-way ranging measurements received from four or more satellites [6]. GPS accuracy depends on a variety of different factors including the receiver being used, the number of satellites that can be seen, and ionospheric interference[20]. With this considered, GPS accuracy can vary drastically. In general, GPS can be expected to be accurate to within 2.5 meters 50% of the time and within 7 meters 95% of the time for horizontal position, and to within 20 meters 95% of the time for vertical position[19, 20].

Differential GPS

Differential GPS utilizes ground stations at known locations that receive GPS signals and transmit the measured errors to other GPS receivers. This provides receivers with the means to correct errors in the GPS signal. Differential GPS can be accurate to within a few centimeters for a fixed observer[6].

Two emerging technologies that utilize differential GPS are the *Wide Area Augmentation System* (WAAS) and the *Local Area Augmentation System* (LAAS). In WAAS (see Figure 2), a GPS signal is received by both an aircraft and a reference station. The signal received by the reference station is sent to a master station, via a terrestrial network, where corrections to the GPS signal are calculated. The corrections are then sent to an uplink station which transmits them to a geostationary satellite. Using the same frequency employed by GPS, corrections are broadcast to the aircraft. In addition to providing measurements that are accurate to within 1.5 to 2 meters, WAAS can also indicate GPS problem or blackout areas [17].

LAAS is meant to allow for Category I, II, and III precision approaches within a twenty to thirty mile radial coverage, and can be accurate to within a meter. LAAS accomplishes this through a series of ground receivers that provide references for GPS signal corrections. These corrections are broadcast to aircraft via a VHF data link. LAAS can be aided by a series of terrestrially-based *pseudolites*, placed near the airport,

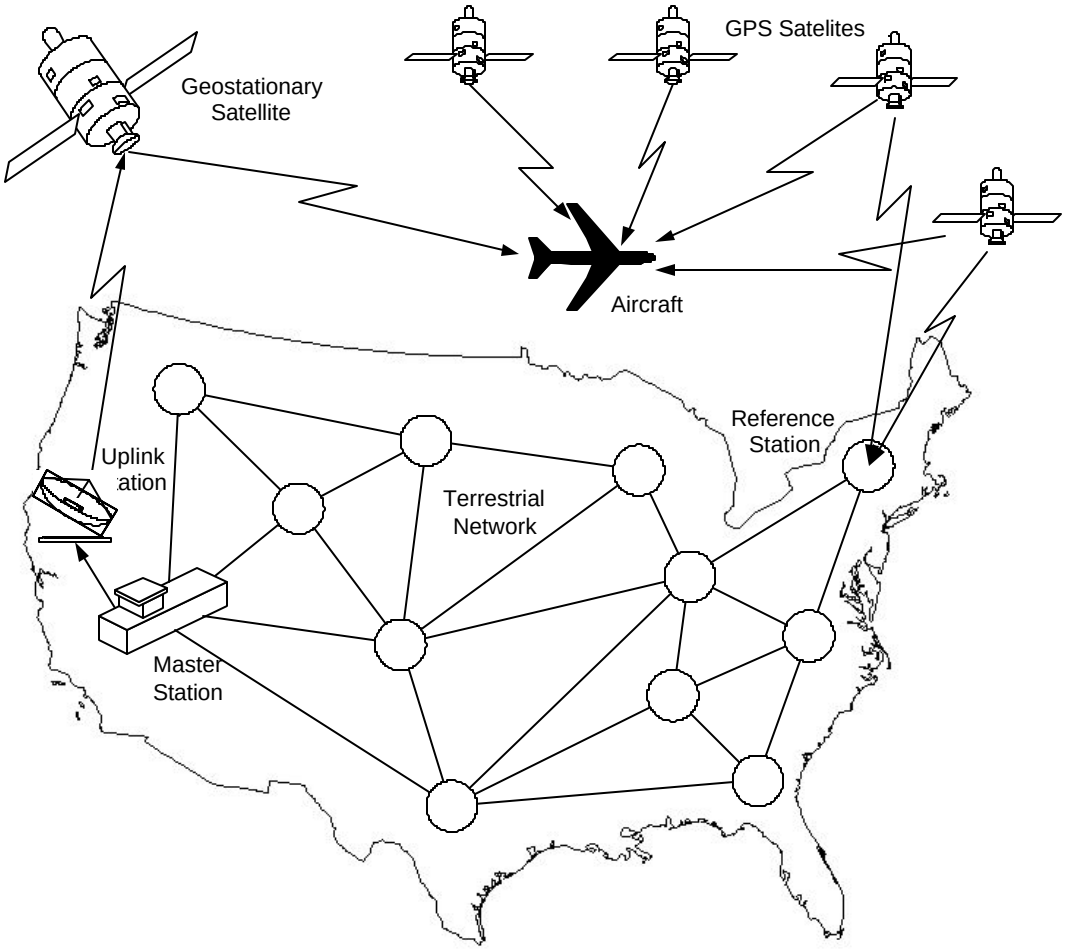

Figure 2. The Wide Area Augmentation System (WAAS)

which mimic GPS satellites and enable a greater precision to be reached over a short range around the airport[17].

Celestial Systems

Celestial navigation has been used at sea for hundreds of years and in aircraft from the 1930s to the 1960s. Modern high-altitude aircraft use celestial navigators, called star trackers, in conjunction with inertial navigation[14]. By using dead reckoning to determine a rough estimate of the telescopic position of a particular heavenly body, a star tracker uses the observed position of the body to derive a more precise position measurement.

Mapping Navigation Systems

Mapping navigation systems use radars, optical sensors, and/or digitized video to map the terrain underneath the aircraft[13]. The observed terrain is matched to known terrain data stored in a database[7]. Such systems are frequently used to reduce the effects of drift, assist landing on fields that lack electronic aids, or assist guidance software that automatically steers the vehicle[13].

2.2. Dead-Reckoning Navigation Systems

Dead-reckoning systems use a series of measurements to determine the state vector of a vehicle based on a known initial position. Dead-reckoning systems must be reinitialized to a new initial position over the course of flight in order to reduce the effect of accumulated errors.

There are two types of dead reckoning systems: those that measure heading and either speed or acceleration, and those that measure continuous radio signals[14]. The following subsections will describe these by first exploring the tools used to measure heading, acceleration, and velocity. Continuous wave radio signal systems and inertial navigation systems will then be discussed.

Heading

The simplest way heading is measured is with a magnetic compass, which is usually accurate to within two degrees. Because the poles of the earth's magnetic field do not correspond to the earth's true poles, adjustment of a magnetic compass's readings is often required. A similar compensation must occur in order to account for the magnetic fields generated by the vehicle in which the compass resides [13].

Another heading sensor is the *gyrocompass*, or *directional gyroscope*. A gyrocompass has two degrees of freedom in a gimbal arrangement. The outer axis is placed normal to the floor plane while the inner axis is within the floor plane. Heading is read from the outer axis. A motor attached to the outer axis allows the compass to hold a preset heading by compensating for the effects of the vehicle's motion[14]. Under such conditions, a gyrocompass can be accurate to within a degree but has a tendency to drift at 0.5 degrees an hour. In order to compensate for this, a gyrocompass is often coupled with a magnetic compass[13].

Speed

The most common way to measure speed is with a *pitot tube*. Using an air-data computer, a pitot tube is able to derive airspeed by examining the pressure of the airstream. The ground speed is calculated by vectorally adding in the wind velocity. Errors can result from unpredicted wind and air currents[13].

Ground speed can also be measured using Doppler radar. This is done by measuring frequency shifts in return radar signals from the ground. Multi-beam Doppler radars can be used to measure all the velocity dimensions of an aircraft's state vector[13].

Continuous-Wave Radio Systems

Navigation systems that monitor signals from continuous-wave radio stations are able to follow lanes that can be counted to keep track of course. Fine positioning can be achieved through phase measurement. If a gap in radio coverage occurs, the system must be reinitialized[14].

Inertial Navigation

Inertial navigation is the most precise dead-reckoning navigation system[13]. The basic structure of such systems is a set of accelerometers that are collocated with a series of gyroscopes[14]. The accelerometers measure multiple dimensions of linear acceleration while the gyroscopes measure rotational accelerations[13]. An onboard computer system translates these readings into navigation coordinates[14] by integration to produce velocity and a second integration to produce distance. High-precision systems are generally known as *Inertial Navigation Systems* (INSs), while lower-precision systems are known as *Air Data/Heading Reference Systems* or *Attitude and Heading Reference Systems* (AHRSs)[4].

Inertial navigators are either directly fastened to the air frame or mounted on a stabilized platform. When mounted on the airframe, the navigator is subjected to all the forces exerted on the aircraft. Stabilized navigators greatly reduce the effect of these forces on the navigator and are often more precise[13].

Inertial navigators can measure orientations to within 0.1^{o}, but have a tendency to drift at about 1 to 2 nautical miles per hour[7]. Fault-tolerant inertial navigators have been able to arrange redundant gyroscopes and accelerometers so that sensor failures can be detected and handled[13]. Additionally, inertial navigators are advantageous for the following reasons: their position and velocity measurements are

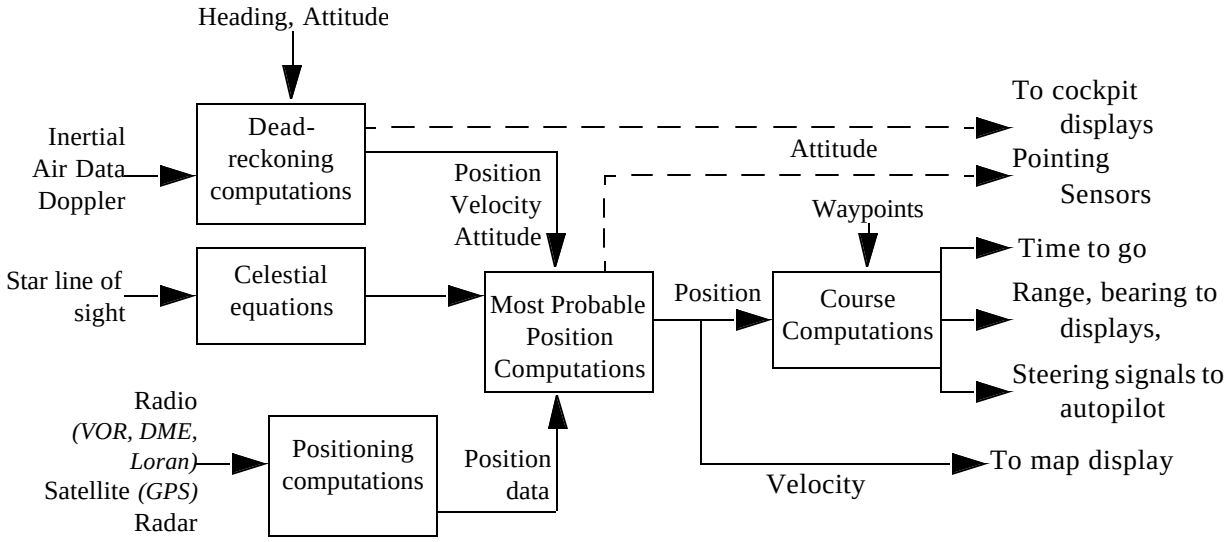

Figure 3. Navigation System [14]

instantaneous and continuous; they are self-contained, making them un-jammable; measurements can be made independent of weather or geographical location; and they are able to produce position and ground-speed information[14].

The most recent inertial navigation systems have been able to reduce drift rates to less than 0.1 degrees per hour by using evacuated cavities or optical fibers that contain counter-rotating laser beams. The phases of these lasers are compared in order to derive acceleration[13].

3. Current Practice in Navigation

In this section common properties and implementation strategies for modern navigation systems are discussed. First, the common implementation strategy of navigation systems is characterized. This is followed by a description of the development strategy for navigation software and a discussion of tradeoffs designers must consider. This section concludes by profiling the navigation systems in several commercial aircraft.

3.1. Navigation System Characterization

Figure 3 is a block diagram which depicts a conceptualization of a general air navigation system. In order to perform its calculations, the system uses information from all three types of navigation: dead reckoning, celestial navigation, and positioning. This information is combined in order to acquire an estimate of the aircraft's position, velocity, and altitude. By combining these estimates with waypoint information, an even more accurate navigation reading can be determined. Once the computation is complete, the navigation data is displayed to the pilots or distributed to the flight management system (described in Chapter 6), autopilot (described in Chapter 5), and other systems[14].

3.2. Navigation Software

Nearly all of the computations in the navigation system are done by computer, whether at the hardware or software level. Navigation software is embedded in either a central processor, with other aviation software, or in navigation computers. It is the job of the software to process sensor data using well-established algorithms. In order to accomplish this, the software utilizes calibration constants, initialization sequences, self-test algorithms, and reasonability tests. The software is also responsible for providing alternate processing schema for handling situations in which sensors fail or are not receiving data[13].

In order to compensate for discrepancies and errors that may occur between sensors, navigation software uses a *Kalman filter*. A Kalman filter is an algorithm that calculates the best estimate of the state vector by considering the dynamics of each sensor. Unfortunately, this filter has poor numerical accuracy when using ill-conditioned measurements[16]. This can be partially alleviated through the use of the U-D measurement update algorithm, which provides better numerical precision, has reduced memory requirements, and requires fewer numerical operations [16].

3.3. Design Tradeoffs

The design of navigation systems, as with the design of all other systems, must deal with all of the following tradeoffs:

- **Accuracy**. Accuracy concerns the amount of error in the system's calculations. The maximum allowable error for the navigation equipment of civil aircraft is based on the probability of a collision for a typical flight. This is specified using a circular error probable (CEP), which is in meters per nautical mile[13].
- **Autonomy**. Autonomy applies to the aircraft's ability to calculate its position without any external help[13]. Autonomy is usually only an issue for aircraft that operate in areas with little to no radio navigation coverage, but can be extremely useful in the event of a failure of non-autonomous navigation equipment.
- **Latency**. Latency is the amount of time it takes the computer to calculate and report the navigation information[13]. This can be caused by a variety of phenomena including computer processing delays, gaps in satellite coverage, and radar scanning[14].
- **Coverage**. Coverage refers to where an aircraft can use a particular service. Terrestrial radio navigation systems that broadcast below 100 MHz can be utilized beyond line of sight. Systems operating above this frequency cannot[13]. Satellite systems can provide much more extensive coverage. A single satellite can cover millions of square miles while constellations of them can provide coverage to the entire world[14].
- **Integrity**. Integrity relates to the system's ability to report excessive errors in a timely fashion. Enroute navigation requires that a warning be generated in thirty seconds if an error threshold is exceeded. Non-precision landing aids must generate warnings within ten seconds. Precision landing aids must generate them in five seconds[14].
- **Cost**. Cost refers to the resources required to build and maintain the system. This can include sensors, electronics, computer software, computer hardware, transmitters, and equipment maintenance[13]. Improvement to any of the other tradeoff areas will inherently add to the cost of the system.

3.4. System Profiles

In order to portray the state of navigation systems in commercial aircraft, this section characterizes the navigation systems of several different aircraft. Feature profiles of the Airbus A300-600, the Airbus A330 and A340, the Airbus A380, the Boeing 767, and the Boeing 777 can be found in Table 1. Most of these aircraft now come with the option to include Future Air Navigation System equipment. The Future Air Navigation system is explored in great depth in next section.

4. Communications, Navigation, and Surveillance / Air Traffic Management

Communication, Navigation and Surveillance / Air Traffic Management (CNS/ATM) is a technology being developed that utilizes satellite communications to manage aircraft. In this system, flight crews and air traffic controllers communicate with each other by means of a data link to a satellite-based network. The system also makes use of current GPS technologies for the purpose of positioning[2].

Aircraft Type	Features
Airbus A300-600	Two VHF omnidirectional radio rangers Two instrument landing systems Marker beacon receivers Two radio altimeters Two Honeywell air data computers
Airbus A330 and A340	Automatic direction finder Two VHF omnidirectional radio rangers Two instrument landing systems Two distance measuring systems Marker beacon receivers Two radio altimeters Two Honeywell air data computers
Airbus A380	Honeywell satellite communications system Goodrich air data systems Rockwell Collins multi-mode VHF and HF receiver radios Northrop Grumman LTN-101E inertial navigation system
Boeing 767	Honeywell VHF omnidirectional radio Integrated Instrument Landing System Marker beacon detector Automatic direction finder Distance measuring system Radio magnetic indicator
Boeing 777	Honeywell air data and inertial reference system (ADIRS) with a six-ring laser gyroscope Honeywell Terrain Collision and Avoidance System Honeywell and BAE Systems twelve channel GPS system ARINC 629 data bus integration

Table 1: Navigation System Profiles [12]

CNS/ATM, or the Future Air Navigation System (FANS), as it is sometimes called, was initially developed by the International Civil Aviation Organization (ICAO) in 1983 in order to rectify limitations in the navigation infrastructure. These included all of the following[10]:

- **Line-of-sight restrictions**: Ground-based navigation aids were extremely limited because their coverage was reduced due to the curvature and terrain of the earth.
- **Scalability**: The existing system was difficult to scale, making it difficult to meet increased demands around airports.
- **Transmitter limitations**: Many terrestrial systems utilized analog information transmission. Analog communications did not permit high rates of data transmission and occupied a large portion of the valuable frequency spectrum.
- **Lack of automation**: Even if the systems in question could transmit and receive data at high speeds, the automation required to interpret that information did not exist.

In order to solve these problems, the ICAO proposed CNS/ATM, which contains three elements (see Figure 4). The first element is communications. In order to provide communications between aircraft and air traffic control, CNS/ATM utilizes a digital data link which provides high speed transfer rates, increased reliability and integrity, improved frequency spectrum utilization, and improved interfacing with automated systems.

The second important element of CNS/ATM is its navigation capabilities. CNS/ATM navigation is based on the Global Navigation Satellite System (GNSS) which includes both GPS and GLONASS[11].

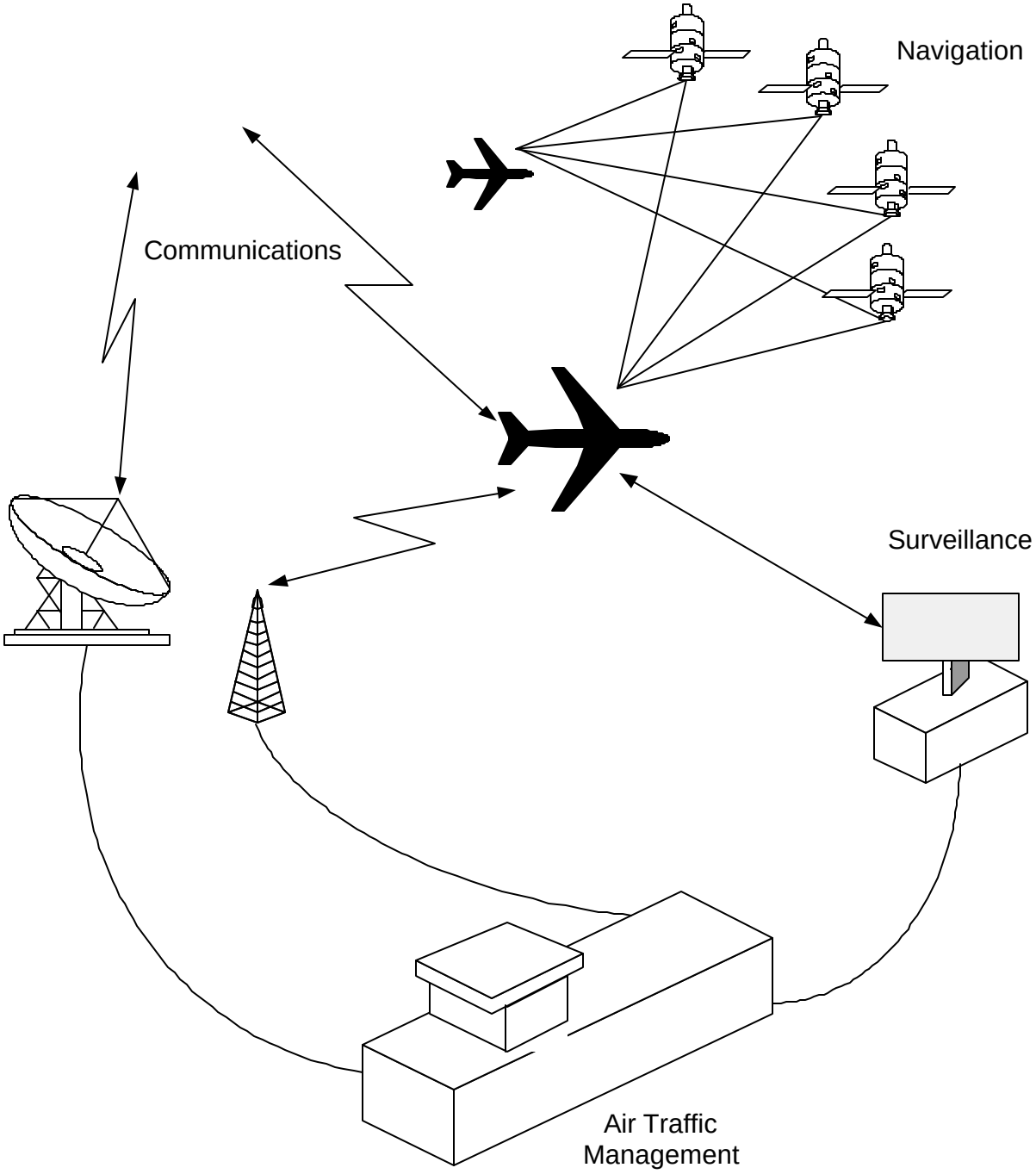

Figure 4. CNS/ATM

CNS/ATM addresses Required Navigation Performance (RNP), a set of standards created by ICAO that requires an aircraft's navigation performance to meet specific levels of accuracy, integrity, continuity, and availability for the enroute, approach, landing, and departure phases of flight [11]. This accuracy may have the added advantage of increasing fuel efficiency[2].

The final element of CNS/ATM is surveillance. By utilizing its new communications system and accurate navigation system, CNS/ATM gives an airplane the ability to transmit its position and other important information to air traffic control and other aircraft. Such communications are required in the CNS/ATM system. This has the added advantage of reducing the amount of separation needed between aircraft[10].

CNS/ATM is intended to be developed in several stages. The first, which covers oceanic and remote airspaces, was put into operation in 1998. The second stage covers high density airspaces. It was intended to be implemented in 2002, but is still under development. CNS/ATM will eventually be available all over the United States and Europe[15].

4.1. CNS/ATM Implementations

Both Boeing and Airbus have reacted to the coming transition to CNS/ATM. The Boeing implementation of CNS/ATM has been deemed FANS-1. It includes all of the following features [2]:

- Communication
  - **An airline operational control (AOC) data link**: Provides the ability to transmit new routes, position reports, and wind data through the data link network to the airline data system.
  - **Air traffic control (ATC) data link**: Allows the flight crew to request new flight plans or changes to an existing flight plan. Air traffic control can also directly request tactical changes to a particular aircraft's flight plan.
- Navigation
  - **GPS integration**: GPS serves as the primary means of determining position. However, it must be able to meet the requirement of the RNP.
  - **RNP**: Addresses the accuracy, integrity, and availability requirements prescribed by CNS/ATM.
  - **Required time of arrival**: This gives the flight crew the ability to assign time constraints to waypoints, resulting in the aircraft passing waypoints at a specific time (plus or minus thirty seconds). If a required arrival time is not going to be met, the flight crew is notified with a visual alert.
- Surveillance
  - **Automatic dependent surveillance**: Reports the current flight position of the aircraft via satellite or VHF data link between the aircraft and the air traffic controller.

This system is currently a standard feature on the Boeing 777 and is an available option on the Boeing 747-400, Boeing 757, Boeing 767, MD-11, and MD-10F[2].

In order to address CNS/ATM, Airbus created a program called Airbus Interoperable Modular Future Air Navigation System (AIM-FANS). AIM-FANS was charged with finding a means of implementing CNS/ATM in Airbus aircraft that would be easy to install and upgrade, reducing the costs and complications of future changes[18].

Airbus's first version of CNS/ATM, FANS-A, was made available as an upgrade to the A330/A340 at the end of 1996. It has since been retrofitted to nearly all of Airbus's commercial airliners. In developing FANS-A, Airbus considered all of the following to be high priority goals[15]:

- Make their aircraft adaptable to various CNS/ATM environments.
- Be able to cope with a moving environment (adapt to meet regional functional requirements).

- Minimize the burden for airlines to transfer to CNS/ATM based systems.
- Establish standards to minimize retrofitting procedures.
- Be compatible with aircraft intermediate standards.
- Minimize the effects of the new system on peripheral equipment.
- Create a user-friendly interface for their ATC data link that can be optimally integrated with the existing functionality.

In both FANS-1 and FANS-A, the ATC and AOC data link features were implemented using the *Aircraft Communications Addressing and Reporting System* (ACARS). Developed in 1978, ACARS was designed to reduce the number of voice communications needed with air traffic control and thus alleviate frequency congestion around airports. Because ACARS was not specifically designed for use with CNS/ATM and AOC, it has several limitations, including controlling access to proprietary networks, management of frequency allotment, and performance[8].

Both FANS-1 and FANS-A are currently limited to areas that lack suitable radar or where voice networks are congested or ineffective [6]. However, coverage will improve as CNS/ATM evolves.

4.2. The Future of CNS/ATM

The development of FANS was meant to be an evolutionary process and as such, continues to change as time progresses. As such, it will likely change in the near future. As of July 2004, an enhanced version of FANS-A, deemed FANS-A+, has been certified for use in the A340 and A330 family of aircraft [1]. Airbus has also been working on the development of its FANS-B standard for use in high density airspace since 1996[18].

From the navigation perspective, GNSS will be implemented progressively through incremental system improvements. Current and near-term GNSS will utilize the existing GPS and GLONASS infrastructure while enhancing their performance with aircraft based augmentations [11]. These include onboard inertial navigation systems and satellite signal integrity profiling through the comparison of signals from multiple satellites [11]. Future GNSS implementations will use other means of performance augmentation to provide the required level of accuracy for all phases of flight [11]. These may include satellite-based augmentations such as WAAS and ground-based augmentations such as LAAS[11]. The ultimate goal is to move GNSS from a supplemental means of navigation, where it provides only the accuracy and integrity components of RNP and the availability and continuity requirements are facilitated by other systems, to the sole-means of navigation, where it would meet all RNP requirements [11].

5. Closing Comments

With the push to move from terrestrially-based navigation systems to ones that are satellite-based, especially since satellite radio navigation is the primary navigation component of CNS/ATM, it is important to recognize the potential problems such systems face. The majority of the problems stem from weaknesses in satellite-based radio navigation systems: interference and ionospheric propagation effects.

Interference occurs when navigation signals are either degraded or made unavailable. It can be either intentional or unintentional. Unintentional interference can take several forms. Television stations that transmit on channel 23 within the line of sight of an aircraft can interfere with the aircraft's received GPS signal. Commercial VHF broadcasts near airports, such as taxi dispatches, have been known to have a similar effect. Problems have also occurred in select regions in which over-the-horizon military radar is in use. Intentional, malicious interference is perhaps an even larger problem, which could result in air traffic control havoc and crashes near airports[5].

Since airplane-to-satellite communications must pass in and out of the earth's atmosphere, the accuracy and availability of satellite-based radio navigation signals is extremely dependent on the state of the ionosphere. Refraction errors, scintillation, and high sunspot cycles can cause serious signal

degradation[5]. While differential GPS technologies can reduce the problems caused by these conditions, they are also vulnerable to them[5].

Before completely abandoning terrestrial radio navigation systems, serious research and development should be invested in order to find solutions to these problems. Systems that utilize satellite radio navigation, such as CNS/ATM, in conjunction with other navigation systems have the advantage of additional redundancy in measurements resulting in more accurate navigation information. However, this comes at the cost of increased software complexity in addition to increased development and certification expenses. Thus, the development of future navigation systems will need to strike the proper balance between new technologies, old technologies, and software.

Acknowledgments

Figure 1 Copyright 2001 From *The Avionics Handbook*, edited by Cary R. Spitzer. Reproduced by permission of Routledge/Taylor & Francis Group, LLC.

References

[1] Bagieu, S. "AIRBUS FANS Update and Future." Presented at the 2004 ATC Data Link Conference, IEE London, September 2004. <http://www.atnconference.com/downloadppt.php?id=251>

[2] Boeing. "Operator Benefits of Future Air navigation System." *AERO* 2 April 1998. 3 March 2004 <http://www.boeing.com/commercial/aeromagazine/aero_02/index2.html>

[3] Cassiau-HAURIE, Christophe "Airbus FANS status." Brussels: 18 June, 2003. 4 April 2004. <http://www.arinc.com/aeec/projects/users_forum/UF_Brussels_03/3-1_airbus_fans_status.pdf>

[4] Collinson, R. P. G. *Introduction to Avionics Systems*. Boston: Kluwer Academic Publishers, 2003.

[5] Corrigan, T.M., and J. F. Hartranft, L. J. Levy, K. E. Parker, J. E. Prichett, A. J. Pue, S. Pullen, and T. Thompson. *GPS Risk Assessment Study.* Final Report Under Contract 98-JHU-01 with Air Transport Association. Laurel: Johns Hopkins University, 1999.

[6] Clarke, Bill. *GPS Aviation Applications.* New York: McGraw-Hill, 1996.

[7] Curran, Jim. *Trends in Advanced Avionics*. Ames: Iowa State Press, 1992.

[8] Georgiades, Paris. *Data Link Manual: The Means for Future ATM*. International Federation of Air Traffic Controllers' Associations, 2003. Apr. 12 2004 <http://www.ifatca.org/pdffiles/datalink.pdf>

[9] Honeywell. *Advanced Air Data Inertial Reference Unit*. 3 March 2004. <http://www.cas.honeywell.com/ats/products/nav_4adirs.cfm>

[10] Howell, Jack. "Address by the Director of the Air Navigation Bureau of the International Civil Aviation organization." Rio de Janeiro. 11 May 1998.

[11] International Civil Aviation Organization. "Global Air Navigation Plan for CNS/ATM Systems."2nd Edition, International Civil Aviation Organization, 2002. <http://www.ibac.org/Library/ElectF/CNS_ATM/9750_2ed.pdf>

[12] *Industry Projects.* Aerospace Technology. 3 March 2004 <http://www.aerospace-technology.com/projects/index.html>

[13] Kayton, Myron. "Navigation Systems." *The Avionics Handbook.* Boca Raton: CRC Press, 2001.

[14] Kayton, Myron, and Walter R. Fried. *Avionics Navigation Systems.* New York: John Wiley and Sons, Incorporated, 1997.

[15] Potocki, Peter. "Meeting the Challenges: A330/A340 A319/A320/A321 AIM-FANS." Airbus, May 11, 1999. <http://www.boeing.com/commercial/caft/cwg/ats_dl/Airbus1.ppt>

[16] Rogers, Robert M. *Applied Mathematics in Integrated Navigation Systems.* Gainesville: American Institute of Aeronautics, Inc., 2000.

[17] *Satellite Navigation Product Teams.* Federal Aviation Administration. 3 Feb. 2004 <http://gps.faa.gov/index.htm>.

[18] Signargout, Luc. “AIM-FANS A Flexible Approach to CNS/ATM” *FAST* June 1995. 3 March 2004 <http://www.airbus.com/pdf/customer/fast17/p27to31.pdf>
[19] See http://www.wsrcc.com/wolfgang/gps/accuracy.html
[20] See http://users.erols.com/dlwilson/gpsvert.htm

Part II

Avionics Components

CHAPTER 4

Flight Control Systems

William S. Greenwell J. Graham Alsbrooks

The *flight control system* (FCS) in an aircraft provides the basic interface between the pilot and the aircraft's control surfaces. This interface is the pilot's most direct means of flying the aircraft and is also used by higher-level control systems such as the autopilot and flight management system. Traditionally, FCSs have not been avionics systems but rather mechanical or hydraulic systems. Since the 1980s, however, digital flight control systems have become nearly ubiquitous in modern commercial aircraft, most notably in the Airbus A320/330/340 and Boeing 777 jetliners. These systems, in which pilot commands are processed by computers and relayed to control surfaces via electrical cables, offer significant benefits over their traditional counterparts. Digital flight control systems carry lower production and maintenance costs, alleviate pilot workload by automatically handling certain aspects of flight, and can improve flight safety and passenger comfort by limiting the pilot's ability to overstress the aircraft. Serious technical and philosophical challenges accompany these benefits, however, such as assuring the dependability of the computer systems required for digital flight control and deciding to what extent the pilot should have authority over the automation.

1. Flight Control Background

In the 100 years since the Wright brothers made their historic flight at Kittyhawk, North Carolina the development of aircraft and aviation systems has accelerated at a phenomenal rate. Modern flight control systems have all but removed the physical connection between the pilot and the aircraft. Space once occupied by rods and pulleys is now occupied by sophisticated digital systems that perform and enhance the same functions. Flight control has been, and continues to be, one of the most critical design considerations for the development of commercial and military aircraft. During the course of this century, advances in flight control technology will drive future aircraft designs, airline economy, and controversy over where the divide between the pilot and the automation should lie.

1.1. Traditional Flight Control Systems

Mechanical FCS

The earliest ventures into manned flight were constrained to either tethered balloon rides or short hops in a glider. The necessity of formal flight control systems was not realized until it was demonstrated that extended flight times over substantial distances were feasible. The experiments of Otto Lilienthal in the 1880s through 1890s showed that limited flight control was possible through a process of weight shifting. Lilienthal discovered that by simply changing the position of his body relative to the aircraft's center of gravity he could affect its motion in any direction, much like hang gliders do today. Building on the dis-

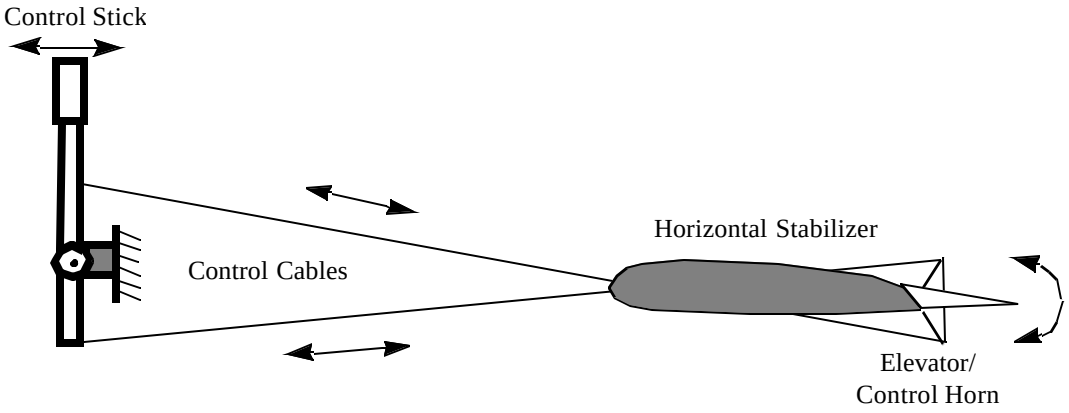

Figure 1. Mechanical Flight Control System

coveries of Lilienthal, in the mid 1890s Octave Chanute began development of what we would now call a *mechanical flight control system* for use on gliders of his own design. He worked closely with the Wright brothers and was largely responsible for many of the control systems innovations present on their 1903 Flyer.

Mechanical systems are characterized by a physical linkage between the pilot and control surfaces, as shown in Figure 1. The pilot's control inputs are transferred to the control surfaces via a series of cables and/or pushrods. This type of FCS proved to be very effective in lightweight and relatively slow moving aircraft because they were inexpensive to build, simple to maintain, and provided the best control surface feedback of any FCS. However, mechanical systems tend to be very sensitive to temperature and are prone to accelerated wear compared to the alternative methods discussed below. Also, as designers began to build bigger and faster aircraft, they discovered that the increased aerodynamic forces incident on the control surfaces were simply too great for pilots to counter. Engineers had to develop a system to augment the pilot's commands[9].

Hydraulic FCS

The first such augmented control system, known as a *boosted FCS*, appeared in WWII era aircraft. As Figure 2 depicts, the boosted system retained the physical coupling between the cockpit and control surfaces with the addition of hydraulic spool valves and ram cylinders tied in parallel into the input controls and actuators. This system retained many of the benefits of a mechanical system such as reversibility for tactile feedback (described in Section 2.2), but suffered from increased weight. As designers began building fly-

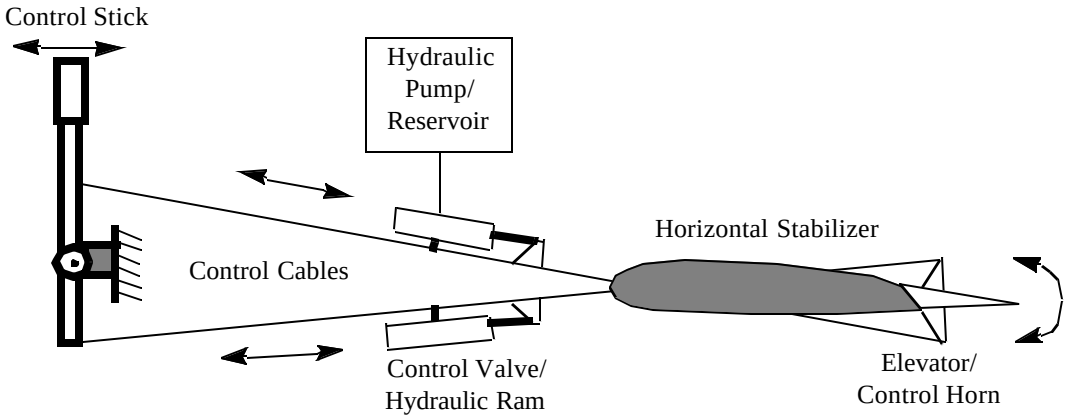

Figure 2. Boosted Flight Control System

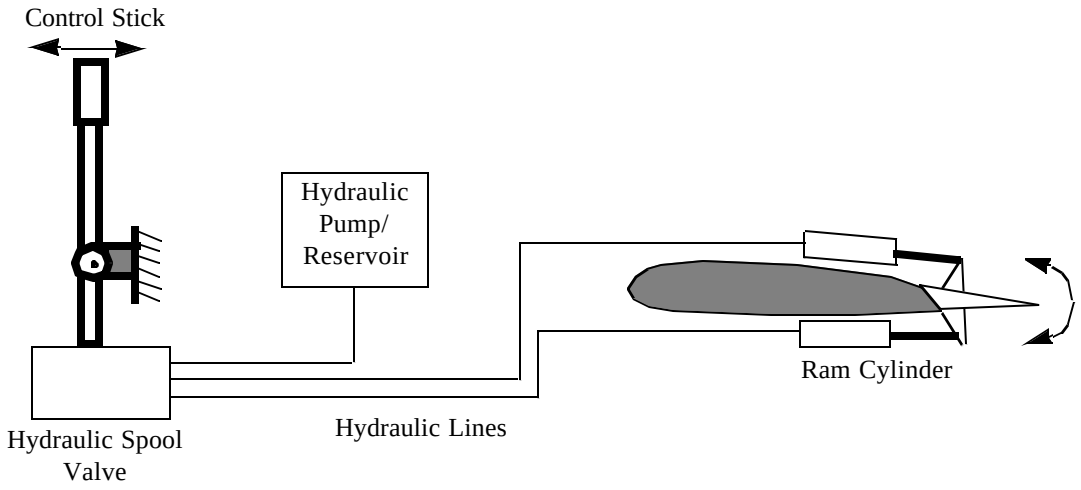

Figure 3. Hydraulic Flight Control System

ing machines capable of flirting with the sound barrier and/or carrying many hundreds of passengers, the boosted flight control system also proved inadequate to the task[12].

Hydraulic flight control systems such as the one shown in Figure 3 were introduced to cater to aircraft pushing the limits of control surface loading. As the name implies, the hydraulic system relies primarily on a series of lines and hoses feeding hydraulic actuators from a pump assembly and fluid reservoir. In this form of FCS, the pilot is no longer physically connected to the control surfaces. Rather, the pilot modulates the fluid pressure within the lines via a spool valve connected to the control yoke or stick. This system has the advantage of being able to generate massive forces to affect the attitude of any aircraft regardless of size or speed. Hydraulic systems also afford designers greater flexibility in line routing and actuator placement because there are no requirements for a "line of sight" coupling between the cockpit and control surfaces[12].

1.2. Digital Flight Control Systems

Digital flight control systems are made possible by replacing the traditional mechanical or hydraulic linkages between the pilot's controls and aircraft control surfaces with electrical signal connections, a technology known as *fly-by-wire flight control*. In the realm of commercial aircraft, digital fly-by-wire FCSs were pioneered by Airbus and debuted in 1984 with the A320 passenger jet. Each subsequent Airbus model has featured fly-by-wire control, and the current A380 jetliner is no exception. In 1990, Boeing responded with the 777 jetliner featuring its own digital FCS, and the 787, Boeing's next commercial aircraft, will use a similar system. With both of the major commercial aircraft manufacturers committed to fly-by-wire flight control for the foreseeable future, a careful examination of the merits and drawbacks of this technology is warranted.

In a fly-by-wire FCS, pilot commands are relayed to control surfaces by electrical signals. As the pilot deflects the control column or moves the rudder pedals, his commands are translated into digital signals that pass through wiring to actuators affixed to each control surface. The actuators then deflect the appropriate control surfaces to execute the pilot's commands according to the signals they received. If tactile feedback is provided to the pilot, sensors mounted on the exterior of the aircraft pass aerodynamic data such as airspeed back to the flight deck, where dampers or servo motors attached to the control column then simulate the aerodynamic forces the pilot would feel through a traditional FCS.

Fly-by-wire systems offer significant advantages over mechanical and hydraulic FCSs. From the design perspective, the fact that fly-by-wire systems use copper wiring to convey pilot commands to control surfaces means that engineers are free to route the wires through the aircraft wherever they choose

without increasing cost or degrading the performance of the controls. For airlines, the reduced weight of a fly-by-wire FCS translates into lower operational costs and higher profit margins.

Digital flight control systems offer additional design benefits. In a typical digital flight control system, control column deflections are measured by analog sensors and then converted into digital signals by an analog-to-digital converter. These digital signals represent the *control objectives*, which are processed by a flight computer and converted into surface actuator signals. The actuator signals travel through the wiring in the aircraft to the control surfaces, where they are converted into analog signals that drive the surface actuators. Representing control signals digitally allows designers to build simpler interfaces to the autopilot (described in Chapter 5) and the flight management system (FMS, described in Chapter 6). The autopilot and FMS are already digital, and digital representation of control signals eliminates the need for separate control infrastructures for the pilot and copilot. Digital flight control systems also facilitate the introduction of computer-based technology to monitor pilot input to ensure that the aircraft does not stall or otherwise depart from its flight envelope.

Considerations in Digital Flight Control

Digital flight control and fly-by-wire technology involve a variety of engineering considerations. While they provide benefits in terms of line maintainability, digital FCSs introduce new concerns, including the use of software in a flight control system, the susceptibility of electrical wiring to *electromagnetic interference* (EMI), and the difficulty in modeling the possible flight conditions an FCS might encounter. Each of these considerations is discussed below.

Line Maintainability

From a maintenance perspective, fly-by-wire systems offer three main benefits [1]. As previously noted, digital fly-by-wire technology weighs significantly less than alternatives. Compared to conventional hydraulic or mechanical systems that utilize long fluid-filled tubes or heavy rods/cables, a fly-by-wire control system requires only very lightweight wires to connect the pilot to the control actuators. In fly-by-wire systems the hydraulic pump/reservoir/actuator assembly or electromotor actuator is self-contained and, as with other forms of FCS, usually triplicated. The weight savings afforded by fly-by-wire technology are translated directly into operating cost savings and reduced fatigue on the airframe.

Next, digital fly-by-wire systems are often able to self-diagnose a problem. This feature enables the air crew to accurately troubleshoot a problem in flight and take the appropriate corrective action. Additionally, maintenance crews are able to quickly get to the root of a problem without having to perform extensive diagnostics to trace a malfunction. The result is much safer flights and greatly reduced turn-around times when a problem is identified on the ground.

Finally, with the reduction of mechanical systems onboard the aircraft, maintenance tasks are further simplified. Regular maintenance items such as cable tension adjustment and setting trim tabs are all but eliminated.

Software Reliability

Digital flight control systems mark the entry of software-based avionics into one of the most safety-critical aspects of flight. Ensuring that the software is safe is an immense challenge for avionics system developers because it is difficult to predict where faults in such systems might arise and what their likely consequences will be. Moreover, large software systems often incorporate several subsystems that interact with each other, which could lead to unanticipated system behavior if one subsystem fails and interacts with others in an undesired way.

Hardware designers typically address the problem of system reliability by replicating critical components. If one component in an array of redundant components fails, the others will likely remain operational and thus mask the failure. The notion that reliability may be enhanced through redundancy relies on the assumption that failures are independent—that is, the failure of one component in the array does not

affect the likelihood that another will fail. Because most hardware failures are due to physically-induced *degradation faults* (e.g., lightning strikes, circuit damage, and normal aging), this assumption largely holds. It does not necessarily hold, however, when failures are due to *design faults*—defects introduced when the systems were designed—because the redundant components share the same design and with it the same set of design defects.

Software is intangible, so it cannot exhibit degradation faults. Rather, software failures are necessarily due to design faults that cannot be masked through simple replication due to the lack of failure independence. *Design diversity* is a popular technique that attempts to overcome this difficulty by employing arrays of redundant components, each with a dissimilar design or implementation. Airbus and Boeing both use design diversity in their flight control systems but in different ways.

In some of its systems, Airbus employs a design diversity technique called multi-version programming or *N*-version programming[3]. In multi-version programming, several system implementations are prepared from the same set of requirements by different developers under the presumption that the designs prepared by each developer will be independent—that is, the probability that one implementation will fail on a particular set of inputs given that another implementation has failed on those inputs is equal to the probably of the implementation's failing alone. The various implementations are then assembled into a classical redundancy architecture in which they are run in parallel on the same inputs and their outputs are passed into a voter to check agreement. If a design fault is activated in one of the implementations, then, according to the theory, it is unlikely that the other implementations will also possess the fault and should continue to function. Clearly, the assurance that can be placed on multi-version programming rests on the assumption of design independence, and evidence exists that this assumption does not hold for all types of software systems[7].

The Boeing 777 FCS was not developed through multi-version programming but rather by employing diversity in the microprocessor architecture. Boeing compiled the software for the 777 FCS for multiple machine architectures and runs each version in tandem during system operation. This approach allows the 777 FCS to tolerate design faults in a specific microprocessor well as those introduced during compilation. It does not, however, provide any resilience to faults resulting from errors in the common source code from which the versions were built[13].

Electromagnetic Interference

All electrical systems are vulnerable to electromagnetic interference (EMI), which at minimum can cause transient failures and, if more severe, damage wiring and circuitry. For aircraft, the primary source of potentially disruptive EMI is lightning, and, because aircraft must be able to operate in thunderstorms, the avionics systems onboard are required to be built to withstand a direct lightning strike on the aircraft. Lightning is not the only source, however, and wires themselves may sometimes generate interference that might induce current in nearby cabling. Fly-by-wire flight control systems are much more sensitive to EMI than mechanical and hydraulic systems because of their reliance on electrical connections between the control column and the control surfaces. In order to prevent EMI from causing uncommanded control surface actuations, the cabling between the cockpit and the control surfaces must be electrically shielded, adding to the cost and weight of fly-by-wire systems. In addition to lightning and emissions from other wires, however, malicious individuals might attempt to detonate devices that generate EMI inside the cabin, which, unlike explosives, are not detected by conventional airport security measures. Thus, fly-by-wire systems must be shielded from EMI originating both from inside and outside of the aircraft.

Flight Condition Modeling

To ensure the safety of a flight control system, one must show that it will perform according to its specification under a variety of possible flight conditions. With traditional FCS, the mathematical functions modeling the behavior of the system are continuous, which means that the behavior of the system under one set of conditions will likely be similar to its behavior under similar conditions. When a digital FCS is used,

however, these functions become discrete and lose their continuity. Consequently, flight tests carry less meaning because there is no guarantee that system performance under the test conditions will correspond to its performance under similar, but not identical, conditions when it is operational. Thus, aircraft designers must conceive of more possible flight conditions and conduct additional ground and flight testing to validate the safety of a digital FCS.

Applications

As mentioned earlier, digital flight control systems offer numerous advantages over their mechanical and hydraulic predecessors. Two key areas in which fly-by-wire systems are employed are carefree aircraft handling and aerodynamically unstable aircraft.

Carefree Handling

Though it is implemented in different ways, carefree handling generally refers to a feature of fly-by-wire flight control that keeps the aircraft operating within its *flight envelope*, the set of aerodynamic conditions under which the aircraft can safely fly. Programming the FCS to keep the aircraft within its flight envelope allows the flight crew to concentrate on higher-level tasks. An aircraft may operate within several flight envelopes concurrently in an effort to strike a balance between performance, passenger comfort and safety.

An aircraft's *design* flight envelope is derived from several factors including aerodynamic limits, thrust/power limits and the structural limitations of the airframe. These limits are typically depicted on a flight envelope chart as a function of airspeed, altitude and load factor (the number of g-forces the aircraft is subjected to). An example flight envelope is shown in Figure 4. Exceeding the flight envelope limits might cause the aircraft to stall or incur structural damage. The software system responsible for keeping the aircraft within the flight envelope is known as the *envelope protection system*.

Also a consideration in modern airliner design is the less formally defined *comfort* envelope. The comfort envelope, delineated within the design envelope, represents the limits of passenger comfort in terms of load factor. Maintaining the comfort level is one feature of a stability augmentation system, described below.

Envelope Protection

Envelope protection is a feature of the FCS that deals with situations where an aircraft is instructed to approach or exceed the limits of its flight envelope. Envelope protection can be implemented in a number of ways. The "hard limits" approach stops the pilot from placing the aircraft in a flight condition that might jeopardize flight safety. The "soft limits" technique provides the pilot with aural, visual and sometimes tactile cues (for example, stick-shakers and progressive control resistance, which are discussed in Section2) that the aircraft is in danger of exceeding its flight envelope. The added protection afforded by this feature frees the pilot somewhat to focus on higher-level tasks such as navigation and interaction with air traffic control because he does not have to worry so much about physically flying the aircraft.

Stability Augmentation

Carefree handling is also often used to optimize the handling of an aircraft in a manner that is transparent to the pilot. For example, airliners typically operate between 0.85 and 1.15 g-forces in the interest of passenger comfort. If turbulence is encountered during flight, however, the pilot usually cannot react quickly enough to counteract the rapidly shifting aerodynamic forces, resulting in spilled passenger beverages and stained clothing. The fly-by-wire system is often able to detect turbulence via the air and inertial data sensors and can subsequently smooth the ride by making hundreds of adjustments per second to the flight control surfaces. This smoothing is an example of *stability augmentation*. In addition to passenger comfort, the capability of digital flight control to rapidly detect and correct for dynamic flight conditions enables it to

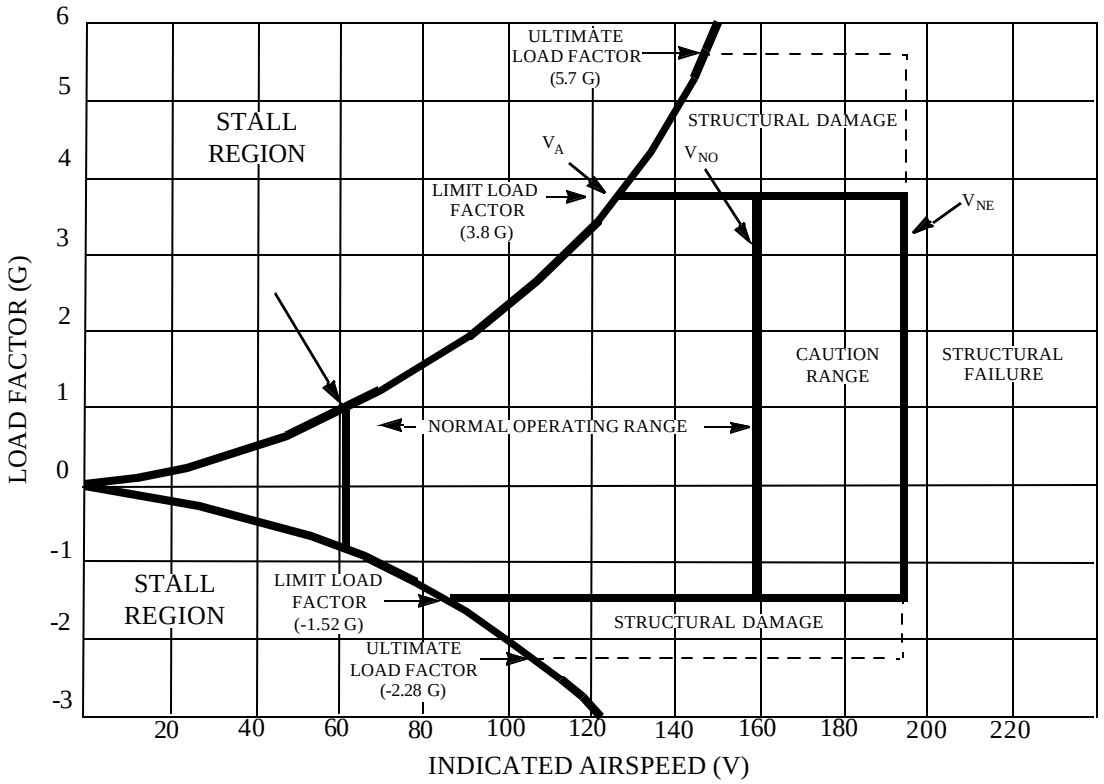

Figure 4. Typical Flight Envelope[6]

address a host of issues including fuel loading, aeroelasticity, and other changing flight conditions without further taxing the pilot.

Aerodynamically Unstable Aircraft

Fly-by-wire and digital flight control are the enabling technologies for developing aerodynamically unstable aircraft. While no present commercial aircraft are unstable, unstable flight is just beyond the horizon for commercial aviation and already in widespread use in military aircraft.

Aerodynamically unstable aircraft can achieve better fuel economy than stable aircraft and, for military applications, are capable of performing maneuvers that would otherwise be impossible. Unfortunately, humans are incapable of directly flying unstable aircraft because they cannot adjust quickly enough to the continuously changing aerodynamic forces. Instead, as in carefree handling, digital flight control systems are situated between the pilot and the control surfaces and stabilize the aircraft automatically, making it seem to the pilot as though the aircraft were stable.

1.3. FCS Architecture

At the highest level, the fly-by-wire FCS takes digitized input from the pilot, filters the commands at the processing core by applying the relevant *flight control laws* (FCLs), and commands the actuators on the corresponding control surfaces to move. The relevant FCLs are determined by referencing the inertial and air data sensor arrays. This loop is repeated many times per second and is depicted in Figure5.

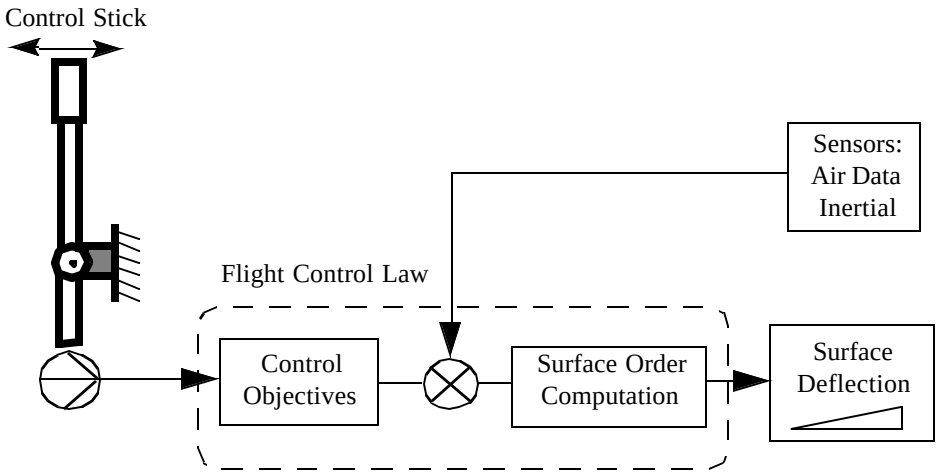

Figure 5. Flight Control System Architecture

1.4. Flight Control Laws

A fly-by-wire aircraft's FCLs are the rules that govern the *handling qualities*, responses to pilot input, of the airplane in different phases of flight. FCLs impart "character" to the aircraft by specifying deflection rates and maximum deflection values for each control surface. FCLs are thus tailored to meet a Handling Qualities Model (HQM), which is derived from the pilot's expectation of how the aircraft should behave in response to control inputs. Some FCLs even specify what the aircraft can and cannot do through rate/deflection limiting and by disallowing certain functions during inappropriate flight conditions (e.g., lowering the gear at 300 knots or maneuvering outside of design limits). It is through the flight control laws that "hard" envelope protection systems limit the pilot's authority over the aircraft.

1.5. Looking Ahead

Neural Networks

Neural networks promise to offer an approach to the mediation of some of the hazards associated with flight control failure or structural damage to the airplane in a technology is referred to as *adaptive control*. Neural network software is distinguished by its ability to "learn" by comparing real-time sensors signal patterns to those it would expect to receive from an aircraft in good health and then making appropriate responses. For example, if structural icing or battle damage were to occur to such a degree as to render a control surface like the rudder inoperable, the neural network software would compensate via a dynamic mixing of surface deflections from other control surfaces such as the elevators, ailerons and the other still functioning components [5]. These changes would theoretically be transparent to the pilot. Each control input to the crippled aircraft would result in an anticipated attitude change, though it would be effected in a completely different way. Adaptive control is presently implemented in experimental form using conventional software.

Fiber Optics

In an effort to counteract the previously discussed issues with fly-by-wire systems and EMI, aircraft designers are investigating the feasibility of using fiber-optic cabling to replace the copper. Dubbed fly-by-light, this variation on fly-by-wire offers three distinct advantages over its copper counterpart. First, fiber optic cable and the transceivers that use it are extremely lightweight. Second, fiber optic systems have higher bandwidth than wire systems, clearing the way for improved flight control systems, super-stability

augmentation and a host of other new technologies still under either design or consideration[15]. Finally, fiber-optic cables are not susceptible to EMI.

Wireless Technology

Some researchers have proposed the possibility of incorporating wireless control devices into aircraft to further reduce the weight and complexity of wiring harnesses by eliminating them altogether. Wireless Flight Control Systems (WFCS) involve the insertion of spread-spectrum RF data links into the communication paths between the flight-control computers (FCCs) and the actuators for the flight-control surfaces at the ailerons, rudder, and elevators[8]. Essentially, a radio transmitter in the cockpit relays signals via the airwaves to the control actuators at various locations throughout the airframe. Needless to say, this technology will have a number of serious hurdles to clear before gaining acceptance in the aviation industry.

2. Survey of Fly-by-Wire Flight Control Systems

Fly-by-wire flight control systems are now the standard for modern commercial aircraft. Both Airbus and Boeing are committed to fly-by-wire FCS for their current and upcoming jetliners, and each company has invested billions of dollars in research for development and certification of its own systems. Because Airbus and Boeing comprise the bulk of the commercial aviation market, this section examines the architecture and features of the FCS onboard the Airbus A320 and the Boeing 777 jetliners, both of which use fly-by-wire technology.

2.1. Airbus A320

The Airbus A320, which was launched in 1984 and entered service in 1988, is the first commercial jetliner to employ a fly-by-wire flight control system[1]. Airbus has since refined the flight control laws and systems integration of its fly-by-wire design to accommodate the greater size and range of subsequent families of aircraft including the A330, A340, and upcoming A380 jetliners. Aside from these improvements, Airbus's fly-by-wire concept has remained fundamentally unchanged, allowing it to reap significant returns from its research investment in the A320 FCS architecture 20 years ago. This section examines the A320 FCS architecture, its flight control laws, and Airbus's flight envelope protection system, which is designed to prevent the aircraft from exceeding its design limitations.

FCS Architecture

The A320 FCS architecture is similar to that shown in Figure 5. Pilots command elevator and aileron deflections through a sidestick, which replaces the control yoke found on traditional aircraft, and rudder deflections through the rudder pedals. The sidestick and pedals translate pilot commands into electrical impulses, which, combined with inputs from the autopilot system, form the control objectives. The FCS computes the control surface deflections needed to achieve the objectives using control laws and sends electrical signals over copper wires to servo motors attached to the control surfaces to perform surface deflection. Depending on various aerodynamic factors, the initial aircraft response to the deflections might be insufficient, and so sensors mounted on the aircraft's exterior measure the actual changes in pitch, roll, and yaw achieved by the deflections. These measurements are fed back into the control laws and adjustments are made to the surface deflections if necessary.

Subsequent families of Airbus aircraft have improved on this architecture to accommodate larger airframes and to enhance passenger comfort. On the A330 and A340 systems, for example, Airbus applied filters to the pilot and sensor inputs to the flight control laws in order to dampen vibrations induced by the larger and thus more flexible airframes. Airbus also added accelerometers to detect turbulence and added a "comfort in turbulence" function between the surface order computation and surface deflection to automatically smooth out the effects of turbulence.

In case of an unrecoverable failure of the fly-by-wire FCS, Airbus provides a backup hydraulic system through which the pilot may temporarily fly the aircraft. The backup system provides pitch control through the pitch trim wheel, which controls the horizontal stabilizer, and lateral control through the rudder pedals. No longitudinal (roll) control is provided by the backup system.

Flight Control Laws

The FCS onboard the A320 and subsequent aircraft are governed by four sets of control laws: the Normal Law, the Alternate Law, the Abnormal Alternate Law, and the Direct Law[11]. The aircraft operates under the Normal Law in nominal conditions and should only transition to one of the other control laws when an anomaly is detected.

Normal Law

Normal Law is the nominal control law used by Airbus flight control systems and is active when no failures have been detected or after a single failure has occurred. The Normal Law comprises three operating modes: ground mode, flight mode, and flare mode. Ground mode is active when the aircraft is on the ground; it deactivates shortly after takeoff and reactivates after touchdown. Ground mode assists the pilot in performing the pre-flight checklist by providing a direct proportional relationship between sidestick deflection and control surface deflection, allowing the pilot to verify that his controls are working properly. When it is reactivated on touchdown, ground mode also resets the stabilizer trim to zero to remove any nose-up trim that might have been applied during landing.

Flight mode is the primary operating mode of the Normal Law and is active from shortly after takeoff until just before touchdown. In flight mode, the FCS automatically handles pitch trim, turn coordination, and yaw damping. Flight mode also maintains a 1 g-force load in pitch when the sidestick is neutral and the wings are level, and it ensures that a given sidestick deflection always commands the same roll rate response independent of airspeed.

The pitch law that is active in flight mode differs significantly from the pitch behavior in traditional aircraft and those developed by Airbus's competitors. In a traditional aircraft employing a yoke for attitude control, subject to aerodynamic factors such as airspeed, deflecting the yoke forward or aft commands a certain rate of pitch and deflecting it left or right commands a certain rate of roll. If the pilot releases the yoke, the aircraft eventually returns to a neutral pitch configuration. Under Airbus's Normal pitch law, however, when the pilot applies forward or aft deflection to the sidestick, rather than commanding a particular rate of pitch, the deflection instead translates into a particular load factor demand, which in turn alters the rate of pitch. With no forward or aft deflection, the aircraft maintains a 1 g-force load in pitch. Forward deflection commands a smaller load, which causes the nose to pitch down, and aft deflection commands a larger load, which pitches the nose up. The difference between this and the traditional approach is that if the pilot maintains the deflection, the aircraft will accelerate the rate of pitch in order to maintain the desired load. For example, with a traditional yoke, if the pilot wanted to enter a 12° climb, he would pull back on the yoke enough to achieve the desired pitch rate and then hold the yoke in that position for the duration of the climb[1]. With a sidestick, however, the pilot would instead pull the stick back to initiate the climb but then release it once the desired pitch angle had been achieved. With the stick in the neutral position, the aircraft would remain in the 12° climb in order to maintain a 1 g-force load. When the pilot was ready to exit the climb, he would push the sidestick forward to reduce the rate of pitch and then let go once the aircraft had returned to level flight.

Flare mode activates when the aircraft is 50 feet above ground level (AGL) during the final stages of landing. When flare mode activates, the system remembers the pitch attitude at 50' AGL and begins to progressively reduce the pitch, forcing the pilot to flare the aircraft. Flare mode remains active until touch-

1. Technically, the pilot could choose to trim the aircraft to hold the desired pitch angle and alleviate the need to maintain control pressure on the yoke, but this is equivalent to holding the yoke for the purposes of this example.

down, when ground mode assumes control of the aircraft. In the event of a go-around, the system will transition back to flight mode at 50’ AGL using the pitch value it remembered earlier.

The Normal Law includes numerous protection mechanisms designed to prevent the aircraft from exceeding its design tolerances as part of the flight envelope protection system. Under the Normal Law, the FCS provides load factor limitation, attitude protection, high angle of attack protection, high speed protection, and low energy warnings. Load factor limitation prevents the pilot from overstressing the aircraft even if full sidestick deflections are applied. Attitude protection limits the pitch angle to 30° nose-up and 15° nose-down pitch and limits the bank angle to 67° left or right bank. In addition, bank angles in excess of 33° require continuous sidestick deflection, and the aircraft will revert to 33° of bank if the input is released. High angle of attack protection prevents pilot-induced stalls by preventing the angle of attack from ever exceeding the critical angle. High speed protection prevents the aircraft from exceeding its design speed limit by introducing nose-up pitch to slow the aircraft down that cannot be overridden by the pilot. Lastly, if the pilot must abort a landing approach and climb back up to a safe altitude, the low energy warning alerts him if additional thrust is needed to sustain the climb.

Alternate Law

If multiple flight computers fail, the FCS transitions to the Alternate Law. The Alternate Law only comprises two modes of operation: ground mode and flight mode. Ground mode is identical to the ground mode for the Normal Law. The Alternate Law provides automatic pitch trim and yaw damping but does not provide turn coordination. Pitch and bank protections are relaxed, and roll rate is no longer independent of airspeed. Since no flare mode is provided, the pitch Alternate Law reverts to the Direct Law, described below, when the landing gear is extended to provide greater feel for flare and landing.

Under the Alternate Law, all of the envelope protections except for load factor limiting are off or reduced. Load factor limiting is similar to that under the Normal Law. The high angle of attack protection is replaced with a low speed stability function that introduces a progressive nose-down pitch as the angle of attack approaches the critical angle to prevent further speed decay. This nose-down pitch may be overridden by the pilot, and the aircraft can be stalled under the Alternate Law. Aural stall warnings are provided and bars denoting stall regions are displayed on the airspeed scale to reduce the likelihood of pilot-induced stalls. High speed protection is modified to allow the pilot to override the nose-up pitch command, and under certain failure modes it might be lost altogether. Bank angle protection is not provided.

Abnormal Alternate Law

The Abnormal Alternate Law is activated if the system detects that the aircraft is in an unusual attitude and is intended to allow a nominal attitude to be restored. Under the Abnormal Alternate Law, pitch is governed by the Alternate Law and roll is governed by the Direct Law. Yaw is controlled mechanically. No automatic trim or protections other than load factor limitation are provided. Upon recovering from the unusual attitude, pitch and yaw continue to be governed by the Alternate Law and roll by the Direct Law for the remainder of the flight.

Direct Law

The FCS transitions to Direct Law after suffering certain combinations of multiple failures. Under the Direct Law, pilot commands are transmitted unmodified to the control surfaces. No automatic trimming, yaw damping, or turn coordination are provided, and control surface sensitivity is dependent on airspeed. If the flight controls transition to the Alternate Law, the Direct Law becomes active when the landing gear is extended. No protections are available under the Direct Law; however, aural stall and overspeed warnings are provided.

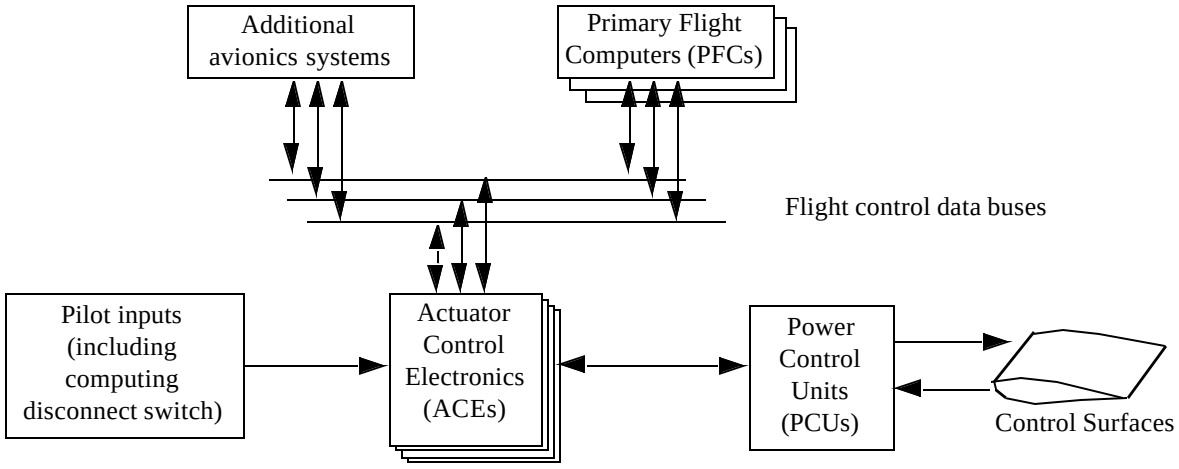

Figure 6. Boeing 777 Primary FCS Overview (excerpted from Yeh [13])

2.2. Boeing 777

In 1990, Boeing began efforts to launch its first fully fly-by-wire commercial transport aircraft in response to the previously discussed European Airbus 3XX series of jetliners. The scope of the project was massive, requiring radically complex new designs of computer hardware and over 2.6 million lines of avionics software code—six times that of any of the existing offerings in the manufacturer's fleet[12]. Boeing anticipated that the 777 would become the world's most technologically advanced and safest airliner.

FCS Architecture

The major components of the 777's fly-by-wire system consist of the Actuator Control Electronics (ACEs), the Primary Flight Computers (PFCs), and the Power Control Units (PCUs) as shown in Figure 6.

During normal operation, a pilot's control inputs are read by multiple transducers on the input devices before being sent to the ACE where they undergo an analog to digital conversion. The ACE in turn passes the digital signal to the PFC via a triply redundant bus where the control input is compared against environmental sensor data and the relevant FCLs before computing the optimal control surface deflection. The digital "solution" is sent back to the ACE for conversion back to an analog signal and transmission to the appropriate actuator.

Each ACE onboard the 777 is a quadruply-redundant system that serves as the backbone of the fly-by-wire system. The ACE bridges the gap between the analog and digital domains to interface between the air crew, flight surfaces, and flight computers. In addition to controlling flight surface deflection via the Power Control Units, the ACEs also control the tactile feedback system.

The PFC is the heart of the 777 fly-by-wire system, responsible for calculating control surface deflections based on pilot inputs, sensor data and relevant flight control laws. The PFC also handles the stability augmentation and automatic trimming while working in close concert with many of the higher-level computer systems such as the flight management system and autopilot to effect a high degree of automation in the cockpit.

Control Modes

Each flight-critical system aboard the 777 features triple redundancy, including the computing system, airplane electrical power, hydraulic power, and communication paths. Should the aircraft encounter a condition in which a failure or multiple failures of the above systems reduces overall functionality, the pilot may switch to an alternate control mode in order to land the aircraft safely[10].

Normal Mode

In the absence of failure, the aircraft operates in the "Normal Control" mode in which the computer takes an active role in augmenting and supplementing pilot inputs. This mode offers all of the available benefits of digital fly-by-wire systems, such as stability control and envelope protection, to the flight crew and passengers.

Secondary Mode

In the unlikely event of a sensor array failure resulting in inadequate or incorrect inertial or air data, the PFC automatically switches aircraft control to the secondary mode. This mode implements modified "open loop" FCLs that continue to augment pilot inputs but reduce the flight automation such as autopilot and envelope protection features as they are heavily reliant on environmental data.

Direct Mode

It is worthy of note that in accordance with Boeing design philosophy of the pilot being the final authority, the pilot can disable the PFC and enter Direct Control mode via a switch in the cockpit should he suspect a fault in the PFC. In Direct Control mode, the digital processing portion of the fly-by-wire system is completely removed from the loop. The pilot's inputs are transmitted from the input device transducers to the ACE where they are subject to only simple analog control laws before being sent directly to the control surface.

Flight Envelope Protection

Boeing's design methodology in the development of the 777 FCS has centered on keeping the responsibility for aircraft safety resolutely in the hands of the pilot. As a result, the warning systems are passive in nature—utilizing aural, visual and tactile warnings. While the 777 takes measures to ensure passenger comfort and avoid structural damage through the aforementioned techniques, ultimately the pilot has the option to exceed the design limitations of the aircraft should the situation dictate the need.

Tactile Feedback

The 777, being a fully fly-by-wire aircraft, has no physical connections between pilot and control surfaces. Consequently, the pilot is unable to experience any of the tactile cues to the flight condition of the plane. Boeing engineers have attempted to solve this problem through the implementation of a feedback system designed to faithfully replicate these sensations via servo motors controlled by the ACEs. To provide a sense of speed, the 777 increases resistance to yoke inputs in 3 lb. increments per 10 knots[13]. Using this system, the aircraft is able to communicate to the pilot the increased loading on the airframe just as a mechanical system would. As the loading increases, so does the yoke resistance, culminating in a form of "soft" envelope protection. Similarly, if the pilot attempts to configure the aircraft in a manner that might induce a stall, the ACEs react by sending a signal to shake the controls in a similar fashion to what might be experienced in a traditional aircraft in the moments prior to stalling.

Backdrive

With autopilot engaged, the pilot is free to focus his or her attention on other cockpit tasks with assurance that the computer is in control of the aircraft due to a feature known as backdrive. Using the same feedback system responsible for the tactile feedback, the autopilot is able to affect the control devices via the ACE as if the pilot had actually moved them. This "visual feedback" system provides the pilot with a greater sense of what the aircraft is doing in an effort to reduce disorientation and increase situational awareness.

3. Flight Control Design Philosophy

The advent of digital fly-by-wire flight control has brought with it new philosophical design questions for aircraft manufacturers. By situating a digital control system between the pilot and the aircraft's control surfaces, it is possible for FCS engineers to almost completely abstract the interface presented to the pilot from the actual underpinnings of the vehicle. Such abstraction is critical to the success of aerodynamically unstable aircraft because the FCS must handle the complexities of unstable flight itself and present an aerodynamically stable interface to the pilot in order for him to be able to fly the aircraft. Taking this abstraction to the next level, a neural-networks-based technology dubbed "Adaptive Reconfigurable Control" being developed by Barron Associates, Inc. would automatically detect malfunctioning control surfaces and emulate their behavior through other available surfaces without any disruption to the pilot[5]. Through abstraction, the manner in which the pilot flies the aircraft may bear little correspondence to the manner in which the aircraft is actually flown.

Modern commercial aircraft do not yet exhibit this level of abstraction in the pilot's interface in part because they are aerodynamically stable but also because the assurance challenges and human factors issues surrounding such a sophisticated level of abstraction are still too great. More basic levels of abstraction are present in production systems, however, through technologies such as flight envelope protection that seek to prevent the pilot from stalling or overstressing the aircraft and handling quality models that smooth out turbulence and make aerodynamically distinct aircraft behave alike. Airbus and Boeing hold different philosophies regarding how the pilot interface should be abstracted. These philosophies are best exemplified by examining the companies' distinct approaches to pilot-automation interaction and flight envelope protection.

Pilot & FCS Interaction

Airbus and Boeing hold different opinions as to what the roles of the pilot and the automation in the cockpit should be. The approach employed by Airbus in its A320, A330, A340, and A380 jetliners is that automation should handle as much of the flight as possible and that the pilot's interaction with the aircraft should only regard those matters to which the automation itself cannot attend. One example of this philosophy is the sidestick Airbus uses for attitude control and its nontraditional load-factor-based pitch law. Airbus provides no tactile feedback to the pilot through the sidestick, so the pilot has less of a feel for how the various aerodynamic forces are affecting the aircraft. Boeing's approach, as evidenced in the handling quality model of its 777 jetliner, is that the automation should conform to the expectations of the pilot. In the 777, the flight control system simulates the aerodynamic forces the pilot would encounter if he were using a mechanical or hydraulic flight control system. As a result, pilots learning to fly the 777 can easily adapt to the digital FCS because it behaves much like the systems to which they are already accustomed.

Flight Envelope Protection

Both the Boeing 777 and Airbus A320/330/340 families of aircraft include flight envelope protection systems. As defined earlier, a flight envelope protection system is a type of carefree handling system that monitors pilot inputs for commands that cause the aircraft to enter a stall or exceed its design airspeed or load limits. In the 777, this system functions in an advisory capacity and alerts the pilot by exerting resistance on the control column to potentially dangerous commands. The pilot may override the system by overcoming the resistance. On Airbus aircraft, the system plays a much more active role and may refuse to execute commands altogether if it judges them to be unsafe. The Airbus philosophy does not allow the pilot to override the flight envelope protection system except when operating under the Direct and Abnormal Alternate Laws and in certain circumstances under the Alternate Law.

Observations

It is unclear what impact the different design philosophies of Airbus and Boeing have on the safety of their aircraft. Although accidents have occurred in which Airbus's active flight envelope protection system was cited as a factor, Airbus insists these accidents were the result of pilot error and stands by its design. Meanwhile, although no major accidents attributable to the aircraft have occurred involving the Boeing 777, the aircraft is still relatively new and not in as widespread use as the A320/330/340 are today, making it difficult to use accident rates as a basis for comparison.

References

[1] Airbus. "Our Advantages: Fly-by-wire." <http://www.airbus.com/media/fly_by.asp>

[2] Alonso, Fernando. "FBW Evolutions." *Proc. 10th Performance and Operations Conference*, San Francisco, CA, September 1998.

[3] Avižienis, Algirdas. "The Methodology of N-Version Programming." *Software Fault Tolerance*, pp. 23-46. Michael Lyu (ed.). Chichester: John Wiley & Sons. 1995.

[4] *Aviation Week and Space Technology*. 2 Feb. 1970.

[5] Barron Associates, Inc. "Adaptive Reconfigurable Control: Self-Designing Controller." <http://www.bainet.com/htm/pr.1-1.html>

[6] Gleim, Irvin N. *Gleim's Pilot Handbook*, 6th ed. Gainesville: Gleim Publications. 2001.

[7] Knight, John C. and Nancy G. Leveson. "An Experimental Evaluation of the Assumption of Independence in Multiversion Programming." *IEEE Transactions on Software Engineering*, vol. SE-12, no. 1, pp. 96-109. January 1986.

[8] Lee, Sangman et al. "Radio-Frequency Wireless Flight-Control System." Technical brief DRC-99-08. *NASA Tech Briefs*. February 2000.

[9] Nelson, Robert C. *Flight Stability and Automatic Control.* New York: McGraw-Hill. 1989.

[10] Norris, Guy (ed.). "Boeing's Seventh Wonder." *IEEE Spectrum*, October 1995, pp 20-24

[11] Sanford, Bob. "Airbus Flight Control Laws." <http://www.airbusdriver.net/airbus_fltlaws.htm>

[12] Schmitt, Morris, Jenny. "Fly-By-Wire: A Historical and Design Perspective." SAE Inc., 1998, Appendix C.

[13] Yeh, Y.C. "Design Considerations in Boeing 777 Fly-By-Wire Computers."

[14] Yeh, Y.C. "Safety Critical Avionics for the 777 Primary Flight Control System." *IEEE*, pp 1.C.2.1-1.C.2.11. 2001.

[15] Zavala, Eddie. "Fiber Optic Experience with the Smart Actuation System on the F-18 Systems Research Aircraft." Technical memorandum NASA/TM-97-206223. National Aeronautics and Space Administration. October 1997.

CHAPTER 5

Autopilot Flight Director Systems

Kathryn A. Klein Sinem C. Göknur Matthew Bolton

The *autopilot flight director system* (AFDS) allows a digital flight guidance computer to control the aircraft instead of the pilot. The pilot can choose to use the AFDS to follow a flight plan entered into the flight management system (FMS, discussed in Chapter 6), or he can enter a specific heading, speed and altitude for the aircraft to maintain[4]. Of course, the pilot still has the option to fly the plane manually using the throttle and yoke (or stick). If the pilot chooses to use the AFDS, the digital flight guidance computer controls the aircraft's speed, pitch and yaw by sending commands to controls such as the ailerons, elevators, or rudder. The AFDS makes it possible for the aircraft to follow a flight plan without needing continuous inputs from the pilot.

Since their addition to aircraft in the mid-1980s, digital AFDSs have become an essential part of avionics systems. Along with the FMS, the AFDS has helped decrease pilot workload, which has otherwise increased as airplanes have become more advanced[11].

1. AFDS Components

A typical AFDS is composed of an *autopilot* and an *autothrottle*. The autopilot is broken down into a pitch channel and a roll/yaw channel. The pitch channel controls the angle of attack of the aircraft, which is used to adjust and maintain altitude. The roll/yaw channel controls the heading of the aircraft by adjusting the rudder and ailerons. The autothrottle component of the AFDS allows for a fixed speed to be maintained. The FMS can be linked with the AFDS to provide two-dimensional *lateral navigation* (LNAV) plus *vertical navigation* (VNAV) guidance. This allows an aircraft to fly procedural routes, which may be required by air traffic control around a busy airport[11].

A typical autopilot can have 20-30 possible modes, many of which can be used together or individually. In order to interface with all the possible modes, a *flight mode selector panel* (FMSP), also called a *mode control panel* (MCP), is essential. The MCP allows the pilot to select, change, and engage all autopilot modes [11]. The pilot can dial in a desired heading, speed, altitude and vertical speed (V/S) using the MCP, and the settings will be displayed. After the pilot has used the MCP to activate an autopilot mode, the *flight mode annunciator* (FMA) displays the active mode at the top of the primary flight display (discussed in Chapter 6, Section 3.2) to indicate what the automation is doing. Figure 1 depicts the relationship between a typical mode control panel, primary flight display (PFD), and flight management system[4].

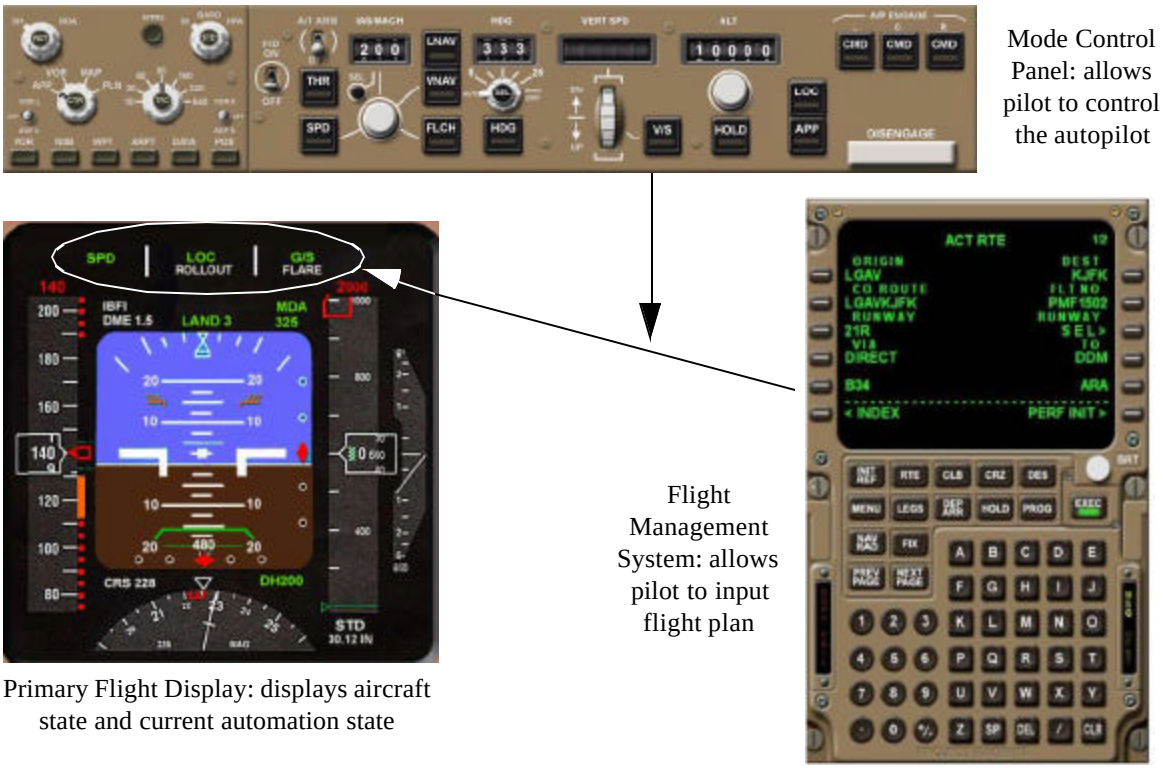

Figure 1. Mode Control Panel, Primary Flight Display, and Flight Management System relationship (images from [14])

2. Modes

As stated above, a typical AFDS can have over 20 modes, all of which the pilot must fully understand in order to properly operate the system[11]. Many of the modes can be used together or individually, creating a large matrix of possible modes the pilot must keep in mind. To complicate matters further, the modes can also be organized differently across airframes. For example, the Boeing 747-400 organizes the modes based on the available *controls*: the autothrottle, roll control, and pitch control. In the development of the MD-11, a different organizational scheme was used. The modes in the MD-11 are organized based on what is *being* controlled: speed, roll, or altitude. Regardless of how the modes are organized on the aircraft, there are three main categories of autopilot modes, based on function. A mode controls either speed, horizontal path, or vertical path.

Complicating things further, different aircraft often use different names to display the same modes. In this section, we describe some of the modes available in an MD-11 and B-747-400. A more detailed list and comparison of the AFDS modes for the MD-11 and B-747-400, created by Michael Palmer[12], is included in the form of tables that augment the relevant discussion. The comparison categorizes modes in an airframe-independent way, then lists the airframe-specific names for the modes. While the two systems have equivalent modes in many cases, they rarely have the same names for those modes. We structure our discussion in the same way that Palmer structured his tables, using general terms to describe the modes and then giving specific examples.

When a pilot wants to use the autopilot to control speed, he has two options. He can use either engine thrust (Table 1) or aircraft pitch (Table 2). If he decides to use thrust, he would normally engage the basic thrust mode, which maintains a speed specified by the FMS or the speed window on the MCP. On the MD-11 this is indicated on the FMA by the “THRUST” speed mode. On the B-747-400 it is indicated by the

Mode name & definition	Mode on MD-11	Mode on B-747-400
Basic thrust mode - Guidance for what speed to maintain is provided by either the FMS or the speed window on the MCP.	THRUST speed mode	SPD autothrottle mode
Idle thrust mode - Thrust is left at idle during a descent in which the vertical path is maintained by pitch.	IDLE THRUST speed mode	HOLD autothrottle mode
Throttles retarding mode - Throttles are in transit, coming back to an idle mode.	RETARD speed mode	IDLE autothrottle mode

Table 1: Modes That Control **Speed** by **Engine Thrust**

Mode name & definition	Mode on MD-11	Mode on B-747-400
Basic pitch mode - Guidance for what speed to maintain is provided by either the FMS or the speed window on the MCP	PITCH speed mode	VNAC SPD or FLCH SPD pitch modes
Take-off & go-around mode - Pitch is used to maintain best rate/angle of climb speed, while thrust is used to gain altitude.	PITCH speed mode in conjunction with the GO AROUND altitude mode	TO/GA pitch mode in conjunction with either the THR (derated) or the THR REF(up to reference limits) autothrottle modes

Table 2: Modes That Control **Speed** by **Aircraft Pitch**

"SPD" autothrottle mode[12]. Alternatively, if the pilot chooses to control speed by using aircraft pitch, he would use the basic pitch mode and the FMA would read "Pitch" on the MD-11 and "VNAV SPD" or "FLCH SPD" on the B-747-400.

To control the horizontal path of the aircraft, the pilot selects modes that control roll, listed in Table3[12]. The basic lateral navigation mode follows a horizontal path provided by the FMS. In the MD-11 this is displayed on the FMA with a magenta "NAV1" or "NAV2" while in the B-747-400 this same mode is called "LNAV". If the pilot instead wants to dial in a particular heading to the MCP and hold that heading, he would use the "HEADING" roll mode on the MD-11 and the "HDG SEL" and "HDG HOLD" roll modes on the B-747-400. If the pilot wants to follow a horizontal path provided by the Instrument Landing System (discussed in Chapter 3, Section 2.1), he would select the localizer tracking mode corresponding to the "LOC" roll mode on both the MD-11 and B-747-400.

Like speed, the vertical path of the aircraft can be controlled by using either engine thrust (Table 4) or pitch (Table 5). To change the vertical path using thrust, the pilot might use the climb thrust mode. This mode is used for normal altitude changes, while pitch modes are usually used to adjust speed. This is displayed as the "CLB THRUST" altitude mode on the MD-11 and by the "THR" autothrottle mode on the B-747-400. For rapid altitude changes, the maximum continuous thrust mode is often engaged. This corresponds to the "MCT THRUST" altitude mode on the MD-11 and the "THR" or "THR REF" autothrottle modes on the B-747-400.

When controlling the vertical path using aircraft pitch, the pilot has a number of options. The basic vertical navigation mode follows a vertical path provided by the FMS and is indicated by the "PROF" altitude mode on the MD-11 and by the "VNAV PTH" pitch mode on the B-747-400. If the guidance vertical

Mode name & definition	Mode on MD-11	Mode on **B-747-400**
Basic LNAV mode - Guidance for what horizontal path to follow is provided by the FMS.	NAV1 and NAV2 roll modes	LNAV roll mode
Heading select & hold mode - Guidance for what heading to turn to and hold is provided by the value in (shared) HDG/TRK window on the MCP, in HDG mode.	HEADING roll mode	HDG SEL and HDG HOLD roll modes
Track select & hold mode - Guidance for what ground track to hold is provided by the value in the (shared) HDG/TRK window on the MCP, in TRK mode.	TRACK roll mode	--
Roll attitude hold mode - Guidance for what bank angle to maintain is provided by the pilot through control inputs.	--	ATT roll mode
Localizer tracking mode - Guidance for what horizontal path to follow is provided by the ILS Localizer for review.	LOC roll mode	LOC roll mode
VOR tracking mode - Guidance for what horizontal path to follow is provided by the VOR receiver	VOR roll mode	--
Rollout mode - Guidance for keeping the aircraft on the runway centerline is provided by the ILS Localizer receiver. Both rudder and nosewheel steering are used.	ROLLOUT roll mode	ROLLOUT roll mode
TO/GA mode - The current ground track is maintained	TAKEOFF roll mode	TO/GA roll mode
Autoland mode - Guidance for what horizontal path to maintain is provided by the Autoland System.	LOC, DUAL LAND, and ROLLOUT roll modes	--

Table 3: Modes That Control **Horizontal Path** by **Aircraft Roll**

Mode name & definition	Mode on MD-11	Mode on B-747-400
Take-off & Go-around mode - Thrust is used to gain altitude while pitch is used to maintain best rate/angle of climb (see above).	GO AROUND altitude mode	THR or the THR REF autothrottle modes
Maximum continuous thrust mode - Maximum continuous thrust is provided for rapid en route altitude changes, while speed is controlled by aircraft pitch.	MCT THRUST altitude mode	THR REF autothrottle mode (when the pitch mode is ALT, V/S, G/S, or FLARE)
Climb thrust mode - Climb thrust is provided for normal en route altitude changes, while speed is controlled by aircraft pitch	CLB THRUST altitude mode	THR autothrottle mode (when pitch mode is FLCH SPD, VNAV SPD, VNAV ALT, VNAV PTH, or TO/GA)

Table 4: Modes That Control **Vertical Path** by **Engine Thrust**

Mode name & definition	Mode on MD-11	Mode on B-747-400
Basic VNAV mode - Guidance for what vertical path to follow is provided by the FMS	PROF (i.e., FMS PROFILE) altitude mode	VNAV PTH pitch mode
Altitude-constrained VNAV mode - Guidance for what vertical path to follow wants to come from the FMS, but is being constrained by the altitude dialed into the MCP altitude window.	HOLD altitude mode	VNAV ALT pitch mode
Altitude hold mode - Guidance for what altitude to maintain is provided by the altitude window on the MCP. No VNAV is pending.	HOLD altitude mode	ALT pitch mode
Vertical speed mode - Guidance for what vertical path to follow is provided by the (shared) VS/FPA window on the MCP, in V/S mode.	V/S altitude	V/S pitch mode
Vertical speed / flight path angle mode - Guidance for what vertical path to follow is provided by the (shared) VS/FPA window on the MCP, in FPA mode.	FPA altitude mode	--
Glideslope tracking mode - Guidance for what vertical path to follow is provided by the ILS Glideslope receiver.	G/S altitude mode	G/S pitch mode
Autoland mode - Guidance for what vertical path to follow, including the flare, landing, and rollout, is provided by the Autoland System.	DUAL LAND, FLARE, and ROLL-OUT altitude modes	LAND3 mode annunciation followed up with the FLARE pitch mode

Table 5: Modes That Control **Vertical Path** by **Aircraft Pitch**

path is provided by the ILS glideslope receiver (see Chapter 3, Section 2.1), then the glideslope tracking mode, usually indicated by “G/S”, must be engaged.

The differences in mode displays between an MD-11 and Boeing 747 can be seen in Figure 2.

3. Mode Confusion

From the discussion in Section 2, it is easy to see how a pilot might get confused about the current mode of an aircraft. As with any complicated, multi-modal automated system, mode confusion can result when the operator fails to identify his current mode of operation. Mode confusion can cause mode errors which occur when “a particular action that is highly appropriate in one mode of operation is performed in a different, inappropriate mode because the operator has not correctly remembered the appropriate context”[18]. Unfortunately, mode confusion and mode errors are an increasing concern in the cockpit. Mode errors that occur during flight have the potential to lead to major accidents.

Mode confusion and a resulting mode error are at the top of the list of potential causes for the crash of an A320 in Strasbourg, France in 1992[15]. The pilots were given a flight path angle of -3.3° to use in an automatic approach from the final approach fix. This angle would have placed the aircraft in the appropriate position for visual descent[2]. Unfortunately, it appears that the pilots entered “-33” in vertical speed mode instead of flight path angle mode. By entering the desired flight path angle into the flight control unit while in the vertical speed mode, the plane began to descend at a rate of 3,300 ft/min. This resulted in a fatal crash into mountainous terrain. It is believed that one of the reasons this mode error occurred was that the display added to the pilot’s confusion and inability to recognize the error. The displays for flight angle

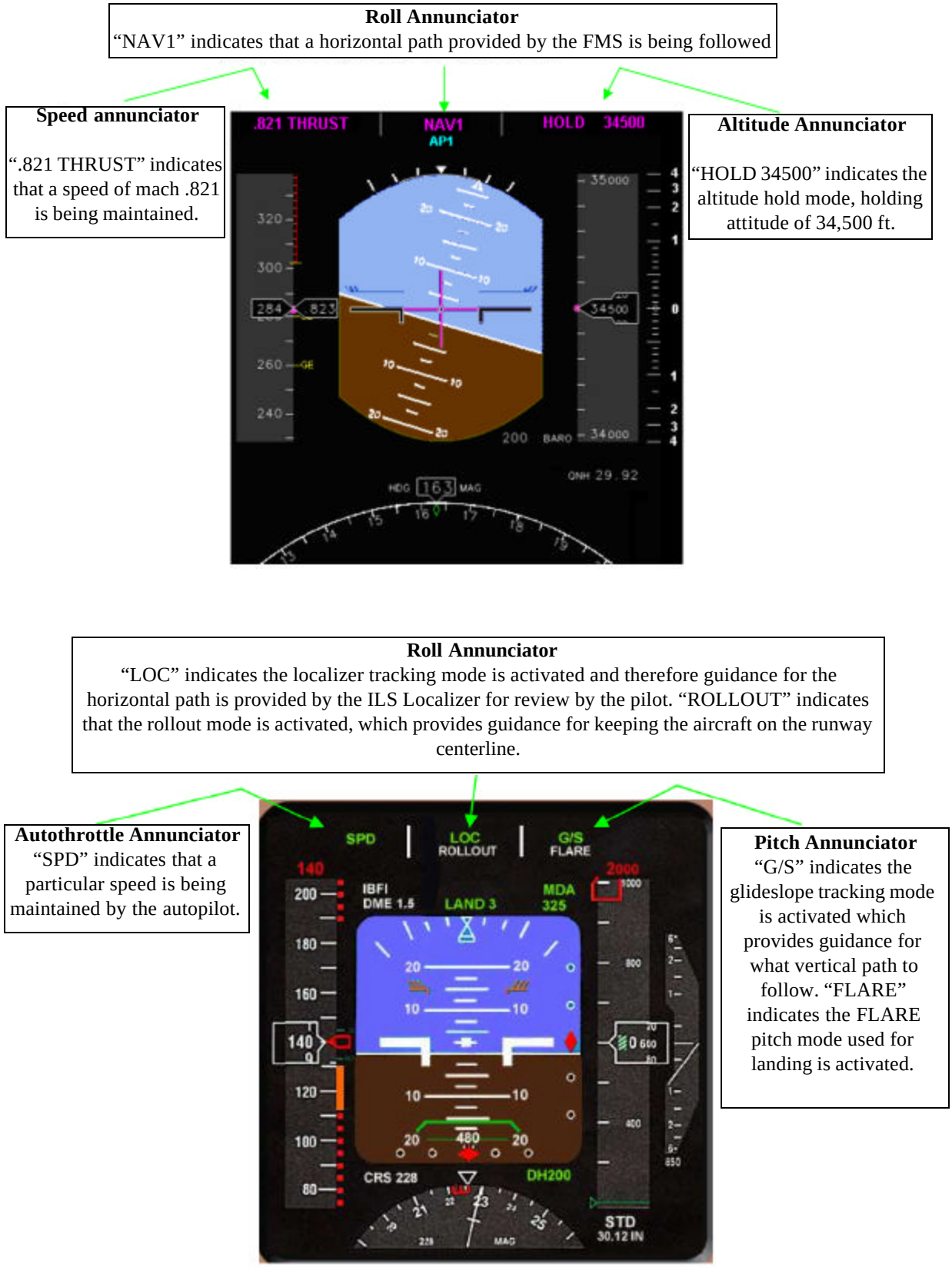

Figure 2. MD-11 (top) and B-747 [14] (bottom) Primary Flight Displays

and vertical speed were both two-digit numbers; a decent of 3,300 ft/min or an angle of 3.3° both appeared as two threes on the display. The only difference in the display between the two modes was a decimal point: the display either read “3.3” or “33”[8]. If four digits had been required to enter the decent rate, it would have been more distinct from the flight path angle display, and this error might not have occurred. This change has since been made to the displays in order to avoid similar mode confusion errors in the future[15].

A survey conducted by Honeywell[4] asked 105 pilots from two different airlines about problems they had faced dealing with mode confusion. The results are unsettling and seem to indicate that current aircraft displays need to be redesigned with a more intensive human factors focus. Almost 75% of the MD-80 pilots surveyed said they have experienced an uncommanded mode change while using an autoflight system. 60% of the MD-80 pilots said they had experienced an uncommanded mode change that led to a failure to capture altitude. 100% of Boeing 747-400 and Airbus A320 pilots also said they have, at one time or another, selected the wrong mode by mistake.

4. Conclusion

Autopilot flight director systems allow for automatic control of an aircraft’s altitude, speed and heading. They reduce pilot workload significantly by making automatic adjustments to hold an aircraft in a predefined orientation or on a predefined route. While the many modes associated with the systems can lead to mode confusion, the benefits they provide in terms of workload reduction make them essential components of modern cockpits.

Acknowledgements

The authors thank Dr. Ellen Bass and Mike Palmer for valuable guidance and helpful information.

References

[1] 767-400ER Flight Deck. The Boeing Company. 24 February 2004 <http://www.boeing.com/commercial/aeromagazine/aero_09/flight_textonly.html#fig03>

[2] Billings, Charles E. *Aviation Automation: The Search for a Human-Centered Approach*. Mahwah, New Jersey: Lawrence Erlbaum Associates, 1997.

[3] Collinson, R. P. G., Introduction to Avionics Systems. Boston: Kluwer Academic, 2003.

[4] Corwin, William H. “Autoflight Mode Annunciation: Complex Codes for Complex Modes.” Honeywell Systems & Research Center.

[5] E-Flight. A320 PFD 24 February 2004 <http://www.e-flight.com/a320pfd.htm>

[6] Fishbein, S., Flight Management Systems: The Evolution of Avionics and Navigation Technology, Praeger, Westport, Conn., 1995

[7] Honeywell. Air Transport Avionics. 21 April 2004. <http://www.cas.honeywell.com/ats/products/fms.cfm>

[8] Krell, Tim A. “Perils of the Glass Cockpit: The Human Factors of Computer Graphics” 23 April 2004. http://www.npl.com/~tkrell/writings/aviation/glass-cockpit.html

[9] Lidén, Sam. “The Evolution of Flight Management Systems.” IEEE, 1994, 157-169. Paper 0-7803-2425-0/94.

[10] Meriweather, Jerome. 2 April 2004. <http://www.meriweather.com/747/center/pfdcapt.html>

[11] Moir, I., Seabridge, A., *Civil Avionics Systems*, American Institute of Aeronautics, Reston, VA, 2003.

[12] Palmer, Michael T. “Generic Modes of Operation of an Autoflight System” Unpublished document, September 30, 1993.

[13] PFD and ND of a 777. 24 February 2004 <http://home.no.net/royeb/instr777.htm>
[14] Project Magenta, 2 April 2004. <http://www.projectmagenta.com>
[15] Sparaco, Pierre. “Human Factors Cited in French A320 Crash.” Aviation Week & Space Technology, January 3, 1997, pp. 30 – 31.
[16] Spitzer. *The Avionics Handbook*. Boca Raton: CRC Press, 2001.
[17] Uczekaj, J.S., “Reusable avionics software. Evolution of the flight management system” Digital Avionics Systems Conference, IEEE. 1995.
[18] Wickens, Christopher D. and Justin Hollands. *Engineering Psychology and Human Performance.* Upper Saddle River, New Jersey: Prentice-Hall Inc, 2000. p. 497.

CHAPTER 6

Flight Management Systems

Sinem C. Göknur Kathryn A. Klein Matthew Bolton

Flight management systems (FMSs) assist pilots with flight planning, navigation, performance management, aircraft guidance and flight progress monitoring. In this chapter we examine the current state of flight management systems by examining their history, current functionalities, components, and some of the differences between the two main FMS groups that stem from Airbus and Boeing designs.

1. Short History of Flight Management Systems

1.1. Origins of the Flight Management System

The birth and acceptance of flight management systems are particularly important events in digital avionics because, prior to the FMS, no avionics system had been as software-intensive. Lidén points out that the introduction of flight management systems "marks the beginning of a phase in avionics where software design plays the dominant role in the system's capabilities and features" [9].

The development of the FMS was a direct response to the oil crisis in 1973, which compelled airlines to find more cost-effective ways of operating their aircraft[6]. Boeing awarded the first FMS contract to Honeywell in 1978. Upon completion of the project in 1982, flight management systems became a part of the standard avionics suite on all Boeing 757 and 767 aircraft[9]. One year after Boeing's contract began, Airbus Industrie signed an FMS contract with Honeywell. In order to fulfill customer requirements and to maintain the uniqueness of the customer features, Honeywell formed two highly separate branches for each FMS' development. In 1983, Airbus officially started delivering an FMS as standard equipment on Airbus 310 aircraft[9].

The early generations of FMS designs were extensively influenced by the Honeywell TERN-100 RNAV (Radio Navigation) system and the Collins AINS-70 RNAV system. Since the initial flight management systems were developed during the late 1970's, their digital equipment was quite primitive by current technology standards. For example, nonvolatile solid-state storage devices with high capacity and speed were not available[9]. In order to cope with computing memory limitations, the software programs had to be short and the data had to be simple.

Following the two pioneer systems, the use of flight management systems spread widely throughout civil aviation. There are three main factors behind the wide acceptance of FMSs: they significantly reduce pilot workload; they minimize travel costs through flight-profile optimization; and they help enable the growth in air traffic [3].

1.2. Development of the Flight Management System

Current commercial FMSs originated from the Honeywell designs for the Boeing 757/767 (influenced by the TERN-100 Automatic Navigation System) and for the Airbus 310 aircraft (influenced by Collins AINS-70 RNAV). This variation in influence, however, did not result in significant differences in the functionality or hardware of the final products[9]. Most of the disparity was situated in the human interface component, such as the arrangement of the *multipurpose control display unit* (MCDU) pages and function keys (discussed in Section 3.1). Moreover, the Airbus design had additional functionality to support the needs of the European aviation community[9].

During the mid 1980s, the strong customer desire to add more capability and functionality without paying full development costs started to create difficulties for the FMS developers. In order to cope with this problem, Honeywell developed a strategic plan for the reuse of design in future programs[17].

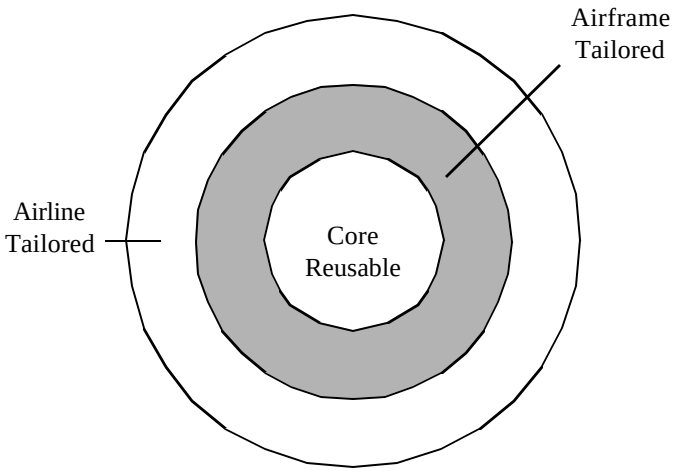

Figure 1. The Three Layers for FMS development[17]

After identifying customers' key needs and determining the FMS functions that were essential, Honeywell determined there to be three key layers in the development process of flight management systems, illustrated in Figure 1. Two of the layers are tailored to specific operating conditions and are centered around a reusable core. The center of the three-tiered reusable strategy represents the main functions that are performed by every FMS. The second ring represents the functions that need to be tailored to a specific airframe. The input and output functions, for example, would be found in this layer. The outermost ring symbolizes the highest level of functionality, usually tailored for specific airline demands that are mostly at the user interface level. An example would be how information pages are accessed [17].

As a result of this approach, most FMSs currently employed by Boeing and Airbus have nearly the same functionalities. Nonetheless, there are definite distinctions in the "Airframe Tailored" and "Airline Tailored" levels.

2. Main Functions of the FMS

The FMS carries out important tasks such as[3]:

- supplying a digital platform for flight planning and navigation,
- providing flight guidance along the flight path,
- monitoring the flight envelope while ensuring safety margins,
- computing the optimal flight profile for minimum travel costs, and
- affording automatic control of engine thrust.

In this section, we focus on the main functionalities of the FMS: flight planning, navigation and guidance, and prediction and optimization.

Flight Planning

Flight planning consists of identifying the correct sequence of waypoints, flight levels and airways, and arranging procedures for takeoff and arrival[8]. The pilot may choose to enter the flight plan information manually or obtain it from the existing databases in the FMS. The FMS contains individual databases for waypoints, airways, airports, runways, radio navigation aids, company routes, and airport procedures[3]. These databases, which are updated every 28 days, are referred to as the *Navigation Data Base* (NDB) and are stored in nonvolatile memory[3]. The information in the NDB can be presented to the pilot through the MCDU (discussed in Section 3.1) by using alphanumeric characters or through the Navigation Display (also discussed in Section 3.1) by using navigation graphics [2].

Flight plans can be created manually by entering a company route from the NDB or automatically by up-linking an existing flight plan via the Aircraft Communications Addressing and Reporting System (ACARS)[8]. The pilot can enter up to two distinct flight plans into the FMS. The *active flight plan* is used for the aircraft guidance. The pilot may choose to switch to the *alternative flight plan*, or use it while considering changes to the active flight plan[8]. The FMS provides considerable flexibility in flight planning tasks. For example, the pilot can easily revise the flight plan at any time and make necessary changes via the MCDU[3].

Navigation and Guidance

The navigation facility provided by the FMS determines position, aircraft velocity, wind velocity, relative wind, drift angle, etc., and manages navigation data sources [2, 3]. The FMS augments the accuracy of the position estimate by combining the data from all navigation sources [3]. The navigation sources used by the FMS consist of the Inertial Navigation System (INS), Global Positioning System (GPS), and various radio navigation aids such as VOR, DME and AHRS (see Chapter 3) [8].

During the flight, radio navigation aids are automatically chosen and tuned by the FMS based on their specification in the flight plan. The processed navigation data is provided to the pilot through a navigation display, and this data is used to generate guidance information for the aircraft[3].

The FMS provides guidance support by calculating error, generating steering commands, and generating lateral and vertical control commands [2]. Once the guidance information is processed, the FMS sends it to the Autopilot Flight Director System (discussed in Chapter 5). The guidance information is displayed to the pilot via Electronic Flight Instrument System displays (discussed in Section 3.1)[8].

Performance Prediction and Optimization

The performance prediction and optimization tasks of the FMS include the determination of the flight trajectory, computation of optimal speeds, predictions of time and fuel at all waypoints of the flight, and the calculation of reference parameters such as optimum altitude, approach speed, and maximum altitude[2]. Since the FMS contains a performance database, it saves the pilot from consulting performance manuals during the flight[2].

The FMS computes the optimal flight path based on predicted cruise winds, gross weight, cost index, speed/altitude/time constraints, and engine performance characteristics. It also utilizes the current state information, such as wind, temperature, airspeed and altitude for these calculations[3]. The performance prediction and optimization computations are continuously carried out during the flight. As a result of this, the FMS provides information on the optimal flight path as well as predictions of time, fuel, wind and temperature for each waypoint. It also presents information on climb and descent and creates altitude and time markers [3].

3. FMS Components

3.1. Flight Management Computer System

The *flight management computer system* (FMCS) has two types of components: the *flight management computer* (FMC) and the *multipurpose control display unit* (MCDU). The FMCS installation on aircraft is usually composed of two FMCs and three MCDUs, for back-up purposes[8]. Figures 2 and 3 provides example pictures of an FMC and an MCDU.

Courtesy of Honeywell International Inc.

Figure 2. Honeywell FMC[7]

Flight Management Computer (FMC)

The FMC is the main processor of the FMCS. This component is typically installed in the avionics compartment of the aircraft. The Flight Management Computer forms a bridge between the main components of the FMS. Since it is such a crucial component, redundancy is used to ensure adequate reliability; two separate FMCs are usually installed in the aircraft, one master and one backup. For consistency, data is cross-fed to these two FMCs through the MCDUs, and both FMCs have to confirm the entry for new data acceptance. The communication between the FMCs occurs through a private data bus [2].

As an example, the FMC software in MD-11 aircraft includes the Navigation Database (NDB), the principal FMC Operational Program, and the Performance Database. Other aircraft have similar software in their Flight Management Computers. The Operational Program for MD-11 consists of over 1400 software modules in order to execute the necessary FMS functionalities [3].

Multipurpose Control Display Unit (MCDU)

The MCDU serves as the main interface between the pilot and the Flight Management Computer. It consists of a cathode ray tube (CRT) or liquid crystal display (LCD) and a keyboard providing alphanumeric characters, line select keys, function keys, and specific mode select keys. Again, as an example, the MD-11 MCDU also provides light sensors and a manual control to adjust the display brightness.

Courtesy of Honeywell International Inc.

Figure 3. Honeywell MCDU [7]

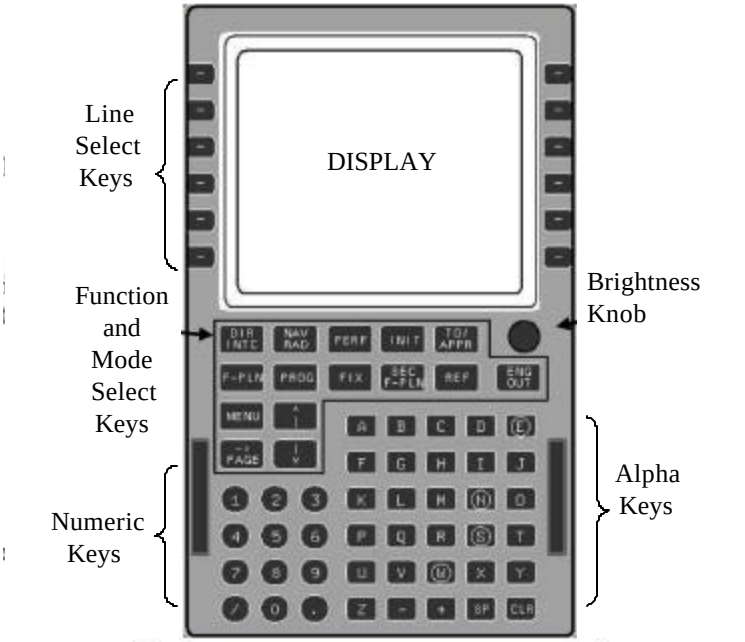

Figure 4. MCDU Layout

Figure 4 illustrates the main parts of the MCDU for an MD-11. The mode select keys are used to access different function pages of the FMS. The alphanumeric key set is used to enter data manually into the FMC. The display of the MCDU is divided into four sections: the title field on top, the main display (which is divided into left and right fields) in the middle, and the scratch pad field at the bottom.

Besides functioning as the interface between the pilot and the FMC, current MCDUs also have the capacity to serve as a backup for the FMC and to control some of its subsystems[8].

3.2. Electronic Flight Instrument System

The FMS produces a large amount of information that needs to be displayed to the pilot and first officer. The data is primarily displayed through the *Electronic Flight Instrument System* (EFIS), which is usually composed of two *primary flight displays* (PFDs) and two *navigation displays* (NDs), discussed below.

Primary Flight Display

The most prominent feature of the primary flight display is the artificial horizon located in the center of the display. The horizon line moves as the aircraft begins to pitch and roll. This allows the pilot to monitor the position of the aircraft relative to the horizon. The pitch scale, or pitch ladder, marks off degrees of pitch using a number of lines parallel to the horizon. The current airspeed and altitude can be read from the vertical bars to the left and right of the display, respectively. Heading information is presented at the bottom of the display, while mode information (described in Chapter 5, Section 2) is displayed at the top of the PFD. The PFD also displays stall warnings and guidance information[1]. The most recent update to the Boeing flight deck can be found in the Boeing 767-400 Extended Range (B 767-400ER)[1]. The Boeing 747 PFD[14] shown in Figure 5 illustrates the features of a modern PFD.

Figure 5. Boeing 747 PFD[14]

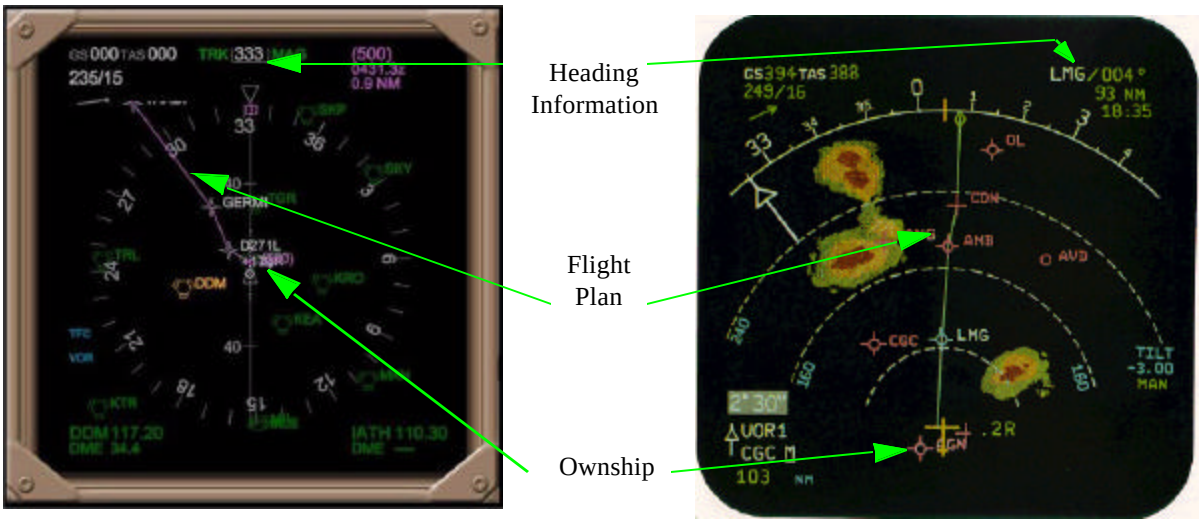

Figure 6. Boeing 777 ND [14] (left) and A320 ND [5] (right)

Navigation Display

The Navigation Display (ND) is used for graphic map display. The ND is an egocentric display: the pilot's aircraft, designated *ownship*, is presented in the center of the map and does not move. All other objects, such as other aircraft or nearby waypoints, move around the ownship icon. As seen below in Section 3.4, the ownship is represented by a triangle pointed up in the Boeing 777 display and with a cross in the A320 display. The pilot can glance at the ND to get a sense of his location with respect to other traffic. The ND also provides the pilot with data on his actual position versus planned position. The line in the center of both displays connects the waypoints that compose the flight plan. The pilot can tell from the display if he is to the left or right of his flight plan. The ".2R" in the A320 display tells the pilot that he is 0.2 nautical miles to the right of his flight plan.

Heading information is also displayed on the ND. The numbered tick marks around the outside of the display represent tenths of a degree. In the Boeing 777 the heading is explicitly displayed at the top center of the ND in a box, while in the A320 it is displayed at the top right hand portion of the screen. It is easy to see in Figure 6 that the Boeing 777 is flying to a heading of 346° and the A320 to a heading of 4°.

Data such as true airspeed, ground speed, and wind data are also displayed on the ND. Additionally, the ND is used to display warnings provided by the Traffic Collision Avoidance System (TCAS) and the Ground Proximity Warning System (GPWS).

The Boeing 767-400ER is equipped with an ND that has the same format as the Boeing 777's. The Boeing 737-600/-700/-800/-900, and 747-400displays are also similar [1].

3.3. Component Interactions

The main interactions among all the components of the FMS[3] are shown in Figure 7. As this figure illustrates, the FMC outputs data to EFIS (which consists of PFDs and NDs), and to the AFDS (discussed in Chapter 5). The FMC receives input from various navigation sources. These include the Air Data Computer (ADC) and the Engine Data component. Finally, the FMC exchanges data with the Flight Data Storage Unit and with the MCDU.

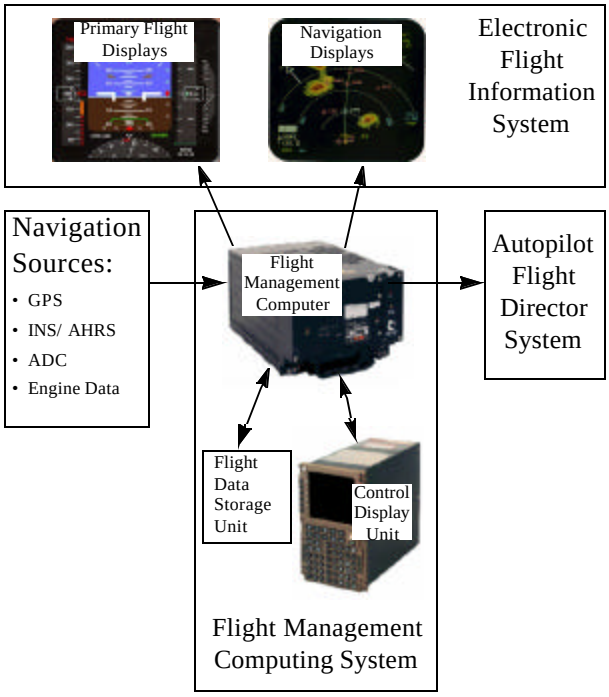

Figure 7. FMS Components

3.4. Brown vs. Gray FMS Systems

Systems based on the initial Boeing 757/767 FMS are called “brown” systems because of the brown color of the 757/767 cockpit. Likewise, all systems that are influenced by the initial Airbus 310 FMS are called “gray” systems, because the Airbus 310 cockpit is gray. Most current Flight Management Systems are categorized either as brown or gray. The Douglas MD-11, which is now part of Boeing, took operational features from both systems, yet it is classified as a gray system[17]. Figure 8 illustrates the evolution of FMSs from the two initial designs of the 1980’s to the later designs of the 1990’s.

Some initial differences between the gray and brown systems have faded away through the addition of features to newer generations of flight management systems. Nevertheless, differences remain between the

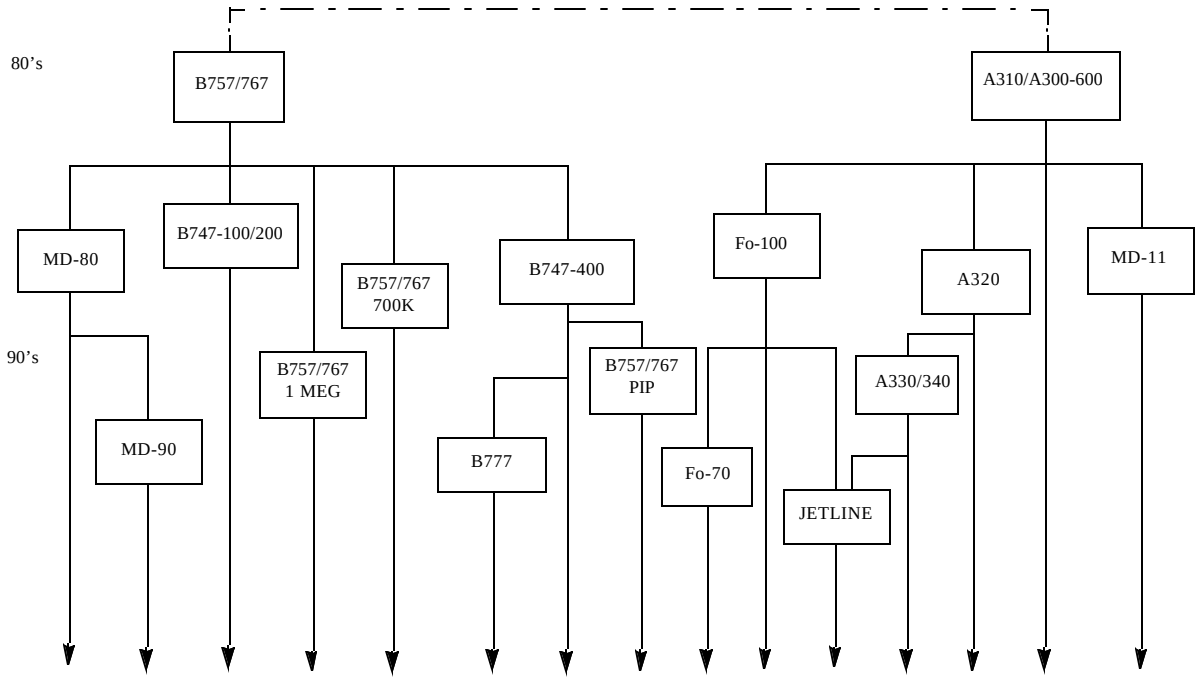

Figure 8. Honeywell FMS family tree[9]

Figure 9. CDU panels from Boeing 747 (left) [14] and MD-11 (right) simulations

two. The most notable differences between the current gray and brown systems are in the CDU page layouts, the design and use of the keyboard function keys, and in the ways the CDU pages are accessed. Figure 9 shows the CDU panels from a Boeing 747 (brown) simulation and an MD-11 (gray) simulation. Unlike the gray system on the right, the brown system scratch pad is separated from the rest of the CDU display by a dotted line. Also, although the grouping of the keys is similar for both panels, the layout of the function keys has some notable differences. Each system has its own strengths and weaknesses; the favorability of design differences is subjective and dependent upon the airline's and pilot's individual needs.

4. Conclusion

Flight management systems and autopilot flight director systems can be seen on almost all modern flight decks. Both systems work hand in hand to significantly reduce pilot workload by providing features that automate much of the flight process. It is clear that these systems provide many beneficial features like performance management, flight progress management, aircraft guidance, and aircraft control, but they also have drawbacks. As with most systems, added capabilities produce added complexity. Due to the many modes of the AFDS, mode confusion can result and possibly negate the positive impacts of the system in the cockpit. The outdated and overly complicated interface to the FMS also can add to pilot confusion. It is clear that while both systems are thought to be very valuable in general, there are still concerns that need to be addressed.

Acknowledgements

The authors thank Dr. Ellen Bass and Mike Palmer for valuable guidance and helpful information.

References

[1] 767-400ER Flight Deck. The Boeing Company. 24 February 2004 <http://www.boeing.com/commercial/aeromagazine/aero_09/flight_textonly.html#fig03>

[2] Billings, Charles E. *Aviation Automation: The Search for a Human-Centered Approach.* Mahwah, New Jersey: Lawrence Erlbaum Associates, 1997.

[3] Collinson, R. P. G. *Introduction to Avionics Systems*. Boston: Kluwer Academic, 2003.

[4] Corwin, William H. "Autoflight Mode Annunciation: Complex Codes for Complex Modes." Honeywell Systems & Research Center.

[5] E-Flight. A320 PFD 24 February 2004 <http://www.e-flight.com/a320pfd.htm>

[6] Fishbein, S., Flight Management Systems: The Evolution of Avionics and Navigation Technology, Praeger, Westport, Conn., 1995

[7] Honeywell. Air Transport Avionics. 21 April 2004. <http://www.cas.honeywell.com/ats/products/fms.cfm>

[8] Krell, T. A. "Perils of the Glass Cockpit: The Human Factors of Computer Graphics" 23 April 2004. http://www.npl.com/~tkrell/writings/aviation/glass-cockpit.html

[9] Lidén, S. "The Evolution of Flight Management Systems." IEEE, 1994, 157-169. Paper 0-7803-2425-0/94.

[10] Meriweather, J. 2 April 2004. <http://www.meriweather.com/747/center/pfdcapt.html>

[11] Moir, I., Seabridge, A., *Civil Avionics Systems*, American Institute of Aeronautics, Reston, VA, 2003.

[12] Palmer, M. T. "Generic Modes of Operation of an Autoflight System" Unpublished document, September 30, 1993.

[13] PFD and ND of a 777. 24 February 2004 <http://home.no.net/royeb/instr777.htm>

[14] Project Magenta, 2 April 2004. <http://www.projectmagenta.com>

[15] Sparaco, P. "Human Factors Cited in French A320 Crash." Aviation Week & Space Technology, January 3, 1997, pp. 30 – 31.

[16] Spitzer. *The Avionics Handbook*. Boca Raton: CRC Press, 2001.

[17] Uczekaj, J.S., "Reusable avionics software. Evolution of the flight management system" Digital Avionics Systems Conference, IEEE. 1995.

[18] Wickens, Christopher D. and Justin Hollands. *Engineering Psychology and Human Performance.* Upper Saddle River, New Jersey: Prentice-Hall Inc, 2000. p. 497.

CHAPTER 7

Mission Specific Avionics

John C. Giordano

This chapter presents a case study and analysis that demonstrates how diverging user requirements can result in vastly different design and implementation decisions within a single airframe. Two very common airframes in the defense community, Sikorsky's UH-60 Blackhawk and Lockheed's C-130 Hercules, are examined and the additional avionics components found aboard their respective special operations variants are presented and discussed.

1. Functional Requirements – Influence on Design

System design and implementation generally reflect considerations for what the system is intended to do. A number of stakeholders contribute to shaping system requirements and come from numerous organizational cross-sections. Managers, maintainers, investors and end users are some of the stakeholders that may influence the design process. There are some instances, however, in which the functional capabilities of a system may be so broad and varied that a single avionics design or configuration is insufficient to meet all expectations.

Special operations aircraft exhibit the characteristics described above, thus serving as a compelling topic for this case study. As a result, the state of avionics systems implemented in common configurations of some well-known airframes differs significantly from special operations variants of the same airframe. This divergence of technology and capability derives from the vastly different requirements that are articulated for each of these platforms. The result is that, within a single airframe, various cockpit or flight-deck designs emerge, each consisting of a number of components that relate to user tasks or requirements. As these tasks and requirements increase in number or complexity, so do the avionics systems that support them. The cognitive and physiological demands placed upon the pilots and flight crew of common airframe variations, while rigorous, are somewhat less than that of their special operations counterparts. This divergence is clearly demonstrated in several airframes, two of which we discuss below.

1.1. The UH-60 and MH-60 Variant

In 1979, the US Army began introducing its training and operational units to a new light transport and utility helicopter known as the UH-60 Blackhawk, manufactured by Sikorsky. The Blackhawk was intended to replace the venerable UH-1 Iroquois or "Huey", which had been used heavily during the Vietnam conflict. The new helicopter was to perform diverse missions, such as transporting combat troops, cargo and supplies, evacuating casualties and serving as a platform for communications, command and control.

Initially, the UH-60 had to convince some hardened critics of its capabilities and performance, mainly the community of pilots and others who had become accustomed to the Huey and its legendary versatility.[1]

Figure 1. Cockpit Illustrations: On the Left, a UH-60A [45]; On the Right, a MH-60 [46].

Having successfully done so, variations of the Blackhawk are now in use by the US Air Force, Navy, and Coast Guard, along with the militaries of several allied nations. The Blackhawk found a home outside of the defense community as well, most notably with civilian law enforcement, firefighting and rescue agencies. Over time, the Blackhawk has evolved and adapted to fill numerous roles. The MH-60, a special operations variant of the baseline UH-60 airframe, contrasts sharply with its humble roots as a utility helicopter.

Designed for long range, low-level penetration of enemy territory at night and in adverse weather, the requirements and capabilities of the MH-60 differ greatly from that of its predecessor. As a result, the cockpit of the MH-60 features a number of components that are not needed on the common versions of the airframe. These components include MFDs, MCDUs, and LCDs, all of which are used for primary flight display (see Figure 1). A weapons system controller is also included in the cockpit; such a component is usually not found on a UH-60. In order to coordinate these additional components, the MH-60 includes a Cockpit Management System (CMS). The CMS was designed using a Common Avionics Architecture Approach which will be applied to two other special operations helicopters in the Army fleet[8]. This represents a design and implementation departure from the baseline Blackhawk configurations motivated by the exceptional demands of the users, managers and maintainers of special operations aircraft. Moreover, although the airframe remains a common denominator between the UH-60 and the MH-60, the cockpit and primary flight displays demonstrate the differences in requirements and capabilities, which will be discussed later. It would be cost-prohibitive to acquire, maintain and employ all of the sophisticated avionics found in MH-60s aboard the US fleet of UH-60s, of which there are hundreds. The US Army maintains only about 60 MH-60s in active service.

1.2. The C-130 and AC-130 Variant

Unlike the Blackhawk, a relative newcomer to military aviation, the earliest versions of the US Air Force's C-130 Hercules transport planes were flown and acquired in late 1955. Originally designed as an assault transport, this four-propeller aircraft has been adapted for even more varieties of missions than the Blackhawk, including intra-theater supply and logistics, airdrop and air resupply, search and rescue, electronic warfare, and most interestingly, special operations close air support.

1. When first fielded, most US Army pilots received inadequate training on the new helicopter, resulting in a dramatic increase in accidents involving this airframe [1]. In over twenty years of widespread use, various design flaws have been identified and fixed, yet some question the reliability of the Blackhawk, as evidenced by several instances when the entire Army fleet has been grounded[12].

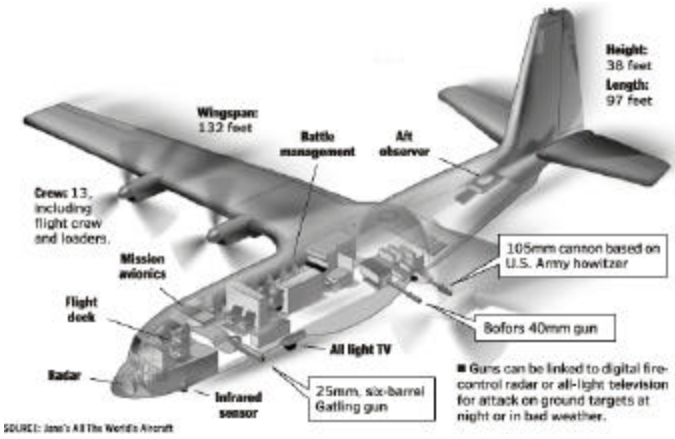

Figure 2. Illustration of the AC-130.

Like the Blackhawk, this capability was not a consideration at design time. Therefore, the systems within the airframe had to be modified in order to accommodate the additional required functionality. As well as coping with more sophisticated avionics, the special operations variant of the C-130, the AC-130, packs a deadly punch. Armed with a 105mm cannon, 40mm cannon, and two 20mm guns, the AC-130 normally orbits above an area of operations, providing continuous coverage of the area below with its armaments, all of which protrude from the left side of the aircraft (see Figure2).

Along with drastically different flight dynamics of its common counterparts, the AC-130 must provide surveillance and fire control capability in order to efficiently employ its weapons. Resulting from requirements for special operations capability, the AC-130 has been equipped with avionics subsystems and architectures that do not appear in common variants of the same airframes. For example, in order to provide ground attack capability, the AC-130 features an APQ-180 radar system, which is a derivative of the APG-70 system found on F-15 fighter aircraft[38].

Integrating the many subsystems that help the crew perform their unique missions has proved challenging. In the AC130, a Mission Computer serves as a key component of the avionics suite. It integrates the functions of flight control, navigation, weapons system management and overall aircraft functionality into a single system. As these Mission Computers age, they have been plagued by parts obsolescence and performance limitations[25]. Like the MH-60, the AC-130 is quite sparse in number. As a result, maintenance and upkeep can become problematic. Only 21 AC-130 variant airframes are maintained in active service.

2. Special Requirements for Special Operations

Since the MH-60 and its crew perform demanding missions in adverse conditions, it follows that this platform requires additional avionics components in order to aid the flight crew. MH-60 pilots are specially selected and trained to perform tasks above and beyond what their counterparts are required to on a baseline UH-60. Some of the tasks associated with a routine flight profile on a UH-60 include[13]:

- participate in crew mission briefing,
- plan VFR/IFR flight,
- verify aircraft weight and balance,
- perform preflight inspection and various checks,
- perform Visual Meteorological Conditions (VMC) takeoff,
- perform internal load operations,
- perform terrain flight,
- navigate by pilotage and dead reckoning,
- perform holding procedures,
- perform VMC approach, and
- perform landing operations.

While the above list does not include all tasks that are conducted during a routine flight on a UH-60, it is representative of the major tasks that the crew must perform in order to ensure a safe, productive mission. There are a number of tasks associated with the routine operation of a UH-60, like fastrope insertion of soldiers, paradrop operations, or repelling procedures[13], which appear exceptionally demanding. While this is true, we invite the reader to consider the tasks articulated in the list above, which must be mastered by UH-60 pilots before they are selected to receive special training for the MH-60. It is difficult to describe any MH-60 mission as routine, and the tasks listed below must be performed in addition to the ones above during a typical MH-60 assault flight profile[14]:

- Conduct detailed rehearsals
- Conduct operational checks with ground force
- Conduct armed infiltration and exfiltration of ground forces
- Maintain secure communications with ground force
- Provide aerial fire support
- Provide aerial refueling

While discussion of the above tasks in great detail is outside the scope of this work, the typical MH-60 mission could be characterized as the union of the two lists in this subsection. Although flight crews may develop expertise on the baseline tasks in the first list, the cognitive demands of performing all tasks in conjunction with a special operations mission (which are usually conducted at night, and often in adverse weather) can become significant. Rather than overwhelm the operators with excessive cognitive workload, additional avionics systems have been introduced into the cockpit.

Similarly, the C-130 fulfills a utility role in its baseline configuration. Transporting troops, equipment, vehicles or supplies, the overarching mission of the C-130 is logistical support; the AC-130 has a drastically different role. Based on the AC-130's distinguishing role as a ground attack aircraft, air crews must perform additional, highly demanding tasks, just like in the MH-60. For example, an AC-130 operator must be able to perform the following[15]:

- Pre-strike: Configure the aircraft for combat prior to crossing the combat entry point
- Tweak procedures: Coordinate with the crew to align sensors/weapons for accurate weapons delivery
- Target acquisition: Pilot must be able to coordinate with tactical crew to acquire the correct target/ friendly position
- Gunnery: Demonstrate proficiency in two gun fire modes

The convergence of additional equipment, sensors and flight crew requirements results in an increased workload for the operator. While those who fly the MH-60 and AC-130 usually possess expertise beyond what is expected for the respective baseline configurations, the human operators are still bound by the constraints discussed earlier in this chapter and are prone to the same kinds of errors that all humans make. As a result, the flight crew are afforded a richer set of avionics components in order to perform their missions with greater efficiency and a lower likelihood of error or failure.

3. Common Airframes for Dissimilar Needs

While the common and special operations requirements that drive designs for the UH-60 and C-130 are not mutually exclusive, they demonstrate that the flexibility of common airframes must be augmented by specialized avionics systems and design methodologies. Since the MH-60 and AC-130 are maintained in such sparse quantities, it would be inefficient and cost-prohibitive to develop specialized airframes in order to yield the desired capabilities. The sponsoring agencies must leverage the efficiency afforded by the common airframe which are purchased in large quantities. Over time, as the specialized avionics systems that

support special operations requirements become obsolete, innovative means for replacing and upgrading these components must be considered in order to continue to realize the cost savings [26]. The design and implementation differences emerging from increased user requirements extend beyond the cockpit and flight deck of these aircraft. For example, consideration has been given to redesigning the aft-scanner workstation on the AC-130 as a result of user fatigue and discomfort[41]. The aft-scanner is a crew member that must recline in a prone position for hours at a time, watching the ground below for anti-aircraft fire. This very unique requirement is not considered on the common C-130 airframe.

The evolution of the baseline airframes to their special operations variants serve as an example of an incremental design process. The user requirements for the MH-60 were not specified when the UH-60 was initially designed and hence, it was not envisioned to be an extensible air frame at design time. Additional avionics aboard the MH-60 were added in an ad hoc fashion and contribute to a disintegrated design process.

References

[1] History of the army aviation warrant officer. www.usawoa.org, 1998.

[2] Airdisaster.com. www.airdisaster.com, 2004.

[3] Enhanced ground proximity warning system. www.egpws.com, 2004.

[4] Abbott,K.H. "Human Factors Engineering and Flight Deck Design." In C. Spitzer, editor, *Avionics Handbook*, Chapter 9. CRC Press, 2001.

[5] Abbott,T.S. "A Simulation Evaluation of the Engine Monitoring and Control System Display." NASA-TP-2960, Feb. 1, 1990.

[6] Arbuckle,P.D, K.H.Abbott, T.S.Abbott, and P.C.Schutte. "Future flight decks." *Proceedings of the 21st Congress International Council of the Aeronautical Sciences*, 1998.

[7] Bailey,I. "Honeywell EGPWS, Getting the Job Done — Part 1." *Avionics News*. March, 2004.

[8] Barbacci,M, P.Clements, A.Lattanze, L.Northrop, and W.Wood. *Using the Architecture Tradeoff Analysis Method (ATAM) to Evaluate software architecture for a product line of avionics systems: A case study*. Technical report, Carnegie Mellon University - Software Engineering Institute, 2003.

[9] Bass,E.J, *et al.* "Architecture and Development Environment of a Knowledge-based Monitor that Facilitates Incremental Knowledge-base Development." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 34, No. 4.

[10] Billings,C.E, *Aviation Automation: The Search for a Human-Centered Approach*. Lawrence Erlbaum Associates, Mahwah, NJ, 1997.

[11] Campbell, R. and M. Bagshaw. *Human Performance and Limitations in Aviation*. Blackwell Science, Oxford, 2002.

[12] Carroll,J. "Blackhawks Have History of Crashes, Problems." *Courier-Journal*, Louisville, KY, April 23, 1999.

[13] U.S.A.A.Center and School. *Aircrew Training Manual Utility Helicopter UH-60/EH-60 - Training Circular 1-212*. Headquarters, Department of the Army, Washington, DC, 2002. [12]

[14] U.S.A.J.F.K.S.W.Center and School. *U.S. Army Special Operations Aviation Aircrew Training Program Training Circular 1-210-1*. Headquarters, Department of the Army, Washington, DC, 2002.

[15] U.S.A.F.S.O.Command. *AC-130 Aircrew Evaluation Criteria - Air Force Instruction 11-2AC-130*. Department of the Air Force, Washington, DC, 2000.

[16] Endsley,M. "Situation awareness in aviation systems." In J.W.D.Garland and V.D.Hopkin, editors, *Handbook of Aviation Human Factors*. Lawrence Erlbaum Associates, 1999.

[17] M.T.P. *et al. A Crew-Centered Flight Deck Design Philosophy for High-Speed Civil Transport (HSCT) Aircraft*. NASA Technical Memorandum 109171, 1995.

[18] Butler,R.W. *et al. A Formal Methods Approach to the Analysis of Mode Confusion.* Technical report, NASA, 1998.
[19] Foyle,D.C, A.H.Ahumada, J.Larimer, and B.T.Sweet. "Enhanced/Synthetic Vision Systems: Human Factors Research and Implications for Future Systems." *SAE Transactions: Journal of Aerospace*, 101, 1734-1741, 1992.
[20] Reising,K.L.J, and R.Munns. "Controls, Displays and Workplace Design. In J. W. D. Garland and V.D.Hopkin, editors, *Handbook of Aviation Human Factors*. Lawrence Erlbaum Associates, 1999.
[21] Kerr, J. R, C. H. Luk, D. Hammerstrom, and M. Pavel. "Advanced Integrated Enhanced Vision systems." *SPIE Aerosense*, April 21-25, 2003, Orlando, Florida, 2003.
[22] W. J. Kubbat, P. M. Lenhart, and H. V. Viebahn. 4d flight guidance displays, a gate to gate solution. 17th DASC-Electronics in Motion, 1998.
[23] N. G. Leveson. Safeware: System Safety and Computers. Addison-Wesley, New York, 1995.
[24] N. G. Leveson and E. Palmer. Designing automation to reduce operator errors. In Proceedings of Systems, Man, and Cybernetics Conference, 1997.
[25] H. Lowery and B. Mitchell. Mission computer replacement prototype for special operations forces aircraft: An application of commercial technology to avionics. In Proceedings of the 19th Digital Avionics Systems Conference, October 2000, pages 4.E.1–1 – 4.E.1–6.
[26] J. A. Luke, D. G. Halderman, and W. J. Cannon. A cots-based replacement strategy for aging avionics computers. CrossTalk: The Journal of Defense Software Engineering, December 2001, 14-17, 2001.
[27] S. P. Miller and J. N. Potts. Detecting mode confusion through formal modeling and analysis. Technical Report NASA/CR-1999-208971, National Aeronautics and Space Administration, 1999.
[28] Mitchell,C.M, *et al.* "OFMspert I: Operations Automation." *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics.* Oct., 2000, Vol. 2.
[29] G. Miyahara and C. P. Satterthwaite. A multiplatform support environment. In Proceedings of the IEEE 1997 National Aerospace and Electronics Conference, pages 815–819.
[30] I. Moir and A. Seabridge. Aircraft Systems: Mechanical, Electrical, and Avionics Subsystems Integration. AIAA, Reston, VA, 2001.
[31] R. Onken, "The cockpit assistant system CASSY as an on-board player in the ATM environment," presented at the US/Eur. ATM R&D Seminar, Paris, France, June 1997
[32] C. Perrow. Normal Accidents: Living with High-Risk Technologies. Princeton University Press, Princeton, New Jersey, 1999.
[33] J. W. Ramsey. Weathering fog and darkness–huds with enhanced vision. www.defensedaily.com, 2004.
[34] J. Reason. Human Error. Cambridge University Press, 1990.
[35] W. Reynish. Enhanced vision: Final link in situational awareness. www.defensedaily.com, 2004.
[36] Rubin,K.S, P.M.Jones, and C.M.Mitchell. "OFMspert: Inference of Operator Intentions in Supervisory Control Using a Blackboard Architecture." *IEEE Transactions on Systems, Man, and Cybernetics.* Vol. 18, No. 4.
[37] P. C. Schutte and K. F. Willshire. Designing to control flight crew errors. NASA Technical Memorandum, 1997.
[38] K. Smith and G. Miyahara. Leveraging an avionics support environment for shared application to multiple platforms. In Proceedings of the 16th Digital Avionics Systems Conference, October 1997, pages 1.3–9 – 1.3– 15.
[39] E. Theunissen, F. D. Roefs, R. M. Rademaker, and T. J. Etherington. Integration of information in synthetic vision displays: Why, to what extent and how? 2003.
[40] Thurman,D.A, A.R.Chappell, C.M.Mitchell. "An Enhanced Architecture for OFMspert: A Domain-Independent System for Intent Inferencing." *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics.* Vol. 1, Oct., 1998

[41] M. B. Tischler, J. D. Colbourne, M. R. Morel, D. J. Biezad, K. K. Cheung, W. S. Levine, and V. Moldoveanu. A multidisciplinary flight control development environment and its application to a helicopter. IEEE Control Systems Magazine,19, 22-33, 1999.
[42] Weiner, E.L. and R.E.Curry. “Flight-Deck Automation: Promises and Problems.” *Ergonomics*. Vol. 23, No. 10, 1980.
[43] C. Wickens and J. Hollands. Engineering Psychology and Human Peformance. Prentice Hall, Upper Saddle River, NJ, 1999.
[44] Verfurth,S.C, T.Govindaraj, C.M.Mitchell. “OFMspert for the 727: An Investigation into Intent Inferencing on the Flight Deck.” *Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 2, Oct., 1991.
[45] http://69.12.6.178/AWA1/101-200/walk158_MH-60G/part3/walk158C.htm
[46] http://us-aircraft.com/walk/MH-60.htm

Part III

Digital Avionics Dependability

CHAPTER 8

Human Centered Design

M. Anthony Aiello John C. Giordano Sarah Waziruddin

In modern commercial air transportation, fuel economy and passenger comfort have gained importance[33]. Every effort is made to reduce flight time and optimize aerodynamic factors for maximum fuel economy. Rates of turn, climb and descent are carefully monitored in order to maintain the high standard of passenger comfort demanded by the FAA, airlines, and their customers. The speed at which modern commercial aircraft travel leaves little time for human operators to directly monitor and control all the systems on the aircraft to achieve these goals.

To help achieve these new performance goals, modern commercial aircraft *flight decks*, the cockpit from which the crew controls the aircraft, have become heavily automated. The job of the flight engineer, who formerly monitored and controlled aircraft subsystems, has been largely replaced by systems such as Full Authority Digital Engine Controllers (FADECs, discussed in Section 2.1). Other tasks previously delegated to the flight engineer have been shifted to the pilot and copilot, who manage subsystems with the aid of automated display and control systems. The pilot and copilot, in turn, have seen much of their traditional workload shifted to automated systems such as flight management systems (FMSs) (see Chapter 8). Thus, on the modern flight deck, the flight crew frequently monitors the automation and provides backup when systems fail.

As technology progresses, safety margins that have existed in commercial airspace for decades are slowly being reduced. Improvements to navigation have made possible Category II and Category III landings (see Chapter 3, Section 1) under instrument flight rules. Enhanced and synthetic vision systems, in addition to aiding in IFR Category II and III landings, offer airlines and business jets the ability to land at poorly equipped airports in inclement weather. Landing under such adverse conditions is only possible due to the assistance the flight crew receives from automation; were the automation to fail, there would be very little time for the crew to take corrective action.

The growing dependence of the flight crew on automation has led to a reconsideration of overall flight-deck design and philosophy. Flight-deck designers must therefore consider the many ways in which humans and automated components of digital avionics systems interact to accomplish the tasks involved in successfully completing a flight. They must also consider the limitations of human performance. While automation has eased the workload of the flight crew in certain situations, it has introduced new tasks and new possibilities for error.

In light of the success of automation and the limitations of human operators, many people question the necessity and desirability of including the human operator on the flight deck. Despite the temptation to declare the human operator an anachronism on the modern flight deck, the human operator is a necessity, because of the many dynamic, and often chaotic, elements of the commercial airspace system. Weather on the ground and in the stratosphere, terrain, large numbers of flights from a variety of carriers, interactions in and around airports, and many different aircraft types all contribute to the complexity of the system.

These factors make it extremely difficult to model and fully automate commercial air transportation and make human operators, both on the ground and in the air, an essential element of the commercial airspace system. Additionally, the ability of human operators to reason from first principles when faults occur allows the flight to be completed safely despite the present of faults[7].

Given the necessity of the human operator, future automation must be designed not with the purpose of replacing the human, but rather assisting the human in performing his or her job. Additional justification of this approach, known as human-centered design, is presented, followed by a brief survey of past and present systems, which give an overview of the current state of practice in industry and future directions for human-centered design.

1. Background

1.1. Current Design Practice

The flight-deck design process is, for the most part, unwritten, complex, and variable across manufacturers[5, 7]. The design process relies heavily on the knowledge and experience of individuals rather than written standards.

Hardware and software integration is the primary goal of flight-deck design; human factors is given little consideration. There is a mismatch between human and system behaviors and, often, the human elements of the aviation system become subservient to the automation[38].

Many manufacturers adopt an incremental design process due to retrofits and government-mandated safety systems. This incremental approach can lead to the disintegration of the design process. Aircraft manufacturers who adopt an integrated and extensible avionics architecture and design process can reduce the extent of the disintegration. However, industry today is reluctant to borrow from modern research due to the risk and the lengthy (and costly) certification process associated with novel designs, and is thus constrained by past design processes. This reluctance to introduce a new design process hinders human-centered design philosophies from becoming a reality in today's flight decks.

Flight Deck Evolution

Modern flight decks have evolved to reflect technological advances [5] and are grouped into generations based on capability, rather than date of manufacture.

Classic Flight Decks

The most basic flight deck is referred to as the classic flight deck. These flight decks lack sophisticated automation and have a relatively simple autopilot. They have simple caution and warning systems that indicate system failures. Each instrument displays a single piece of data; such dedicated displays increase pilot workload by comparison with integrated data displays, because in the former, information consolidation is the responsibility of the crew.

Glass Cockpits

Glass cockpits represent the next step in flight-deck evolution. Glass cockpits are named after cathode ray tubes (CRTs) that, along with some dedicated instruments, display flight information. However, instruments are still used to display primary flight information. Glass cockpits introduce a map display integrated with the flight management system (FMS). This innovation allows the pilots to view programmed flight paths mapped onto the ground. Glass cockpits also introduce more advanced autopilots and an integrated caution and warning system that groups system failures into hierarchies. This feature is an example of an integrated data display, which reduces pilot workload in comparison to dedicated displays.

Second-Generation Glass Cockpits

Second generation glass cockpits are the current generation of modern flight decks in commercial aircraft. CRTs and liquid crystal displays (LCDs) are more prevalent in this generation of flight decks and are used to display primary flight information (attitude, altitude, airspeed, etc.) as well as secondary flight information (navigation, terrain, weather, etc.). This generation of glass cockpits also supports some integration of the flight management system and the autopilot. Levels of automation vary from manufacturer to manufacturer and from aircraft model to aircraft model. The way in which data is displayed and presented to the pilot has advanced to include menus and cursors as well as displays that indicate system failures graphically.

1.2. Cognitive Considerations

In order to make modern flight safer, more comfortable, and continually efficient, an increasing number of automated processes have been introduced with the intent of reducing operator workload and ensuring that the operator or crew are not overwhelmed by task requirements. While these automated processes generally provide an operational benefit and solve existing problems, they may also create or induce additional, unforeseen problems [5,43]. Consequently, the fundamentals of human factors engineering must be considered so that the net result of introducing digital systems or automating existing processes is positive.

Campbell describes not only the physiological limitations that human operators face in the cockpit, but the psychological and cognitive shortcomings of humans that affect perception, information processing, planning, decision-making and execution[12]. Some of these limitations are:

- bounded short-term and working memory,
- decreasing attention and vigilance during periods of non-optimal arousal,
- susceptibility to visual or auditory illusion and misperception, and
- degraded decision-making ability resulting from incorrect mental models, motor function, and lack of or excessive feedback.

As a result of these and other limitations, we must consider the role that the human should play in directing flight operations on the ground and in the air. Current commercial implementations utilize many of the same automation components and systems, but with different parameters or constraints. In this section, we address some of the most common components of modern avionics by describing the motivation for and evolution of instrumentation and information displays. We then assess the role of automation in enhancing, and in some cases degrading, operator situational awareness.

Situational Awareness

Abbott defines *situational awareness* (SA) as "the perception on the part of the flight crew member of all the relevant pieces of information in both the flight deck and the environment, the comprehension of their effects on the current mission status, and the projection of the value of these pieces of information into the near future"[5, p.9-4]. Subsequently, she attributes many aviation accidents and incidents to the loss of SA. While situational awareness can be generalized as above, Endsley proposes three distinct levels to SA: perception of elements in the environment, comprehension of the current situation and projection of future status [17]. Furthermore, there are five elements, or sub-components, of SA that must be considered:

- **Geographical SA**: the relationship between the aircraft and terrain features, political borders or other points of reference;
- **Spatial and Temporal SA**: a broader term, which applies to location of the aircraft in a three dimensional space over a period of time;
- **System SA**: knowledge of how the functional components of the aircraft are performing;

- **Environmental SA**: the way in which the elements of nature affect the aircraft in flight and on the ground;
- **Tactical SA**: the immediate operations of the aircraft, usually bounded by short-term parameters (i.e., flight, mission, leg, etc.).

Each of these subcomponents corresponds to a different type of automation that has been introduced into the flight deck. These subcomponents include Traffic alert and Collision Avoidance System (TCAS), Enhanced Ground Proximity Warning System (EGPWS) and Radar Warning Receivers (RWR), all of which are intended to improve the operator's SA. Loss of SA can impact flight operations to varying degrees, including catastrophic errors that result in controlled flight into terrain (CFIT) accidents. This tragic, yet mostly avoidable, phenomenon can be induced by excessive workload relating to the cognitive processes associated with maintaining situational awareness.

1.3. Considerations of Human Error

Human error is cited as a primary cause in many aircraft accidents [5,19,28,33]. It is clear that careful consideration of human error is necessary if the safety of commercial air transportation is to be increased and if the risk introduced by new automated systems is to be understood. In this section we present a cognitive model for human error. We consider how automation, while it has been introduced onto aircraft flight decks in an effort to remove human error, has actually led to new forms of human error. We then discuss error management, and conclude that neither adding new automation nor managing errors is sufficient to deal with human error on the flight deck.

A Cognitive Model for Human Error

Reason presents a systematic description of human error with a basis in cognitive psychology[35]. He proposes three classes of errors that correspond to the following three levels of behavior:

- **Skill-based**: sensory-motor behavior directed by a statement of intent and characterized by nearly unconscious execution;
- **Rule-based**: if-then sequences where environmental conditions are compared against familiar patterns and used to determine a particular course of action;
- **Knowledge-based**: high-level behavior invoked in unfamiliar circumstances. Plans of action are formulated through trial and error where selection is based on projected effectiveness in reaching a goal, or reasoning about a solution from conceptual understanding of the environment[35,24].

At the skill-based level, two types of error are possible: slips and lapses. A slip is an incorrect execution of a correct action, for example, entering a heading of 57 degrees when the intended heading was 75 degrees. A lapse is an omission of a correct action usually following an interruption, for example, forgetting to enter a heading change due to an incoming communication from ATC. Both of these errors are errors in which the correct action was known but not executed[35].

Mistakes are characterized by incorrect action taken on the part of the human, and they fare present at both the rule-based and knowledge-based levels. Rule-based mistakes often come about because of a change in the environment that was not detected. For example, the human attempts to apply a rule believing the environment to be in a particular state and is not made aware of the environmental change until the rule is seen not to work as anticipated[24].

When operating at the knowledge-based level, humans must "think outside the box" and experiment with solution techniques in an attempt to learn about the state of the environment and solve problems. While making these attempts, humans monitor the effects of their actions and constantly attempt to recover from errors. Knowledge-based errors—mistakes—can be viewed as "a lack of recovery from the unacceptable effects of exploratory behavior"[24, p.106]. Moreover, it would be impossible to eliminate these types of errors and still expect humans to cope with emergency situations in which automation is incapable

of handling the emergency. These errors are the natural by-product of the human problem-solving process: "human errors are the inevitable side effects of human adaptability and creativity and are very different from faults in technical components" [24, p.106].

Automation Produces New Errors

Flight-deck automation is becoming increasingly complex. Much of the complexity and functionality of the automation has been put into place with the belief that, since humans are error-prone, they should be replaced by machines[24]. Ironically, the increased complexity of the automation has made it increasingly difficult for the operator to keep track of what the automation is doing. Weiner and Curry note:

> The question today is not whether a function can be automated, but whether it should be, due to various human factors issues. It is highly questionable whether total system safety is always enhanced by allocating functions to automatic devices rather than human operators, and there is some reason to believe that flight-deck automation may have already passed its optimum point[43].

It is interesting to note that these concerns were raised in 1980.

The inclusion of complex automation in modern flight decks has led to an important new source of error: mode confusion[19,28] (see Chapter5, Section3). The growing number of modes in which the automation may be at any given time makes it more difficult for pilots to keep track of the current mode of the system[25]. At some point the automation may transition to a new mode without the human operator being aware that the mode has changed. In the event of an emergency, this can lead to incorrect actions which exacerbate the situation.

Modes define mutually exclusive sets of system behavior, and are a property of the design of the system. Due to the complexity of the automation and because they are not experts in computer systems, pilots are forced to form abstract notions of how the automation is working. These mental models allow and are part of the natural mechanism by which pilots cope with the complexity of their environment[24]. The mental models pilots form correspond to their training and experience, and they may be very different from the reality of the state of the automation[24]. This disparity in perceived and actual state makes it difficult for the pilot to accurately understand what the automation is doing. The automation may transition to a new mode without the pilot's knowledge, leading to possibly catastrophic consequences.

While simply including additional automation increases the potential for mode confusion, careful analysis and design can all but eliminate sources of mode confusion[19, 25, 28].

Error Management

Abbott suggests that human error should be "managed" and presents a definition of error management that follows the concept of fault tolerance (see Chapter9, Section2.2). It includes error avoidance, error detection, and error recovery[5]. Achieving this in practice is difficult at best.

Error Detection

Reason[35] identifies three sources of error detection: self-monitoring, environmental error cuing, and error detection by other people. The difficulty in error detection is apparent when each of these sources is examined. Self-monitoring is crucial to performance, particularly when operating at the knowledge-based level. However, as mentioned above, errors in human-machine systems, particularly those errors that occur during the exploratory phases of knowledge-based problem solving, are most likely to appear when self-monitoring has broken down. Thus, relying on self-monitoring as the sole form of error detection in human-machine systems is unacceptable. Error detection by other people may be possible in some situations: there are always two people on a commercial flight deck, a pilot and a copilot. They can be seen to function as a self-checking pair. However, in the event of a crisis, it is likely that each will be performing different jobs in parallel, in order to reduce the overall workload placed on one person. In these circum-

stances, it is unrealistic to expect that each will be carefully monitoring the actions of the other. Furthermore, seeking to detect human errors with automated assistants (for example,[37,10,32]) is also likely to be problematic during an emergency, as the human is working "outside the box" to save the system, and hence is beyond the purview of the standard operating procedures by which automated assistants typically judge human performance. Finally, Reason[35] notes that, irrespective of error type, the rate of error detection drops significantly as task complexity increases. In one study, the *correction* rate for speech errors was 64% (Reason suggests that the detection rate would have been much higher), while the detection rate for human errors in simulated nuclear power plant emergencies was only 38%.

Error Recovery

Environmental cues aid both the detection and correction process. Often, environmental cues come from "forcing functions"[35] which prevent further progress toward solving a problem until the error has been corrected. In designed systems, there are six possible responses to human error:

- **Gagging**: the system prevents the operator from entering impossible values.
- **Warning**: the system warns the operator that an activity is not legal.
- **Do nothing**: the system simply takes no action when given illegal input, requiring the operator to determine the source of the error.
- **Self-correct**: the system attempts to determine what the operator actually intended to do, and takes that action instead of the commanded action.
- **Let's talk about it**: the system asks the operator what the intent was behind the erroneous input.
- **Teach me**: the system asks the operator to tell it what it should do given input of the type received[35].

Clearly, some of these are inappropriate for responding to errors on a flight deck ("teach me" and "do nothing"), and others must be handled with extreme caution. For example, a self-correcting system must be built with extreme care to prevent the system from taking some other inappropriate action that causes the original error condition to be compounded rather than mitigated and would likely lead to additional instances of mode confusion. Likewise, "let's talk about it" must conduct the human-machine dialogue in a manner that does not further exacerbate the situation by distracting the pilot from his or her job.

As with design faults, the exact nature of human error when it manifests in a human-machine system is unlikely to be accurately predicted when the system is designed. Indeed, were the exact types of error anticipated, the design of the system could be changed to make their appearance highly unlikely, if not impossible.

Error Avoidance

Given the difficulty posed by error detection and error correction, error avoidance is the only remaining solution to human error. Human error cannot be entirely avoided, however: as mentioned previously, simply striving to design the pilot out of the cockpit by including more and more advanced automation has, rather than solving human error problems, generated new sources of human error.

Rather than attempting to build further complexity into automated systems in an attempt to protect the automation from errors caused by the human operators, "the goal of human-machine interface design should be to preserve the human capability to intervene positively while making harmful intervention as difficult as possible" [24, p.126]. This is the goal of human-centered design.

1.4. Human-Centered Flight-deck Design and Philosophy

Human-centered design philosophy affects components of the current flight-deck design process as well as the overall process itself. Current design practice places importance on the performance of individual components, while human-centered flight-deck design places importance on the integrated performance of the crew with the flight deck. Human-centered design also advocates that the test phase be integrated with the design process itself, instead of being done at the end of the process. Early and continuous testing enables a constant evaluation of preliminary concepts in the design[18].

Human-centered design begins with the acknowledgment that the flight crew, fight deck, and aircraft are part of a much larger transportation system. It is a top-down, requirements-and-systems driven approach that addresses the interaction between the flight crew and the aircraft. It also seeks to elevate the importance of design issues to the same level as technological issues such as hardware performance and reliability[18].

Human-centered design principles dictate that the pilot is always in command of the aircraft, even when automation is engaged[7]. It also dictates that the flight crew is the most critical aspect of the flight deck. Therefore, each design decision must consider overall flight safety and crew performance. The design philosophy emphasizes integration of the flight crew with the flight deck to optimize the interaction between the two. It views humans and automation as complementary, not comparable, entities. Additionally, automation should support the many roles of the pilot, specifically, the pilot's roles as team member, commander, individual operator and occupant[18].

Function allocation—deciding what roles to assign to the automation and what roles to assign to the human—is at the heart of human-centered, flight-deck design. Ideally, the crew should only be involved in decisions that affect the high-level mission of the flight, and tedious, repetitive tasks should be automated. However, this component of the design philosophy presents a challenge to designers since they must identify the information of which the pilot needs to be aware making high-level decisions. Another challenge that designers face is to engage the pilot sufficiently so that he or she does not lose situational awareness, while not significantly increasing his or her workload.

Task-oriented displays are also an integral part of human-centered, flight-deck design and are based on the idea of function allocation. They present information to the pilot after combining it for him or her, rather than presenting raw data that the pilot has to integrate.

2. Example Systems

2.1. Displays and Information Automation

As aircraft have become increasingly complex, a need has arisen for automation capable of distilling information about the state of the aircraft into a form that can be more readily managed by the crew. While the development of automated information display (information automation) began as an evolutionary process, the advent of digital technology brought radical changes to the way in which information is managed and displayed.

Reising discusses the evolving eras of instrumentation and displays [21]. These are the mechanical era, the electro-mechanical era, and the electro-optical era. In the mechanical era, instruments were powered by pressure differentials resulting from motion through the atmosphere. The electro-mechanical era introduced electric-powered, analog instruments and the detailed research regarding their optimal design, layout and alignment along the primary flight display (PFD). The electro-optical era has seen the introduction of CRTs, LCDs, and other screen-projected devices, which are generally classified as multi-function displays (MFDs). Reising attributes this evolution to the improved durability and reliability, superior perceptual attributes, and compact, flexible form factor of modern display technologies[21]. Also, treatment is given to the potential for operator overload or confusion, based on the massive amount of information that is available for display[21].

Early Subsystem Displays

Subsystems on early jet aircraft were controlled with switches and monitored by discrete analog displays. The large number of displays and switches necessitated the introduction of an additional member of the flight crew, the flight engineer, who was responsible for monitoring and operating hydraulic, electrical, fuel, power and pneumatic systems. In the event of a subsystem failure, the flight engineer was responsible for determining which subsystems needed to be shut down or adjusted so that the aircraft could continue safe flight under most optimal conditions.

For a system as complex as even early jet aircraft, anomaly detection and troubleshooting could present the user with a nearly impossible task. Often, a flight engineer would have to analyze and predict the interaction between several control variables, and choose an adequate corrective action or sequence. The difficulty of accomplishing this successfully was magnified by a display and control panel that violated almost every characteristic of good process control and information display[44]. In an effort to reduce pilot workload on aircraft with a flight crew of two, cockpits were simplified through the automation of subsystem control. For example, load-shedding (shutting down some subsystems to conserve power) following a reduction in power generation was automatic, as was deactivation of air-conditioning systems following an engine loss. This proceeded in an ad hoc fashion, on a system-by-system basis [11].

Subsystem Information Automation

The advent of digital technology and its subsequent inclusion on the flight deck provided designers with near-limitless flexibility of information display[11]. The large display screens in glass cockpits, first introduced in the Airbus A-310, facilitate the distillation and synthesis of complex data and its presentation to the crew.

Power Control

The removal of the flight engineer necessitated new automation to manage the power plants on multi-engine aircraft. *Full authority digital engine controllers* (FADECs) control fuel flow to balance engine thrust and optimize engine performance. The term “Full Authority” indicates that the crew has no ability to control the engine directly, aside from shutting it down[31].

In order to keep the crew informed of the status of the engines, Engine Indication and Crew Alerting Systems (EICASs) have been installed on Boeing aircraft since the B-747-400[11]. Departing from the individual analog dials on earlier aircraft, the EICAS uses a tape display (similar to a bar graph), which unifies engine display, making it easier for crew members to monitor engine performance.

Synoptic Subsystem Displays

Many aircraft with glass cockpits provide synoptic information about aircraft subsystems. Rather than giving full information to the crew in the event of a fault in a subsystem, data is condensed, analyzed, and provided in a summarized form.

On the MD-11, Douglas (which became part of McDonnell-Douglas and is now part of Boeing) chose to automate the reconfiguration process whenever possible. When an abnormal condition occurs, rather than alerting the crew of the condition through synoptic displays and waiting for the crew to handle reconfiguration according to the checklist, the automation reconfigures the aircraft subsystems automatically. The synoptic information alerts the crew to the fault, the reconfigured status of subsystems, and the implications of the reconfiguration for the remainder of the flight. While such abstractions result in a decrease in overall workload, they may increase cognitive demand due to the opacity of the automation[11], or create the potential for mode confusion.

Synoptic displays are included on the Airbus A320, A330 and A340, as a primary means of subsystem feedback. Many of the checklists the crew must follow are also accessed through synoptic displays. In Boeing aircraft, on the other hand, synoptic displays provide the crew with supplemental information related to a subsystem failure.

Considerable debate exists over the appropriateness of including synoptic displays as opposed to making full information available to crew members[11]. Aircraft designers take great care to develop procedures that must be followed to reconfigure the aircraft correctly in the event of a subsystem failure. They feel as though giving too much information to crew members might encourage inventiveness on the part of the crew when faced with a failure. Crew members, however, feel as though they should have as much information as possible available so that they can attempt to cope with faults and failures unforeseen by the aircraft designers.

While it is reasonable to suppose that not all failure conditions will be foreseen by system designers, it is not clear that making additional information available to operators during an emergency will allow the operators to correctly respond to faults. In particular, considering the low rate of error detection in stressful circumstances related to complex systems[35], it seems unreasonable to suspect that crew members will have the necessary time and expertise to solve the complex problems that arise from subsystem failures on aircraft. Moreover, as modern flight is an example of a complex, tightly-coupled system[33], there is little time or room for the kinds of mistakes likely to be made during the natural problem-solving strategies employed by humans [24].

Data Presentation of the Future

Advanced information automation of the future attempts to address some of the limitations of current approaches by injecting additional reasoning capabilities into the automation.

Power Control

Research carried out at NASA Langley Research Center suggests that an improved EICAS might contain a mathematical model predicting engine output based upon current conditions[6]. The system, called the Engine Monitoring and Control System (E-MACS) displays both the measured thrust as well as the predicted optimal thrust, enabling the crew members to easily detect engine anomalies. One potential risk with employing this approach is that the accuracy of the display is tied to the correctness of the mathematical model.

Diagnostic and Warning Systems

While synoptic displays have greatly reduced the amount of information crew members are required to process in the event of a fault, there is room for considerable improvement in the automation of warning and diagnostic information on aircraft. Billings relates an incident in which the crew was bombarded with 42 EICAS messages, 12 warnings, and repeated stick shaking[11]. It has been suggested that the safe landing of the flight in question was only accomplished due to the presence of two additional crew members on the flight. More advanced diagnostic and warning systems could mitigate this problem.

Current diagnostic and warning systems are rule-based systems [11]. Reasoning is done at the level of: "if condition one and condition two hold, then raise some warning message." Unfortunately, this type of system degrades human-machine performance, as the human is not privy to the reasoning the machine is following. Furthermore, the system must frequently stop and ask the human for information relating to the current state of the world in order to continue its reasoning. In time-critical situations, this is often unacceptable.

Model-based diagnostic systems perform deeper reasoning based on a model of the system in question, identifying discrepancies in actual values and expected results [11]. These systems are much more robust as they are not tied to predefined rule sets for operation. Unfortunately, most model-based systems cannot reason about uncertainty and also lack the ability to differentiate symptoms differing in rate of manifestation (e.g. a rapidly oscillating fan speed and a slowly decreasing fan speed). Moreover, as with all reasoning systems of this type, the number of possible symptoms in complex systems explodes rapidly and pruning this space is difficult.

2.2. Advanced Display Technology

Display technologies have evolved from the presentation and management of information to providing the pilot with additional information in order to heighten situational awareness. Enhanced and synthetic vision systems are being developed to supplement (in the case of enhanced vision) and even replace (in the case of synthetic vision) the pilot's view through the windscreen at night and during Instrument Meteorological Conditions. We present enhanced and synthetic vision in this section. EGPWS is also briefly presented for the purposes of making a distinction between it and enhanced vision.

Enhanced Vision

Enhanced vision is a technology used to visually supplement the pilot's view through the windshield. It enables the pilot to view obstacles and terrain ahead of him or her when the view is obstructed, and it increases the pilot's situation awareness in all conditions. Weather conditions (such as rain and fog) along with environmental conditions (such as darkness) are prime motivators for the use of enhanced-vision technology[36].

Enhanced vision increases the situational awareness of the flight crew, reducing the risk of runway incursions and CFIT accidents, and it aids the crew during takeoff, approach and landing[22]. It is particularly useful when attempting Category I and *non-precision approaches*, those approaches "where no electronic descent path guidance is available and pilots must fly a series of progressively lower 'steps' to reach the runway" [36]. The Korean Air flight 801 crash in August, 1997 at Agana, Guam that resulted in the deaths of 228 passengers and crew members is thought to be a result of an attempted non-precision approach[3].

Most commercial enhanced-vision systems uses infrared radiation to obtain data that supplements the pilot view by displaying the infrared image on top of the visual image. Infrared radiation is partitioned into short-range (1-2 microns), medium-range (3-5 microns), and long-range (8-14 microns). All objects emit infrared radiation, but the wavelength of radiation emitted depends on the temperature of each object. For example, approach and runway lights are picked up very well by sensors that operate in the short range but very faintly at the medium and long ranges. Due to this phenomenon, systems equipped with sensors that operate at multiple-micron ranges are most effective for enhancing pilot vision.

Enhanced-vision systems also utilizes image processing technology. Typical image processing functions in enhanced-vision systems include non-uniformity correction, auto-gain, and various other enhancements. After the raw sensor data has been processed, the images from each of the sensors are consolidated and presented to the flight crew [36].

Enhanced Vision Displays

Traditionally, Head Up Displays (HUDs) are used to display enhanced-vision information to the flight crew. HUDs project enhanced data onto the pilot windshield. However, HUD technology is costly and complex. In particular, two HUD systems that utilize enhanced-vision systems cost an additional $500,000 due to their complexity. Compact, conventional head-down displays, such as CRTs, are an alternative to using HUDs for display purposes. This type of enhanced-vision system costs about $120,000[36].

Both HUDs and head down displays need to be certified by the Federal Aviation Administrations (FAA) when used in commercial aviation. However, only HUDs need to be certified by the FAA in corporate aviation. Head-down displays are not subject to FAA certification standards in corporate aviation since the FAA regards head-down enhanced-vision systems in the same category as television displays. However, the head-down display cannot be used for primary guidance whereas the HUD can[36].

Superimposed supplementary information has been shown to lead to visual and attentional fixations under certain conditions. These fixations occur when pilots are less likely to process other information. However, when supplementary information is integrated with the standard visual scene, the pilots are less likely to suffer from visual and attentional fixations [20]. Therefore, display technology should try to integrate as much of the data as possible before rendering it.

Enhanced-vision systems are in greater demand by smaller private planes that cater to the corporate community than in commercial aviation. This is due, in part, to the nature of corporate jets, which fly into small airports lacking sophisticated runway landing equipment[34]. Bombardier, Cessna and Gulfstream are examples of manufacturers of aircraft equipped with enhanced-vision systems. Boeing's 767 is also slated to be equipped with an enhanced-vision system in the near future[36].

Synthetic Vision

Synthetic vision systems anticipate using geographical databases, navigation systems and flight guidance displays to replace the pilot windshield[23]. Instead of the windshield, the pilot would rely solely on visualizations generated from databases for situational awareness and context. This pure model of synthetic vision has several potential flaws. The most apparent is that pilots will be blind to dynamic elements in the environment, such as other aircraft and weather. Another potential flaw is that terrain databases are error prone and relying on data from them alone seems an unnecessary risk.

An alternative to this plan for synthetic vision is a hybrid approach, in which information from geographical databases would be augmented with measurements from a radar-altimeter and information from an imaging sensor[40]. In this approach, dynamic elements in the environment would be visible to the pilot.

The Enhanced Ground Proximity Warning System and Enhanced Vision

Honeywell's EGPWS, has reduced the risk of CFIT by about a factor of a 100 since 1974[4]. EGPWS utilizes an on-board terrain database that graphically displays the terrain around the aircraft to the flight crew[8]. If by following its present course the aircraft is in danger of crashing into terrain within the next several minutes, EGPWS issues audio warnings to the flight crew. In general, most pilots respond within two seconds of a warning issued by EGPWS when flying under instrument meteorological conditions[4]. The effectiveness of EGPWS relies heavily on the integrity of the on-board database, and the possibility that EGPWS can provide faulty data is quite feasible.

While EGPWS may prevent the majority of CFIT accidents from occurring, it does not address the dangers encountered when carrying out non-precision approaches, runway incursions, approach, takeoff and landing. For example, the most catastrophic accident in aviation history occurred when two Boeing 747s collided near Tenerife, Canary Islands, Spain in March 1977 killing a total of 583 people[3]. This accident was due, in part, to a loss of situational awareness because of foggy conditions. A more recent accident, also a result of poor weather conditions, was the Singapore Airlines crash of October 2000 where a Boeing 747 crashed into construction vehicles parked on a runway closed for repairs [3]. EGPWS technology would not have prevented either of these accidents or others like them since terrain was not a factor leading to the crash. Instead, an enhanced-vision systems could have improved visibility for the flight crew given the poor weather and might have prevented the accident.

An integrated system, including EGPWS and enhanced vision, would be a good candidate solution for preventing a combination of CFIT and accidents similar to runway incursions. A 1990 study provides further argument in favor of integrating enhanced-vision systems with modern avionics systems. This study shows that "nonterrain objects were recognized faster with IR imagery than when viewed with television"[20]. Aircraft equipped with enhanced-vision systems would allow faster pilot recognition of non-terrain hazards than aircraft without enhanced-vision systems.

2.3. Automated Pilot Assistants

There has been significant research over the past 20 years in the design and application of computer assistants to pilots. These assistants, rather than taking over tasks from human operators, are meant to complement human activity by providing warnings of deviations from inferred goals and supporting operators

through the delegation of functions during periods of high cognitive workload. We present three such systems here, which, though research prototypes, offer promising results.

Operator Function Model—OFMspert

OFMspert is a system designed to enhance human abilities and overcome human limitations, rather than replace human operators[37]. Using a model of correct operator procedures called an Operator Function Model (OFM), OFMspert attempts to infer operator intent by opportunistically matching operator actions and system state output with goals specified in the OFM. By matching actions to goals opportunistically, OFMspert allows operators to solve problems in whatever order and by whatever path they choose, as long as operator actions support the set of inferred goals. When operator actions do not support any goals, OFMspert notifies operators that it does not understand their actions, thereby potentially alerting operators to erroneous actions. In this manner, OFMspert complements human preferences and strengths.

More complete implementations of assistant systems would provide OFM-based system control, as well as intent inferencing. Such a system would allow the operator to delegate tasks to the automation during periods of high workload. The OFM-based approach to delegation is advantageous, as it facilitates operator understanding of the automation's behavior[29].

OFMspert has been generalized so as to be domain independent[41], has been applied to satellite control [29], and has been evaluated with promising results on the flight deck of a 727[45].

Hazard Model—HM

HM, the Hazard Monitor, aids the operator by providing information about unmet expectations in the satisfaction of an inferred goal[10]. Similar to OFMspert, HM compares expected system states to operator actions and alerts the operator when operator actions do not meet expected system states. HM has been shown to be useful in detecting data-entry and procedures errors in both military and commercial aviation.

Cockpit Assistant—CASSY

The cockpit assistant CASSY incorporates intent inferencing similar to that of OFMspert with speech recognition and both interactive and human-guided mission replanning support[32]. CASSY aids pilots in flight-plan generation for either portions of or the complete flight plan and evaluation of alternative airports for emergency landings. It monitors pilot behavior, checking for adherence to the flight plan, correct configuration of flaps, gear, navigation settings, etc., and flight envelope violation. The speech recognition interface allows the pilot to change the configuration of CASSY without requiring the use of an MCDU or similar input device.

CASSY has been evaluated in the experimental cockpit of a testing aircraft, in which a number of regional flights in high traffic areas were performed. CASSY was able to successfully identify instances of pilot error about 80% of the time. Most pilot errors were safe, but less than perfectly efficient maneuvers; three had the potential to impact safety.

References

[1] History of the army aviation warrant officer. www.usawoa.org, 1998.

[2] Kollsman.com. www.kollsman.com/avievswindow.html, 1999.

[3] Airdisaster.com. www.airdisaster.com, 2004.

[4] Enhanced ground proximity warning system. www.egpws.com, 2004.

[5] Abbott,K.H. "Human Factors Engineering and Flight Deck Design." In C. Spitzer, editor, *Avionics Handbook*, Chapter 9. CRC Press, 2001.

[6] Abbott,T.S. "A Simulation Evaluation of the Engine Monitoring and Control System Display." NASA-TP-2960, Feb. 1, 1990.

[7] Arbuckle,P.D, K.H.Abbott, T.S.Abbott, and P.C.Schutte. "Future flight decks." *Proceedings of the 21st Congress International Council of the Aeronautical Sciences*, 1998.
[8] Bailey,I. "Honeywell EGPWS, Getting the Job Done — Part 1." *Avionics News*. March, 2004.
[9] Barbacci,M, P.Clements, A.Lattanze, L.Northrop, and W.Wood. *Using the Architecture Tradeoff Analysis Method (ATAM) to Evaluate software architecture for a product line of avionics systems: A case study*. Technical report, Carnegie Mellon University - Software Engineering Institute, 2003.
[10] Bass,E.J, *et al.* "Architecture and Development Environment of a Knowledge-based Monitor that Facilitates Incremental Knowledge-base Development." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 34, No. 4.
[11] Billings,C.E, *Aviation Automation: The Search for a Human-Centered Approach*. Lawrence Erlbaum Associates, Mahwah, NJ, 1997.
[12] Campbell, R. and M. Bagshaw. *Human Performance and Limitations in Aviation*. Blackwell Science, Oxford, 2002.
[13] Carroll,J. "Blackhawks Have History of Crashes, Problems." *Courier-Journal*, Louisville, KY, April 23, 1999.
[14] U.S.A.A.Center and School. *Aircrew Training Manual Utility Helicopter UH-60/EH-60 - Training Circular 1-212*. Headquarters, Department of the Army, Washington, DC, 2002. [12]
[15] U.S.A.J.F.K.S.W.Center and School. *U.S. Army Special Operations Aviation Aircrew Training Program Training Circular 1-210-1*. Headquarters, Department of the Army, Washington, DC, 2002.
[16] U.S.A.F.S.O.Command. *AC-130 Aircrew Evaluation Criteria - Air Force Instruction 11-2AC-130*. Department of the Air Force, Washington, DC, 2000.
[17] Endsley,M. "Situation awareness in aviation systems." In J.W.D.Garland and V.D.Hopkin, editors, *Handbook of Aviation Human Factor*s. Lawrence Erlbaum Associates, 1999.
[18] M.T.P. *et al. A Crew-Centered Flight Deck Design Philosophy for High-Speed Civil Transport (HSCT) Aircraft*. NASA Technical Memorandum 109171, 1995.
[19] Butler,R.W. *et al. A Formal Methods Approach to the Analysis of Mode Confusion*. Technical report, NASA, 1998.
[20] Foyle,D.C, A.H.Ahumada, J.Larimer, and B.T.Sweet. "Enhanced/Synthetic Vision Systems: Human Factors Research and Implications for Future Systems." *SAE Transactions: Journal of Aerospace*, 101, 1734-1741, 1992.
[21] Reising,K.L.J, and R.Munns. "Controls, Displays and Workplace Design. In J. W. D. Garland and V.D.Hopkin, editors, *Handbook of Aviation Human Factors*. Lawrence Erlbaum Associates, 1999.
[22] Kerr, J. R, C. H. Luk, D. Hammerstrom, and M. Pavel. "Advanced Integrated Enhanced Vision systems." *SPIE Aerosense*, April 21-25, 2003, Orlando, Florida, 2003.
[23] W. J. Kubbat, P. M. Lenhart, and H. V. Viebahn. 4d flight guidance displays, a gate to gate solution. 17th DASC-Electronics in Motion, 1998.
[24] N. G. Leveson. Safeware: System Safety and Computers. Addison-Wesley, New York, 1995.
[25] N. G. Leveson and E. Palmer. Designing automation to reduce operator errors. In Proceedings of Systems, Man, and Cybernetics Conference, 1997.
[26] H. Lowery and B. Mitchell. Mission computer replacement prototype for special operations forces aircraft: An application of commercial technology to avionics. In Proceedings of the 19th Digital Avionics Systems Conference, October 2000, pages 4.E.1–1 – 4.E.1–6.
[27] J. A. Luke, D. G. Halderman, and W. J. Cannon. A cots-based replacement strategy for aging avionics computers. CrossTalk: The Journal of Defense Software Engineering, December 2001, 14-17, 2001.
[28] S. P. Miller and J. N. Potts. Detecting mode confusion through formal modeling and analysis. Technical Report NASA/CR-1999-208971, National Aeronautics and Space Administration, 1999.
[29] Mitchell,C.M, *et al.* "OFMspert I: Operations Automation." *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*. Oct., 2000, Vol. 2.

[30] G. Miyahara and C. P. Satterthwaite. A multiplatform support environment. In Proceedings of the IEEE 1997 National Aerospace and Electronics Conference, pages 815–819.
[31] I. Moir and A. Seabridge. Aircraft Systems: Mechanical, Electrical, and Avionics Subsystems Integration. AIAA, Reston, VA, 2001.
[32] R. Onken, "The cockpit assistant system CASSY as an on-board player in the ATM environment," presented at the US/Eur. ATM R&D Seminar, Paris, France, June 1997
[33] C. Perrow. Normal Accidents: Living with High-Risk Technologies. Princeton University Press, Princeton, New Jersey, 1999.
[34] J. W. Ramsey. Weathering fog and darkness–huds with enhanced vision. www.defensedaily.com, 2004.
[35] J. Reason. Human Error. Cambridge University Press, 1990.
[36] W. Reynish. Enhanced vision: Final link in situational awareness. www.defensedaily.com, 2004.
[37] Rubin,K.S, P.M.Jones, and C.M.Mitchell. "OFMspert: Inference of Operator Intentions in Supervisory Control Using a Blackboard Architecture." *IEEE Transactions on Systems, Man, and Cybernetics.* Vol. 18, No. 4.
[38] P. C. Schutte and K. F. Willshire. Designing to control flight crew errors. NASA Technical Memorandum, 1997.
[39] K. Smith and G. Miyahara. Leveraging an avionics support environment for shared application to multiple platforms. In Proceedings of the 16th Digital Avionics Systems Conference, October 1997, pages 1.3–9 – 1.3– 15.
[40] E. Theunissen, F. D. Roefs, R. M. Rademaker, and T. J. Etherington. Integration of information in synthetic vision displays: Why, to what extent and how? 2003.
[41] Thurman,D.A, A.R.Chappell, C.M.Mitchell. "An Enhanced Architecture for OFMspert: A Domain-Independent System for Intent Inferencing." *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics.* Vol. 1, Oct., 1998
[42] M. B. Tischler, J. D. Colbourne, M. R. Morel, D. J. Biezad, K. K. Cheung, W. S. Levine, and V. Moldoveanu. A multidisciplinary flight control development environment and its application to a helicopter. IEEE Control Systems Magazine,19, 22-33, 1999.
[43] Weiner, E.L. and R.E.Curry. "Flight-Deck Automation: Promises and Problems." *Ergonomics*. Vol. 23, No. 10, 1980.
[44] C. Wickens and J. Hollands. Engineering Psychology and Human Performance. Prentice Hall, Upper Saddle River, NJ, 1999.
[45] Verfurth,S.C, T.Govindaraj, C.M.Mitchell. "OFMspert for the 727: An Investigation into Intent Inferencing on the Flight Deck." *Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 2, Oct., 1991.

CHAPTER 9

Dependability in Avionics Systems

John Knight Rajat Tikoo Lori Stotler

Avionics systems rely on computing platforms, and these platforms must be designed to provide the required levels of safety to passengers, crew, and maintenance staff. The avionics functions must be sustained appropriately in order to ensure a safe flight, yet the hardware components on aircraft operate in a hazardous environment while running software that itself might contain defects.

Many techniques have evolved for constructing dependable computer platforms for avionics systems. Architectures have been developed that use various forms of redundancy and reconfiguration to allow continued operation when components fail. In addition, in many cases replicated components are separated within an airframe to prevent their simultaneous loss in the event that there is damage to the airframe. Various techniques are employed to aid in the correct construction of software, and software development is required to follow a rigorous process. Finally, various analysis techniques can be used to estimate some of the important probabilities related to dependability of computing platforms. This chapter discusses these topics and how they relate to digital avionics systems.

1. Dependability

Dependability is a comprehensive term that means that a particular system is fit for the use for which it is intended. A dependable avionics system, therefore, is one that can be trusted to support safe aircraft operation. This notion of dependability is intentionally a high-level concept and, without further elaboration, is essentially untestable.

To provide a basis for determining the necessary specific properties that a system has to have, six different attributes of dependability have been defined[3]. The attributes and their definitions are:

- ***Reliability***: Probability that a system will operate correctly in a specified operating environment for a specified time period.
- ***Availability***: Probability that a system will be operating correctly at a specified time.
- ***Safety***: Absence of catastrophic consequences on the users or the environment.
- ***Confidentiality***: The absence of unauthorized disclosure of information.
- ***Integrity***: Absence of improper state alterations.
- ***Maintainability***: Ability to undergo repairs and modifications.

In practice, the general dependability requirement for an avionics system or any of its components has to be refined and elaborated, finally being stated in terms of these six attributes. Thus, for example, an autopilot that is sufficiently dependable for use in a passenger aircraft might be required to: (1) have a reliability of its main functions of 0.99999999 for ten hours of operation; and (2) have been shown to fail in a

manner that is considered safe should it be unable to continue operating. Such a system would almost certainly have no requirement for data confidentiality or data integrity, although confidentiality and integrity are becoming increasingly important with increasing interconnectedness of dependable systems.

2. Faults

2.1. Types of Faults

The reason that avionics systems are not perfect is because of faults. A *fault* can be thought of informally as a system component with a problem[1]. Avionics systems are subject to a myriad of different types of fault. The component might have worked correctly until it broke; in this case it is referred to as a *degradation* fault. Some faults on the other hand—in particular, all software faults—are present as a result of mistakes having been made in the system's design. They are referred to as *design* faults.

Degradation faults occur when physical hardware components deteriorate over time and fail. Since all hardware components have a limited lifetime, degradation faults will inevitably occur at some stage. In addition, degradation faults in avionics systems are caused by various elements of the system's operating environment. These include vibration, humidity, temperature variations, electrical power transients, mechanical damage, and electromagnetic interference. These factors may lead to abrupt failures or gradual deterioration of physical components.

Design faults are introduced into both hardware and software systems during development and maintenance. Design faults are unpredictable and usually result in sudden and unexpected errors. Examples of design faults in hardware include wiring mistakes, improper sizing of power components, and logic mistakes in digital circuits. Examples of design faults in software include erroneous computation of numeric quantities, incorrect logic in program control flow, memory leaks caused by failure to manage dynamic memory correctly, excessive loss of precision in numeric computation, and missed deadlines in real-time computation.

The effect of a fault need not be manifested *permanently*. Sometimes a fault is *transient*, i.e., it manifests itself briefly and might not manifest itself again, or it is *intermittent*, i.e., it manifests itself temporarily but repetitively and irregularly.

Although degradation faults are important and must be addressed comprehensively in any avionics system design, modern techniques are quite effective in dealing with them. The complexity of modern systems, especially the software, has led to a situation where design faults are the dominant threat. Our ability to deal with design faults is far less well established.

2.2. Dealing With Faults

Since faults are the reason for a system's lack of dependability, the faults that might arise in a system have to be determined and provision made to deal with them. Identification of the specific faults that might arise in a given system has to be carried out very carefully, and every effort must be made to try to ensure that no possible fault has been missed. Many techniques are used to identify faults, but the most common one is *fault tree analysis*. In this technique, each hazardous state (a state that might lead to an accident if it arises) is examined, and each event that might lead to that state is identified. These events are then examined to determine what events might lead to them, and this process of refinement is continued until so-called basic events are identified. A basic event usually corresponds to the failure of a single component, i.e., a fault. Thus, for example, an aircraft's attempting to land with its landing gear up is a serious hazard. This hazard might arise because of pilot error or equipment failure. An equipment failure might be a mechanical failure, an electrical failure, a hydraulic failure, or a digital command failure. An electrical failure might be a

1. The detailed terminology of this topic is quite complex and will not be discussed here. The interested reader is referred to the work of Avižienis et al.[3].

loss of power, reduced power, excess voltage, and so on. Eventually this process leads to a list of possible faults that might arise. Although the process is informal, it works relatively well for most current systems.

For each fault identified by fault-tree analysis, the system designers must ensure that the system's design reduces the probability that the fault arises to an acceptable level. Four techniques are used to deal with faults and thereby to attain the necessary dependability in computing platforms. These techniques are:

- ***Fault prevention***: Prevention of the occurrence or introduction of faults.
- ***Fault removal***: Reduction in the number or severity of faults that exist in the system.
- ***Fault tolerance***: The use of system structures that allow the effects of faults to be tolerated should they manifest themselves during operation.
- ***Fault forecasting***: Estimation of the number of faults remaining in a system prior to deployment, their future rate of occurrence, and the likely consequences of those faults.

Fault prevention is attained by quality control techniques used during the design, manufacturing, and operation of hardware and software. For software, these techniques include the entire range of rigorous development methods. For hardware, rigorous design rules and design-rule checking are used. Additionally, shielding is used during operation to prevent or delay physical faults from sources such as heat and radiation, rigorous practices are used for maintenance, and firewalls are used to help prevent malicious faults.

Fault removal is performed during system development and operation. During development, whether a system implements the required functionality correctly is determined in a process known as *verification*. If verification fails, the problem is diagnosed and corrected. *Validation*, also carried out during development is the process of checking whether a system provides the functionality that is actually needed. It is quite possible to have a correct implementation of the wrong functionality. During system operation, corrective maintenance and preventative maintenance seek to remove faults that have produced errors and remove faults before they can cause errors[3].

Fault tolerance is the ability of a system to provide service even in the presence of faults. It is a major tool in dealing with degradation faults because it has been shown to be highly effective in that case. Since one cannot be sure that a processor, for example, will not fail during operation, the prospect of a degradation fault in which a processor is lost has to be faced. By replicating processors in various ways, loss of a processor can often be circumvented by allowing other processors in the set of replicates to provide continued service.

Finally, fault forecasting aims to identify failure modes that might lead to system failures and to evaluate the degree to which dependability attributes are satisfied. This process is mainly statistical and is heavily dependent on the actual system design, the anticipated operating conditions, and the availability of suitable statistical models. It is at this point that software becomes particularly problematic because statistical models of software failure are difficult to apply and notoriously inaccurate.

2.3. Electromagnetic Interference

A major threat to modern avionics that is becoming increasingly important and that can result in common–mode failure across a set of replicated components is *Electromagnetic Interference* (*EMI*). EMI is a disturbance caused by electromagnetic waves that "interrupts, obstructs, or otherwise degrades or limits the effective performance of electronics/electrical equipment"[18]. It has been a factor considered during the design and manufacture of aircraft since 1930.

The importance of EMI cannot be overstated for both modern analog and digital circuits. The energy needed for a disturbance is very small: "a typical discrete transistor can be hampered with 10^{-5} J for 50μs" and "a typical integrated circuit can be upset with 10^{-9} J". To give an idea of how small 10^{-5} J (Joules) is, it is a tenth of a millionth of the energy required to run a 100 watt light bulb for a second. EMI can be caused

by phenomena like lightning and sunspots, as well as the operation of systems or devices like radar, missile targeting systems, jamming devices, radios and personal electronic equipment.

The Federal Aviation Administration (FAA) and the European Joint Aviation Authorities (JAA) have identified three primary factors that contribute to the increased importance of EMI. First, electric and electronic systems are being used increasingly to perform functions that are responsible for the safe flight and landing of aircraft. Second, the power level required to operate most devices is decreasing as are feature sizes. Lastly, composite materials are being used increasingly in aircraft structures. The poor conductivity of such materials compared to metals results in greater voltage dissipation across the structure, development of significant magnetic fields, and the propagation of external electromagnetic energy. This leads to less shielding of the aircraft electronics from EMI by the structure.

3. Major Computing Elements and Architectures

The fundamental components of avionics system architectures are computers (including processors, memories and their interconnections), data buses, and software. These components may be configured in various computing system architectures, where the purpose of an architecture is to meet the functional demands of the computing platform and the dependability requirements. The most common architectures are *centralized*, *federated*, *integrated*, and *distributed*. Avionics architectures are often composed of modular components called *line replaceable units* (*LRUs*). The term refers to the basic physical building blocks that are used, and the important word is *replaceable*. Since an LRU can be replaced easily in the field, it is the basic unit of replacement when something breaks.

Federated Architectures

Traditional avionics systems have used *federated architectures* in which each function is implemented in one or more LRUs[22]. Only minor dependencies exist between functions, so there is a reasonably robust guard against error propagation. Functions do not share resources, and the failure of one function has little effect on others[36]. While this architecture is robust, it comes with a number of problems. One issue is that the lifetime of an aircraft is long, typically 25-30 years, while the underlying hardware rapidly becomes obsolete. Frequently, the computing hardware is out of date before the aircraft is even in service, and it is, consequently, expensive to replace. A tight hardware/software coupling often exists making it necessary to upgrade software when hardware is replaced. Additionally, the many different systems use many different types of hardware[13]. Federated architectures are also expensive, since each function has its own replicated system. Consequently, federated architectures are being replaced with integrated architectures.

Centralized Architectures

In a *centralized architecture*, computations take place in one or more computers in an area of the aircraft referred to as the avionics bay. Signals are usually transmitted over one-way data buses from sensors or other input devices to computers and to actuators or other output devices from computers. This architecture has the advantage that all computers are situated in an accessible location. Also, software development is simplified since there are only a few processor types and a few large programs that can be physically integrated. On the other hand, many long, expensive buses are required to collect and distribute data. Furthermore, the system is vulnerable to damage from a single hazardous event in or close to the avionics bay[40].

Integrated Modular Avionics Architectures

Integrated modular avionics (*IMA*) systems differ from federated systems in that some resources are shared across functions, i.e., several functions are implemented in a single computing unit. This integration allows fewer hardware units to be used and so offers many benefits such as lower weight, lower power

consumption, increased reliability, less frequent maintenance, and greater flexibility[22]. Many current IMA systems also manage the hardware problems found in federated architectures by making shared resources available through the use of a defined Application Programming Interface (API) layer. Using the API ensures that applications are portable to new hardware. Unfortunately, current certification standards and practice are still based on the federated approach and, as a result, do not support independent analysis of applications on a shared resource. Thus, IMA systems are more challenging to certify than federated systems [13].

Distributed Architectures

Finally, distributed architectures are being examined for future systems. A distributed architecture has several processors throughout the aircraft that are assigned different computing tasks and are connected together by one or more data buses. Sensors or actuators are sometimes connected to processors as peripherals and other times connected to the data bus (or buses) via an interface unit. Advantages of the distributed approach include: greater system flexibility because processing resources can be added quite easily; fewer, shorter buses connecting the computing system to sensors and actuators; and natural partitioning of subsystems. The disadvantages include a greater difficulty in software development because of the need to implement distributed algorithms for some functions, and possibly a greater variety of processor types that might complicate software generation, software validation, and the stockingof spares [40].

4. Dependability in Avionics Hardware

Avionics systems must operate amid vibrations, forces generated by acceleration, humidity, temperature extremes, temperature cycling, electrical power transients, and electromagnetic interference. As a result, the physical components of these systems inevitably deteriorate with age and fail at a rate that will lead to loss of important functionality unless something is done to maintain that functionality. In addition, avionics system hardware must be able to provide continued service in the presence of design faults. These issues necessitate the use of fault tolerance. In this section, we discuss a few techniques for creating hardware and bus architectures that are fault-tolerant.

Fault tolerance is always based on some form of redundancy. Redundant hardware can be used to maintain system operation in the presence of degradation faults. System failure occurs only after the exhaustion of spares or upon the occurrence of an unhandled fault.

The three basic forms of redundancy are *static*, *dynamic*, and *hybrid* (illustrated in Figure1). Static redundancy is based on a generalization of the *triple-modular redundancy* (*TMR*) concept known as *N-modular redundancy* (*NMR*). In NMR, N separate modules each compute an output. The outputs are compared and a voter chooses the majority output to propagate [25]. In this case, error detection is performed by the voter, and the effects of the fault are completely isolated from the output. The fault is thus said to have been masked.

Another approach to handling faults is explicit reconfiguration, also known as dynamic redundancy. In this approach, one unit is in use and there are one or more standby units. The operational unit includes checking circuitry, and, if the unit should fail permanently, the system will be reconfigured to use a standby unit. This approach does not necessarily mask the effects of a fault since service might be interrupted during reconfiguration[42].

Hybrid fault tolerance uses a combination of these approaches. A system will have N active, redundant modules and an additional S spare modules. If a module output disagrees with the voter, then a switching circuit replaces the failed module with a spare. When all spares are used, then faulty input can be switched out by the voter[25].

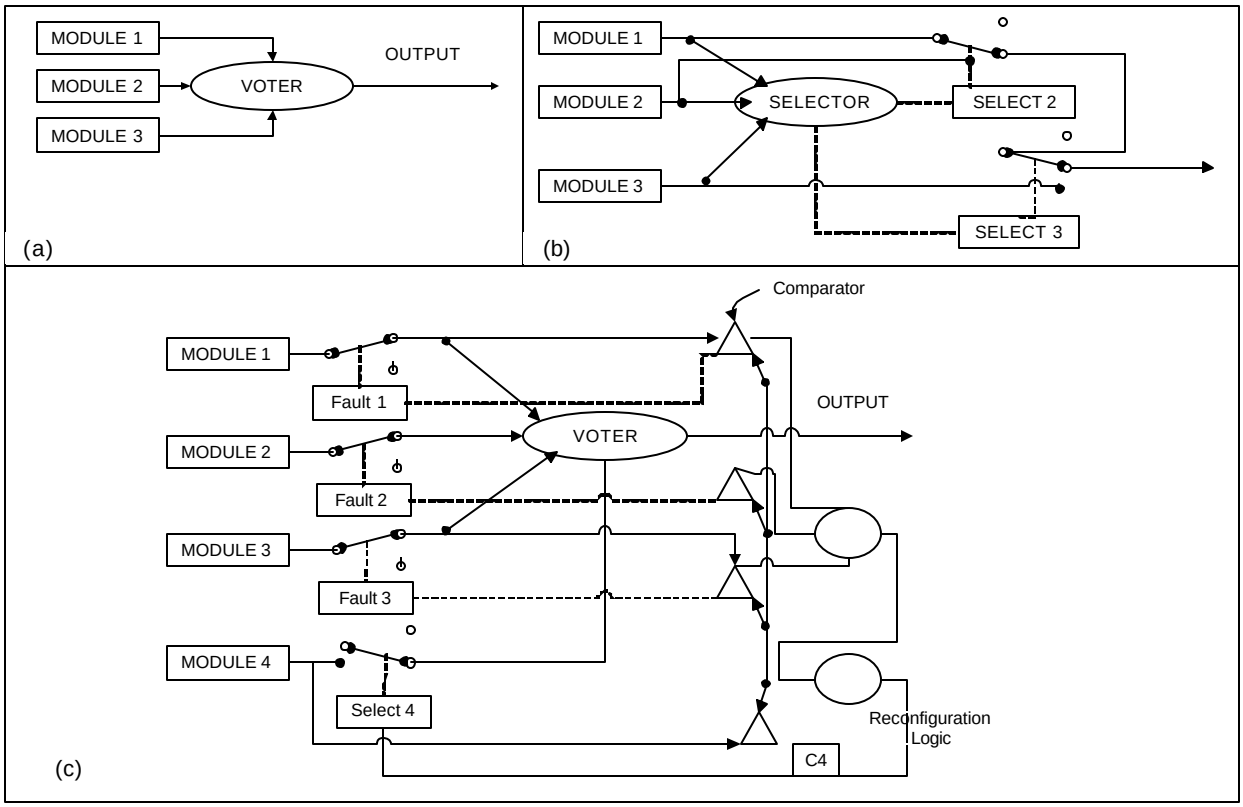

Figure 1. Basic Forms of Redundancy

(a) Fault Masking with TMR; (b) Reconfiguration; (c) Hybrid TMR

5. Bus Architectures

As discussed above, the systems used in avionics applications have traditionally been federated; each function had its own fault-tolerant embedded control system and only minimal connections were made between the systems. Because each function has its own replicated system, this approach is very expensive. Thus, recent applications are moving toward integrated and distributed solutions, sharing resources between functions (as described in section 3). In these systems, communication between components is an essential service, and so the communication bus has become a fundamental system component[37].

5.1. Terminology

In modern distributed avionics systems, application programs run on *host* computers. Sensors and actuators are connected to the hosts. An *interconnect* medium provides communications (often broadcast) between the hosts, which are connected to the interconnect via *interface* devices. The bus comprises the interfaces and the interconnect. The interconnect may be a physical bus or a centralized hub. A host and its interface are a *node* [36]. A *bus guardian* (BG) protects the bus from defective nodes by, for example, only allowing each node to transmit when it is scheduled to do so. A generic architecture is shown in Figure 2.

5.2. Time-triggered Versus Event-triggered Buses

Data buses are generally either *time-triggered* or *event-triggered.* Each bus type lends itself to different types of application, so the trigger type is a fundamental design choice in the design of a new bus. Some modern buses incorporate both trigger mechanisms, although, in doing so, they tend to compromise the qualities of one or the other.

Time-triggered buses

Time-triggered buses interact with components attached to the bus according to an internal time schedule. For instance, a host may be assigned to read a sensor and broadcast its value 20 milliseconds after the start of each time partition (frame). This global schedule is determined at design time. Each node knows the schedule and the time and therefore knows when to send messages and when messages should be received. Contention for the bus is resolved at design time.

With a time-triggered bus, there is no need to attach source and destination addresses to a message, since each node knows the sender and receiver based on the time the message is sent. Thus, the size of each message is reduced relative to the message size in an event-triggered bus.

An essential requirement for time-triggered buses is accurate clock synchronization. A global clock must be achieved where each node has a local clock that is synchronized with the clocks of all other nodes. The quality of the synchronization is based on the quality of the clock oscillators at each node and the algorithm used in synchronization. Averaging and event-based algorithms are two classes of algorithms used for synchronization. With averaging algorithms, each node measures the skew between its clock and the clocks of other nodes and then sets its clock to a "fault-tolerant average." The skew is measured by knowing when other nodes are supposed to transmit on the overall system schedule. Any difference between the expected time and the actual time is a difference between the clock of the sender and the clock of the receiver [36].

Event-triggered buses

Event-triggered buses are driven by events as they occur. A host, for example, might transmit the value of a sensor whenever it changes. Naturally, this leads to unpredictable bus capacity demands, and either the possibility of collisions on the bus or the need for a bus arbiter.

Because event-triggered systems are under the control of the environment and not a schedule, they can provide a flexible allocation of resources. This is attractive for systems with variable demands, but is a problem in safety-critical applications. In an event-triggered system, events will arrive at different nodes and cause them to contend for the bus. Safety-critical applications, however, must guarantee some basic quality of service to all participants. For this reason, time-triggered buses are generally preferred for safety-critical applications like avionics [36].

5.3. Data Bus Dependability

Data buses are subject to a wide range of faults. Faults can affect value, time, and space. Value faults cause an incorrect value to be computed, transmitted, or received. Timing faults cause a value to be computed,

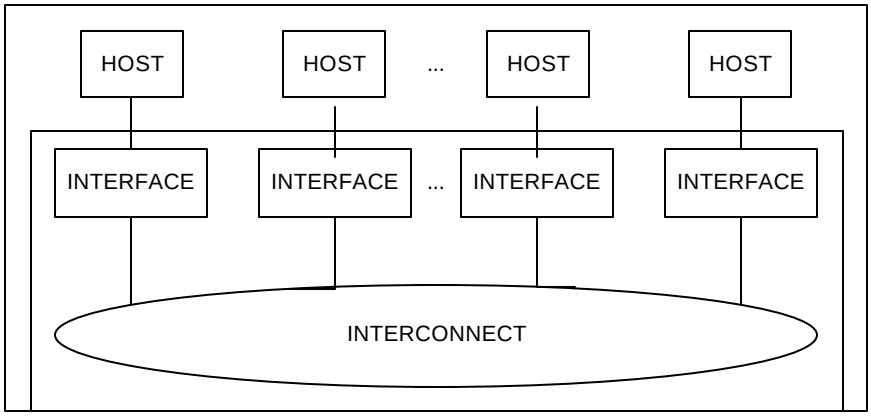

Figure 2. Generic Bus Configuration[37]

transmitted, or received at the wrong time. And spatial proximity faults occur when everything within a physical proximity is destroyed[36].

The central role that data buses play in modern distributed, embedded systems has led to major advances in bus dependability in recent years. Many analytic results have been established that allow important dependability characteristics of data buses to be predicted. The state of the art is such that very wide ranges of faults are known, and bus designs have been developed to deal with them effectively.

The importance of data buses in many application domains has also led to the availability of commercial implementations of the hardware and software needed for some of the ultra-dependable bus designs. An example is the wide range of products that are available from TTTech Computertechnick AG of Vienna, Austria[44].

6. Software Systems

Software is of vital importance in avionics, and it is the major source of complexity in avionics systems. There are three major categories of avionics software with which one has to be concerned: operating systems, system software, and applications.

Traditionally, operating systems used in avionics have been much less capable than those found in desktop machines. The difference lies mostly in the variety of services provided; avionics operating systems provide relatively few because many desktop services are not required. This relative simplicity even led to the term operating system being dropped in favor of *executive* or *monitor* in some cases.

The operating systems used in avionics systems are usually custom software created by a company to support its avionics products, and they are, therefore, typically proprietary. This situation is changing, however as the industry moves to the use of commercial-off-the-shelf (*COTS*) products. There is great interest in using variants of desktop operating systems, like Linux and Microsoft Windows, in avionics systems. The issues that this raises in terms of software quality and in product certification by the FAA are yet to be resolved.

An interesting development has been the creation of extremely high-quality operating systems that are intended for application domains like avionics by companies that do not develop specific avionics components. These systems are COTS products but they differ from typical COTS operating systems in that they are not primarily designed to support desktop machines. In addition, these COTS products come with the ancillary information needed by an avionics supplier in preparing a product for FAA certification. Thus, the operating system can be used in avionics applications but the effort required to get product certification is both minimal and predictable. Two examples of this new type of COTS operating system that are used extensively in the avionics industry are VxWorks produced by Wind River Inc.[47] and Integrity produced by Greenhills Software Inc.[23].

System software is the software that supports applications but which is not normally viewed or constructed as part of the operating system. This category includes middleware, device drivers, graphics support, and tools such as compilers and linkers. The situation with system software is very similar to the situation with operating systems. Traditional proprietary approaches have been challenged by the goal of using COTS products, and software manufacturers have started to produce COTS products tailored to the needs of safety-critical systems such as avionics. In some cases, the products are supplied with all the necessary support materials to permit the user to obtain FAA certification of the systems containing these products. An example of this type of COTS software is the graphics and related software produced by Quantum 3D[34].

Application software is not in the fortunate state of having independent suppliers of COTS products with associated certification materials. By definition, application software is unique to the product in which it operates, and it is built to provide a unique or at least different avionics product. Application software is, therefore, written mainly by avionics system suppliers, is proprietary, and likely to remain so. In the remainder of this section, we summarize various topics related to the development of application software.

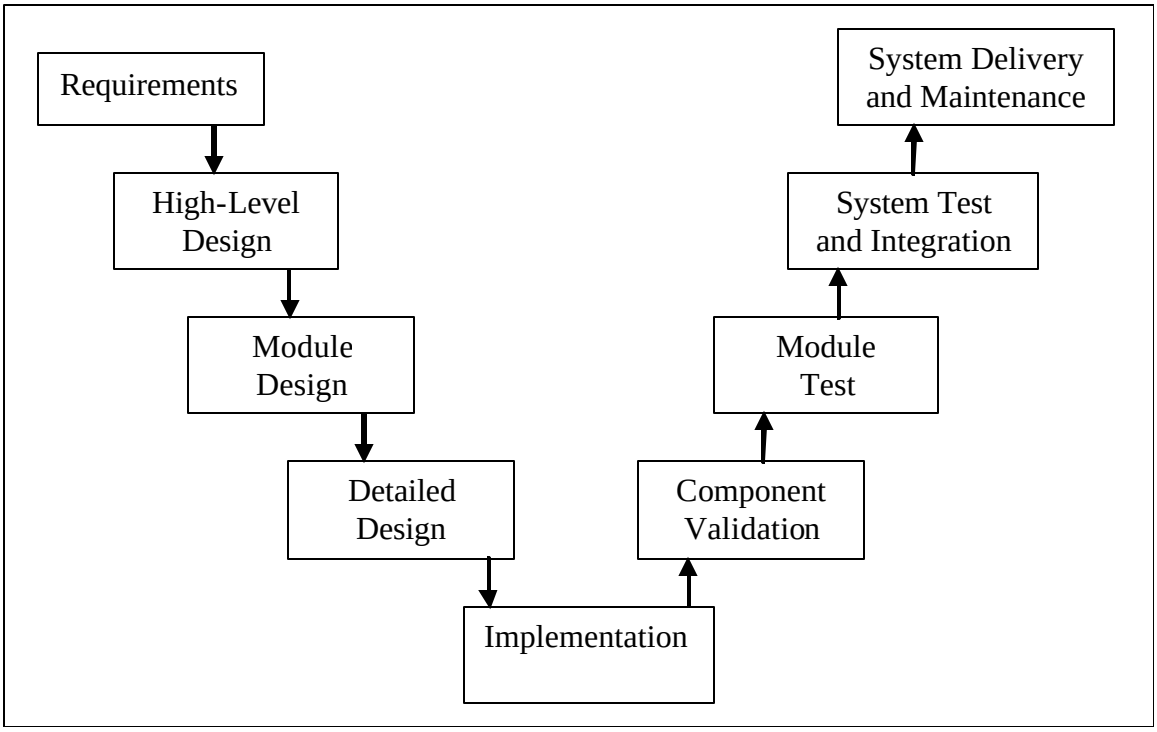

Figure 3. A Typical Software Lifecycle

6.1. The Software Lifecycle

The overall approach to software development is referred to as the *software lifecycle* (see Figure 3). There are many different implementation of the lifecycle in entities referred to as software development *processes*. Different processes have different goals, for example minimize time to market or minimize total development costs. Naturally, even though cost and timeliness are issues in avionics software development, the key goal is to make sure that the software properly supports the requirement for safe flight operation.

Software development starts with the analysis of requirements by the customer and might include various prototyping or feasibility studies designed to try to understand the problem. The goal is to identify all of the customer requirements and all of the details for each requirement so that no functionality or pragmatic detail is missed. The requirements are then transformed into a specification that documents the functionality, dependability, target hardware details, and any other restrictions that might be required. Once the specification is complete, the overall system is designed and then each sub–system is designed. Several issues like the allocation of memory resources, interface design and constraints due to timing requirements can be addressed at this point[40]. The partitioning of the system into appropriate sub-systems goes hand-in-hand with interface design. Together, these activities can ease the process of making code changes at later stages and facilitate system certification[40]. Similarly, modularity and encapsulation of data are two important aspects of component design in software. During each design stage in the software lifecycle, the customer should be involved to make sure that the product being built matches the customer's requirements and expectations.

Implementation is the creation of the program code that constitutes the software, usually in a high–level language. Testing is employed to convince the development team and customers that the system works correctly. Testing is done both at the component and module levels as well as at the system level once the system has been integrated. Certification requires that particular test coverage metrics be used, the

specific metric depending on the criticality of the software component. For the most crucial avionics systems, an important metric known as *Modified Condition Decision Coverage* (*MCDC*) is employed to determine test adequacy. Informally, MCDC requires that every decision in the program (if statement, while statement, etc.) has been executed to ensure that every condition (predicate or Boolean variable) within each decision has been tested with values of True and False and has been shown to affect the flow of control correctly. This metric is quite complex and ensuring complete MCDC coverage is typically the responsibility of experts who have a lot of experience with this particular coverage metric.

Various techniques have been developed to supplement the testing process, but they are rarely used by the avionics software community primarily because the product certification process does not give what is perceived to be adequate credit for the use of the techniques. An example is *static code analysis*, the automatic analysis of code by a tool. Static analysis is carried out before the code is executed, and it can help to eliminate several possible errors in the code such as code that can never be reached ("dead" code), infinite loops, and uninitialized variables. An example static analysis tool is Splint[19]. If testing is used to verify a system, it must be done exhaustively to make the verification truly convincing; however, this has been shown to be an infeasible approach for safety–critical systems such as those on aircraft[7]. Thus static analysis and other automated, mathematically-based checking mechanisms must be adopted in order to ensure that the software has certain crucial properties. This change will require a substantial revision to the FAA's current policy on software.

6.2. Aviation Standards and Regulations Governing Software

Achieving software quality for avionics systems is important and is the responsibility of the avionics manufacturer. Monitoring that quality is the responsibility of a regulating agency. In the United States, that agency is the Federal Aviation Administration; in Europe it is the Joint Aviation Authorities; and there are other agencies in other parts of the world.

The world's aviation regulating agencies have chosen to undertake their responsibility in this area by requiring development of software to follow certain guidelines. The degree of compliance with the guidelines is checked by the regulating agency and software is deemed of adequate quality if the guidelines have been followed appropriately. The aviation guidelines are referred to in the United States as RTCA Inc's DO–178B and in Europe as the European Organization for Civil Aviation Electronics EUROCAE ED–12B [20].

This guideline-based approach is successful in practice, but it relies on an assumption that is not necessarily valid. The assumption is that following a good process will produce a good product. There is no reason to believe that this will always be true; especially for future systems that will be much larger and more complex than current systems. Despite this, it is clear that certain classes of faults are precluded by following the guidelines.

The failures of different avionics services will have different consequences, and so a set of software levels have been defined that specify the required software process guidelines of specific services based on their consequences of failure. The software levels are identified by the letters A through E, with A being the most critical, and summaries of the definitions of the various levels are shown in Table 1[1].

It is important to note that these assurance levels apply to specific services without regard to how they are implemented or how they might fail; assurance is of the service, not a specific aspect of its implementation. Thus, it is not meaningful to refer to "level A software" only to "software for a level A system". Building a level A system requires that the highest levels of attention be paid to the electronic hardware, the software (including any data files that it uses), and the physical packaging that is used.

1. The detailed wording of the software levels is contained in RTCA DO-178B, section 2.2.2.

Table 1: Assurance Levels

Software Level	Effect of Anomalous Behavior	Maximum Permissible Probability of Failure Per Hour
A	Software whose failure could cause a catastrophic failure condition.	10^{-9}
B	Software whose failure could cause a hazardous failure condition.	10^{-7}
C	Software whose failure could cause a major failure condition.	10^{-5}
D	Software whose failure could cause a minor failure condition.	10^{-3}
E	Software whose failure would have no effect.	NA

6.3. Tools and Techniques for Avionics Software Construction

A wide variety of innovative tools and techniques have been developed to help deal with the development of avionics and other safety-critical software. The tools and techniques tackle the problem in many different ways, and in this section we examine three that are very different from each other. We begin with programming languages. Next we discuss a combination of a process and a support toolset that is, in fact, an innovative process called *Correctness by Construction*. Finally we present a summary of a technique known as *Model-Based Development* and one particular implementation of the technique.

Programming Languages

Implementation of avionics software is generally in a high–level language. Some of the programming languages that have been used are C, C++ and Ada. Although some aspects of software development costs can sometimes be reduced by using popular languages like C, C++ and Java, these languages are not ideal for safety–critical systems. Software built with them cannot achieve the high reliability and low failure rates required for avionics without great difficulty because the languages do not provide the developer with much help in getting the software right. C, for example, is essentially a typeless language and so programs with even elementary semantic errors will often compile and execute. Sometimes simple abuses of the types of variables are not caught without extensive testing. Given the effort that has to be expended on testing programs written in C and the number of ways in which subtle mistakes can be made, it is hard to justify the decision to use C in avionics applications.

C++ and Java have better type systems than C but they suffer from their own difficulties. Both languages have very complex semantics, and it is difficult for the average programmer to be sure they understand what a given program will do. C++ includes pointers and their use makes verification quite difficult. Java omits explicit pointers but, in doing so, incurs performance penalties that are sometimes problematic in safety-critical systems where predictable timing is important. Neither C++ nor Java were designed with the special needs of the avionics community in mind and so important features such as those that support real-time computation, predictable management of dynamic storage, access to machine facilities, and concurrency are either absent or not included in the language in an efficient way. Real-time Java is being developed with extensions to Java that support real time, but the language has a variety of problems including size, complexity, and lack of a formal definition.

Ada is based on an entirely different set of design principles. From the outset, Ada was designed for embedded systems and safety-critical applications, and so it has many features that are desirable in avionics system development. It has been considered a programming language of choice for digital avionics systems for many years[11, 40], and it has a proven record in both military and civil aviation applications. While other languages are often used in current avionics projects, their selection is due primarily to convenience rather than appropriateness of the language for the application.

Researchers have addressed the issue of programming language design extensively, and significant advances have been made in language concepts for safety-critical applications. Some of the language features that have presented significant challenges include dynamic memory management, concurrency, inheritance, dynamic dispatch, the object–oriented and aspect–oriented programming paradigms, and dependence of a language on the underlying architecture and operating system. Publications presenting results in these areas are too many in number to be discussed here. The reader is referred to the archival literature.

Correctness by Construction

The goal of Correctness by Construction is to develop software that is essentially correct from the moment it is written. Ada is one of the few languages that provide support for software correctness by virtue of its design. This is very valuable, yet more could be done at the language level if one were prepared to restrict the use of language features in such a way that analysis, by both humans and machines, was facilitated. This is the philosophy behind SPARK, a variant of Ada.

SPARK is the *SPADE Ada Kernel*. It was developed in England at DERA Malvern, Southampton University, and Praxis Critical Systems [11]. It is a subset of Ada ’95 designed to maximize the potential for analysis, and, as such, it is very well suited to the implementation of high-integrity systems. SPARK addresses several concerns in analyzability and assurance of software implementations. More specifically: (1) a complete formal specification of the language has been developed so that it is clear exactly what any program means; (2) features that would impact the feasibility of comprehensive verification have been excluded from the language; (3) a sophisticated annotation language permits developers to include what amounts to a program specification in with the program itself; and (4) a powerful set of tools has been developed to support the establishment of important properties of software written in the language.

An example of the impact of these features on software development can be seen in its treatment of exceptions. Full Ada has an exception mechanism with very complex semantics. SPARK does not include exceptions, and so it is necessary for a developer to show that his or her software will not raise any. This is possible with SPARK because the necessary state conditions for the proof can be written using the annotation language and the proof itself can be either completed or checked by the SPARK tools.

SPARK has been used successfully in many safety-critical applications such as the Mission Critical computer on the Lockheed C130J aircraft, which has both military and commercial transport versions. SPARK helped in meeting the required DO-178B standards for the commercial Lockheed C130J. Lockheed reported an 80% savings in the budget that was allocated to Modified Condition/Decision Coverage (MC/DC) testing[12] for this project.

Model-Based Development

Conventional high-level languages have changed little since the first ones appeared in the 1950s. Java and Fortran, for example, are both procedural languages that expect the programmer to write software at a level that is very close to the semantics of the underlying hardware. Various ideas have been tried to replace this paradigm, and one that is gaining considerable success and popularity in the aerospace community is *model-based development*.

Model-based development is a radical departure from traditional approaches to software development. One approach to its use (there are a number of variants) is to allow application experts to define a specification for the software they want using a formal specification notation that is frequently graphical. This

specification is validated by application experts using inspection, analysis, and also by simulating it. Once the specification is complete and validated, the actual software that will implement the specification is synthesized by a tool that is part of the overall support toolset.

Model-based development embraces the concept of correctness by construction since, assuming that the tool that generates the code is correct, then the code itself is correct. The assumption about the code generator might seem like a large and unreasonable one, but it must be kept in mind that the tool is used on every product and so substantial effort in its development is justified.

A good example of a system for model-based development is the Safety Critical Application Development Environment (SCADE) [10]. SCADE comes in a variety of versions, each of which is designed to support a different application domain. The avionics version, SCADE Suite, is the de facto standard for software development and has been used widely in production avionics systems, including much of the software on the Airbus A380[8].

The major tools that SCADE provides include: (1) a graphical editor for developing specifications; (2) a verifier that allows substantial formal proofs of important properties of a specification to be created; (3) code generators for creating the program in a variety of high-level languages; (4) various report generators that permit synthesis of required documents; and (5) links to other tools (from other manufacturers) that provide various forms of traceability and different forms of specification.

Among its many impressive features, SCADE includes several that provide comprehensive support for software development that is compliant with *RTCA DO-178B/EUROCAE ED-12B*. The SCADE Suite code generator, KCG, produces C code that is designed to integrate into a user's DO-178B qualification plan. Thus, it supports the required tool qualification, documentation, traceability, test coverage, configuration management, source-code standards and restrictions, and so on that are part of the DO-178B process. Although certification is not immediate, it is streamlined, substantially automated, and much less expensive than when developing software using a traditional process.

7. System Failures

Unfortunately, incidents and accidents involving commercial aircraft occur occasionally, and in this section we present brief summaries of some examples in which digital systems were causal factors. In almost all cases, incident and accidents are the result of complex combinations of events and circumstances. It is important to keep in mind that the details of the incidents and accidents discussed here are extensive, and that none was caused by some relatively simple defect in a digital system. An excellent compendium of aviation incident and accident reports has been developed by Peter Ladkin of the University of Bielefeld, Germany[28, 29].

American Airlines 965

On December 20, 1995 a Boeing 757 aircraft operated by American Airlines crashed in Cali, Columbia. The aircraft was fully operational and the crew was in control at the time of the accident, and so this accident was of the type referred to as *controlled flight into terrain* (CFIT). Of the 163 people on board the flight, only four survived.

The accident occurred on approach to Cali and followed extensive interaction between air traffic controllers and the flight crew. Concerns have been raised about the possibility of miscommunication during this interaction because the communication was primarily in English, but English was not the first language of all involved.

At the outset of the approach, the runway to be used was changed from the runway documented in the flight plan. This meant that the crew had to fly different headings and follow a different descent profile from that planned. During the approach, the crew was required to enter the identifiers of various navigational aids, and a significant causal factor cited in the accident is confusion by the crew over the symbols used to describe the navigation aids. The crew thought they were heading for one particular aid when, in

fact, the aircraft was heading for a different aid in an entirely different location. The terrain was extremely mountainous and the aircraft struck the ground at a high altitude.

Singapore Airlines SQ286

On March 12, 2003 the tail of a Boeing 747-412 operated by Singapore Airlines struck the runway during takeoff rotation at Auckland International Airport in New Zealand causing substantial damage to the aircraft but no injuries. The accident was caused by erroneously low takeoff reference speeds being entered into the aircraft's FMS. The gross takeoff weight was 347.4 tonnes but the value entered into the FMS was 247.4 tonnes. The error occurred when the first officer wrote down the incorrect value after the captain read out the correct value from the dispatch paperwork. The rotation speed calculated from the incorrect weight was 130 knots whereas the correct value was 163 knots. A similar incident occurred on landing with a Delta aircraft in 1998.

The circumstances surrounding these accidents were quite complex, and many different checks or procedural changes could have avoided them. The National Transportation Safety Board published a safety recommendation in response to these accidents in which it recommended software changes in the FMS to either disallow or challenge certain entries that could be identified as erroneous[33].

Malaysian Airlines 124

On September 9, 2005 the FAA issued an airworthiness directive for Boeing 777s[16]. The directive requires a change in the avionics software and various changes to the flight manual for the aircraft. The problem that the airworthiness directive addresses was incorrect processing by the avionics software of data from failed sensors that could lead to anomalies in the fly-by-wire primary flight control system, the autopilot, the auto-throttle, the pilot display, and the auto-brake system.

This airworthiness directive was found to be necessary by the FAA following an incident in which a Boeing 777 operated by Malaysian Airlines effected a significant nose-up pitch while climbing through 36,000 feet on departure from Perth, Australia. Subsequent actions by the crew stabilized the aircraft despite various errors in the displays and the aircraft's approaching a stall. The faulty sensors were in the air data inertial reference unit; the avionics software failed to recognize that the data was not meaningful.

Resources

[1] Aleska, B. and Carter, J. "Boeing 777 Airplane Information Management System Operational Experience." Proceedings of the 16th AIAA/IEEE Digital Avionics Systems Conference, Irvine, CA, October 1997.

[2] Anderson, T. and P. A. Lee. Fault Tolerance, Principles and Practice. London: Prentice-Hall, 1981.

[3] Avižienis, A., Laprie, J., Randell, B., and Landwehr, C. "Basic Concepts and Taxonomy of Dependable and Secure Computing.". IEEE Transactions on Dependable and Secure Computing, Vol 1, No. 1, pp. 11-33 (January 2004)

[4] Baufreton, P., et al. "SAFEAIR: Advanced Design Tools for Aircraft Systems and Airborne Software." Proceedings of the 2001 International Conference on Dependable Systems and Networks. IEEE Computer Society. Göteborg, Sweden July 2001.

[5] Berry, G. "The Effectiveness of Synchronous Languages for the Development of Safety–Critical Systems", Esterel Technologies, France, 2003.

[6] Brower, R. "Lockheed F-22 Raptor." The Avionics Handbook. New York: CRC Press, 2001.

[7] Butler, R. W., and Finelli, G. B. "The Infeasibility of Experimental Quantification of Life-Critical Software Reliability." ACM SIGSOFT '91 Conference on Software for Critical Systems, New Orleans, LA, December 1991.

[8] Camus, J. and Dion, B. "Efficient Development of Airborne Software with SCADE SuiteTM."

[9] Carter, N. “The Background to the HIRF Requirements for Civil Aerospace” DERA Farnborough, <http://www.compliance-club.com/archive1/980212.html>
[10] Caspi, P., Curic A., Maignan, A., Sofronis, C., Tripakis, S., Niebert, P., Sergent, T. L., “From Simulink™ To SCADE™/Lustre To TTA: A Layered Approach For Distributed Embedded Applications” <http://www.esterel-technologies.com>
[11] Chapman, R. “SPARK - a state-of-the-practice approach to the Common Criteria implementation requirements.” Praxis Critical Systems Limited, Presented at the 2nd International Common Criteria Conference, Brighton, UK, July 2001.
[12] Chapman, R. “Industrial Experience with SPARK.” Praxis Critical Systems Limited, Presented at ACM SigAda 2000 conference.
[13] Conmy, P. and McDermid J. “High level failure analysis for Integrated Modular Avionics.” Proceedings of the Sixth Australian Workshop on Industrial Experience with Safety Critical Systems and Software, Brisbane, Australia, July 2001.
[14] Croxford, M. and Sutton, J. “Breaking Through the V and V Bottleneck”, Published by Springer-Verlag in “Lecture Notes in Computer Science” Volume 1031, 1996.
[15] deLong, C. “AS 15531/MIL-STD-1553B Digital Time Division Command/Response Multiplex Data Bus.” The Avionics Handbook. New York: CRC Press, 2001.
[16] Department of Transportation, Federal Aviation Administration, AD 2005-18-51, Federal Register, vol. 70, no. 175, pp. 53,547 - 53,550.
[17] Dorsey, C. “Certification” Presentation in Digital Avionics Systems, March 24th 2004, Department of Computer Science, University of Virginia.
[18] EMI, <http://www.its.bldrdoc.gov/fs-1037/dir-013/_1935.htm>.
[19] Evans, D., et al. “LCLint: A Tool for Using Specifications to Check Code.” SIGSOFT Symposium on the Foundations of Software Engineering, December 1994.
[20] Ferrell, T. K. and Ferrell U. D. “RTCA DO-178B/EUROCAE ED-12B”, Chapter 27, edited by Spitzer, C. R. The Avionics Handbook. New York: CRC Press, 2001.
[21] Glenair Homepage, Glenair. “Electromagnetic Interference In High Reliability Electrical Interconnect Systems.” 26 Feb. 2004 <http://www.glenair.com/html/emi.htm>.
[22] Goshen-Meskin, D., Gafni, V., and Winokur M. “SafeAir–An Integrated Development Environment and Methodologies” INCOSE 2001 Symposium, Melbourne, Australia 1-5 July, 2001.
[23] Greenhills Software Inc. http://www.ghs.com
[24] Hess, R. “Electromagnetic Environment (EME).” Chapter 25, edited by Spitzer, C. R. The Avionics Handbook. New York: CRC Press, 2001.
[25] Hitt, E. F. and D. Mulcare. “Fault-Tolerant Avionics.” The Avionics Handbook. New York: CRC Press, 2001.
[26] Hoyme, K. and Driscoll, K. “SAFEbus.” Proceedings of the IEEE/AIAA 11th Digital Avionics Systems Conference, Seattle, WA, October 1992.
[27] Knight, J.C. and Leveson, N.G. “An Experimental Evaluation of the Assumption of Independence in Multi-version Programming”, IEEE Transactions on Software Engineering, Vol. SE-12, No. 1, January,1986.
[28] Ladkin, P.B., “Abstracts of References and Incidents”, http://www.rvs.uni-bielefeld.de/publications/Incidents/DOCS/FBW.html
[29] Ladkin, P.B., “Computer-Related Incidents with Commercial Aircraft”, http://www.rvs.uni-bielefeld.de/publications/compendium/incidents_and_accidents/index.html
[30] Ladkin, P. B., “Electromagnetic Interference with Aircraft Systems: why worry?” 26 Feb 2004 <http://www.rvs.unibielefeld.de/publications/Incidents/DOCS/Research/Rvs/Article/EMI.html>.
[31] McDermid, J. A. “Software Safety: Where’s the Evidence?” 6th Australian Workshop on Industrial Experience with Safety Critical Systems and Software (SCS ’01), Brisbane.
[32] Morgan, M. “Boeing B-777.” The Avionics Handbook. New York: CRC Press, 2001.
[33] National Transportation Safety Board, Safety Recommendation A-05-03, March 8, 2005.

[34] Quantum 3D Inc. http://www.quantum3d.com
[35] RASSP Architecture Guide. <www.eda.org/rassp/documents/sanders/arch_guide_b.pdf>.
[36] Rushby, J. “Bus Architectures for Safety-Critical Embedded Systems.” Lecture Notes on Computer Science 2211 (2001): 306-323.
[37] Rushby, J., “A Comparison of Bus Architectures for Safety-Critical Embedded Systems.” SRI International Technical Report, September 2001.
[38] SAE International, “Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment”, 1996.
[39] Seeling, K. A. “Reconfiguration in an Integrated Avionics Design.” Proceedings of the 15th AIAA/ IEEE Digital Avionics Systems Conference, Atlanta, GA, October 1996.
[40] [Spitzer 87] Spitzer, C. Digital Avionics Systems. Englewood Cliffs, NJ: Prentice Hall, 1987.
[41] [Spitzer] Spitzer, C. R. “F-22.” Presentation slides obtained from Spitzer. Date unknown.
[42] Storey, N. Safety-Critical Computer Systems. New York: Addison Wesley, 1996.
[43] Systems Integration Requirements Task Group, SAE. “Certification considerations for highly-integrated or complex aircraft functions (ARP 4754)” 1996.
[44] TTTech Computertechnick AG. http://www.tttech.com
[45] United States General Accounting Office. “Status of the F/A-22 Program.” April 2, 2003.
[46] Weaver, R.A. “The Safety of Software: Constructing and Assuring Arguments.” Ph.D. dissertation. University of York. 2003.
[47] WindRiver Inc. http://www.windriver.com
[48] Witwer, B. “Developing the 777 Airplane Information Management System (AIMS): A View from Program Start to One Year of Service.” IEEE Transactions on Aerospace and Electronic Systems. April 1997.
[49] Winter, D. C. “Avionics Software Challenges and Initiatives” Briefing to Software Design and Productivity Workshop, 18-19 April, 2001
[50] Yeh, Y. C. “Triple-Triple Redundant 777 Primary Flight Computer. Proceedings of the 1996 Aerospace Applications Conference. IEEE Computer Society Press: Los Alamitos, CA, 1996, pp. 293-307.
[51] Zhang, J. and Pervez, A. “Avionics Data Buses: An Overview.” IEEE Aerospace and Electronic Systems Magazine February 2003.

Annotated Bibliography

Editors' note: This bibliography was compiled by all of the book's authors. Each author contributed the references he/she found the most useful in writing his/her chapter.

1. **Alonso, Fernando. "FBW Evolutions." Proc. 10th Performance and Operations Conference, San Francisco, CA, September 1998.**

   This paper discusses the evolution of Airbus's flight control system beginning with the A320 and progressing through the A330 and A340 toward the upcoming A380 jetliner. It describes enhancements Airbus made to its FCS since the A320 to enhance passenger comfort through an improved feedback loop and to improve the electrical rudder control.

   The paper provides a good overview of the Airbus flight control system. While detailed, it avoids going into the details of the control laws or complex system design. More importantly, it shows how Airbus has refined its high-level FCS design through successive models of aircraft.

2. **P. Douglas Arbuckle, Kathy H. Abbott, Terence S. Abbot and P. C. Schutte. "Future flight decks." Proceedings of the 21st Congress International Council of the Aeronautical Sciences, 1998. <http://techreports.larc.nasa.gov/ltrs/PDF/1998/mtg/NASA-98-21icas-pda.pdf>**

   The authors argue for the requirement of the human operator and states that the current airspace relies on humans as control elements in the airspace. They argue for an analysis of why pilots make errors and present a model for aviation accidents and well as guide the reader through the commercial transport flight deck evolution. Trends in the Global Environment, such as speech recognition and eyeglass based displays, are also discussed. Lastly, the authors discuss human centered flight deck design. They talk about human centered design principles, function allocation, flight deck mission categories, task oriented display design and fault management.

3. **Avizienis, A., J.-C. Laprie, B. Randell, and C. Landwehr. "Basic Concepts and Taxonomy of Dependable and Secure Computing."** ***IEEE Transactions on Dependable and Secure Computing*** **1(1), pp. 11-33, Jan.-Mar. 2004.**

   This paper summarizes the fundamental concepts of dependability. It is explained that dependability is a property that integrates the attributes of reliability, availability, safety, security, survivability, and maintainability. Definitions are given. Additionally, faults, errors, and failures are discussed, as well as the means for attaining dependability, such as fault prevention, fault tolerance, fault removal, and fault forecasting. The paper gives background material that is essential for anyone researching safety-critical systems.

4. **M. Barbaci, P. Clements, A. Lattanze, L. Northrop and W. Wood. "Using the Architecture Tradeoff Analysis Method (ATAM) to Evaluate the Software Architecture for a Product Line of Avionics Systems: A Case Study." Carnegie Mellon University - Software Engineering Institute, 2003.**

This Technical Note from Carnegie Mellon's Software Engineering Institute details the contextual background of a proposed software architecture (Common Avionics Architecture System) for all US Army special operations helicopters, including the MH 60, as well as a proprietary acquisition methodology (Architecture Tradeoff Analysis Method) to evaluate architectural attributes. In order to reduce the risk inherent in evaluating software, the authors proposed the use of CAAS in conjunction with ATAM. This has been proposed for all airframes the Army employs for special operations, in order to leverage the commonality among development stakeholders. Evaluation of the ATAM took place in October and December 2002.

5. **Baufreton, P., et al. "SAFEAIR: Advanced Design Tools for Aircraft Systems and Airborne Software." Proceedings of the 2001 International Conference on Dependable Systems and Networks. IEEE Computer Society. Göteborg, Sweden July 2001.**

This paper talks about the SAFEAIR project which aims at realizing high dependability in real time embedded systems including those used in the aerospace industry. It discusses the role of an Aircraft System Development Environment (ASDE) in building SAFEAIR. ASDE combines a Model Builder, a Model Verifier, SCADE and Statemate into a single toolset. This greatly simplifies various stages in the software life cycle and thus helps in reducing costs. The paper also describes the properties that make SCADEa useful language. It also describes CVT-C and CVT-A, which are code verification tools in the ASDE toolset.

6. **Berry, G. "The Effectiveness of Synchronous Languages for the Development of Safety–Critical Systems." Esterel Technologies, France, 2003.**

This paper describes the role of synchronous methods in the development of languages for safety-critical systems. The SCADE language and suite developed by Esterel Technologies, France is used as an example. An embedded software application is essentially a control loop. In SCADE, block diagrams are used for continuous control and safe state machines are used for discrete control to model system state. Based on formal semantics and a synchronous model used by the underlying LUSTRE language, SCADE simplifies code verification. One of the other key requirements for embedded systems is determinism and this is provided by a cycle-based computation model.

The paper has an important impact in the area of software for safety-critical systems because it advocates a different mathematical model. The success of SCADE is enhanced by the fact that KCG, which is a SCADE to C compiler, is DO-178B certified. Depending on the extent to which the various tools in the SCADE suite are used, there can be a 60% reduction in the software development life cycle. SCADE is being used by several firms in the avionics industry including Airbus and Eurocopter.

7. **Charles E. Billings. *Aviation Automation: The Search for a Human-Centered Approach*. Lawrence Erlbaum Associates, Mahwah, New Jersey.**

This book is foundational in the understanding of the human factors on the flight deck. Additionally, it surveys a great many specific components of avionics, past, present and future, from a human-factors standpoint.

As this book examines aviation automation as a whole, considering human factors related to the design of the automation, it surveys a wide range of systems too numerous to list exhaustively. In particular, the following aircraft types are mentioned in the discussion of specific automation components: Boeing 747, 747-400, 757, 767, and 777; Airbus A310, A320, A330, and A340; and the MD-11.

8. 767-400ER Flight Deck. The Boeing Company. <http://www.boeing.com/commercial/aeromagazine/aero_09/flight_textonly.html#fig03>

This website discusses the improvements in the displays, including the Primary Flight Display and Navigational Display, and controls on the new Boeing 767-400ER. It also talks about some of the design goals for the aircraft as well as improvements in the maintenance and dispatch reliability, training requirements and capacity for growth. It does not talk about design issues or tradeoffs considered, rather the final design and features of the systems are listed.

Since this website provides information about the newest Boeing aircraft, it serves as a looking glass into the newest technology onboard commercial aircraft. Since a primary objective was to integrate the aircraft into the existing fleet of 757 and 767 planes, most of the displays on this new aircraft are very similar if not identical to previous older aircraft. It does not seem to provide a solution to many of the problems discussed in other sources, like mode confusion.

9. Boeing. "Operator Benefits of Future Air navigation System." AERO 2 April 1998. <http://www.boeing.com/commercial/aeromagazine/aero_02/fo/fo02/index.html>

This paper addresses the reasons under which the Future Air Navigation System (FANS) was developed. It also elaborates on the many benefits of FANS over technologies that were and are being used. These include a reduction in the amount of separation needed between airplanes, more efficient route changes, and more direct routing. The paper claims these benefits will come as a result of satellite communications systems in which information can be shared between airplanes, other airplanes, the air service provider, and air traffic control. The use of satellite radio navigation technology was also claimed to aid these improvements.

Additionally, it describes the motivating factors, design issues, and deployment strategies used by the Boeing Corporation for FANS-1, their first implementation of FANS. FANS-1 was designed to include all of the following features: an airline operational control data link, automatic dependent surveillance, and air traffic control data link, GPS integration, required navigation performance compliance, and required time of arrival functionality.

This is extremely important to navigation because it describes how one of the top two aircraft manufacturers is approaching the future of aircraft navigation. By embracing FANS and placing it in its aircraft, Boeing contributed to the adoption of FANS as a navigation standard.

10. Brière, Dominique, and P. Traverse. "AIRBUS A320/A330/A340 Electrical Flight Controls: A Family of Fault-Tolerant Systems." Proc. 23rd International Symposium on Fault-Tolerant Computing (FTCS-23), 1993, pp. 616-623. June 1993.

This paper provides a description of the fly-by-wire flight control system aboard the Airbus A320 and compares it to traditional mechanical control systems. It also discusses the system validation procedure.

The paper is significant because the system it describes is the first such system to be used on a passenger-carrying commercial jetliner. It provides a comprehensive discussion that includes the principles of fly-by-wire, failure recovery modes of the A320 FCS, flight envelope protection, and an overview of the system architecture.

11. Butler, R. W., and Finelli, G. B. "The Infeasibility of Experimental Quantification of Life-Critical Software Reliability." ACM SIGSOFT '91 Conference on Software for Critical Systems, New Orleans, LA, December 1991.

The problem examined is whether software reliability can be quantified using statistical models. Reliability growth models and models for software fault tolerance are discussed in this paper. It is shown that these models are incapable of assuring reliability for software which is to be used for safety-critical systems. In general, software testing is infeasible for such systems.

12. **R. D. Campbell and M. Bagshaw. *Human Performance and Limitations in Aviation*. Oxford, UK, Blackwell Science Ltd., 2002.**

This seminal text addresses the physiological and psychological constraints regarding human performance in the aviation domain. The last chapter addresses advanced cockpit automation entirely. Before focusing on automation alone, the book provides an in-depth review of the factors that stem from limitations of the human body and the organs and systems associated with cognition.

13. **Chapman, R. "SPARK—a state-of-the-practice approach to the Common Criteria implementation requirements." Praxis Critical Systems Limited, Presented at the 2nd International Common Criteria Conference, Brighton, UK, July 2001.**

This paper discusses the role of SPARK in development of dependable systems. The programming languages used for such systems should have code that is unambiguous in meaning. SPARK, which is subset of the high-level language Ada, provides answers. SPARK has several useful properties including simpler code verifiability, secure code, bounded time and space requirements, and unambiguity. The paper also describes the role of annotations in SPARK, which are comments provided by the developer, go a long way in ensuring correct data and information flow as well as code verification. The SPARK Examiner is a power tool. It can do static analysis of the code as well as provide proofs of correctness using Verification Conditions. The paper also talks about the use of SPARK in building MULTOS CA, a secure operating system. Use of SPARK in several aerospace applications has also led to their DO 178B certification.

14. **Collinson, R.P.G. "Fly-by-wire flight control." *Computing & Control Engineering Journal*, Aug. 1999, pp. 141-152.**

This paper discusses fly-by-wire flight control in general using the Boeing 777 and military systems such as the Black Hawk as examples. It also discusses the survivability of fly-by-wire systems including their resilience to electromagnetic interference.

Collinson treats fly-by-wire flight control at the system architecture level. He provides block diagrams of the electrical linkages between the pilot and the control surfaces and of the basic feedback loops found in FBW systems. He also shows how FBW flight control is useful in aerodynamically unstable aircraft.

15. **Collinson, R. P. G. *Introduction to Avionics Systems*. Boston: Kluwer Academic, 2003.**

This book has a section on Flight Management Systems (FMS) under the "Autopilots and flight management systems" chapter. The first section gives an overview of FMS functionalities and architecture. In this section, the author briefly describes the A330/A340 FMS architecture and uses MCDU diagrams of the A330/A340; however he does not get into details of any implementation differences across manufacturers.

The section on autopilots describes basic principles and provides detailed mathematical equations describing the autopilot height and heading control. The section on the FMS was much more readable and informative than the autopilot section. Finally, this book does not provide any information on specific design issues for either FMSs or Autopilots. This book provides an introduction to avionics systems. Therefore, its biggest impact in the aviation field is on beginners who are just being introduced to the complexities within avionic systems. Also, it may serve as a quick and helpful guide book for the experts.

16. **Corwin, William H. "Autoflight Mode Annunciation: Complex Codes for Complex Modes." Honeywell Systems & Research Center.**

This paper begins with an example of an incident in which an aircraft enters a stall at 29,800 ft because the pilot enters a command into the autopilot in the wrong mode. This serves as motivation throughout the paper for the study of autopilot mode displays. The paper then gives a brief introduc-

tion to the purpose and function of an autopilot and autothrottle systems. It also depicts the relationship between the Mode Control Panel (MCP), the Primary Flight Display (PFD) including the Flight Mode Annunciator (FMA), and the Flight Management System (FMS) as they relate to the use of the autopilot system. The paper then discusses the results of a survey conducted by Honeywell of 105 pilots about problems using autopilots and autopilot mode confusion. In particular, the survey questions focus on whether there is a problem with the FMA or the MCP that causes pilots to make mistakes or become confused.

This paper establishes that there are flaws that exist in the display of autopilot modes involving the FMA and MCP. It further impacts the field because it makes clear that the lack of standardization of FMAs is a problem that needs to be addressed and "could prove to be the single biggest 'growth' area for accident research in automated aircraft."

17. Fielding, C. and Luckner, R. "Industrial considerations for flight control." ***Flight Control Systems*****. Pratt, Roger W. (ed.) Stevenage: IEE Press. 2000.**

This chapter of Pratt's *Flight Control Systems* provides a detailed introduction to flight control systems using several military and civil aircraft as examples, including several models of Boeing jetliners up through the 777, the Airbus A320 and A330, the Pavina Tornado, and the Eurofighter Typhoon.

Fielding and Luckner provide an introduction to flight control systems for the novice reader who wishes to quickly become familiar with these systems. They touch on nearly all aspects of flight control in both civil and military systems including basic control laws and a description of a generic FCS development process.

18. D. C. Foyle, A. H. Ahumada, J. Larimer, and B. T. Sweet. Enhanced/synthetic vision systems: Human factors research and implications for future systems. ***SAE Transactions: Journal of Aerospace*****, 101, 1734-1741, 1992.**
<http://human-factors.arc.nasa.gov/ihi/hcsl/pubs/Foyle_SAE92_syntheticvision.pdf>

This paper reviews research related enhanced and synthetic vision at the Aerospace Human Factors Research Division at NASA Ames. The four main areas covered are Head-Up Displays, Scene Augmentation, Sensor System Characteristics and Sensor Fusion. It reviews how to display flight guidance information and symbology on conventional head-up displays (HUD). It reviews pathway in the sky displays and scene-linked HUD in the scene augmentation section. It reviews field of view and infrared imagery in the sensor system characteristics section and multi-resolution image analysis, noise analysis and sensor fusion image in the sensor fusion section. HCI issues discussed include presenting and superimposing symbology on HUDs, the mean recognition time for terrain and non-terrain objects using either TV or IR imagery.

19. Georgiades, Paris. ***Data Link Manual: The Means for Future ATM*****. International Federation of Air Traffic Controllers' Associations, 2003. <http://www.ifatca.org/pdffiles/datalink.pdf>**

This paper describes how the data bus features of FANS-1 and FANS-A are implemented using the Aircraft Communications Addressing and Reporting System (ACARS). Since ACARS was not originally intended to be used with the Future Air Navigation System, the paper claims that it has a variety of problems. These include its difficulty to control access to proprietary networks and manage frequency allotment. It also has many performance constraints. To remedy these problems, ACARS will be replaces with the Aeronautical Telecommunications Network (ATN). The paper claims that this technology will be advantageous because it has an established protocol and equipment, was designed to support airline operational control, supports aeronautical administration communications and aeronautical passenger communications, and applies error checking at each layer of its architecture.

The paper is important because ATN technology will be used for communications in the next generation of FANS technology. Given that FANS is slowly growing as the navigation standard for commercial airliners, ATN technology will likely be used heavily in the near future.

20. Industry Projects. Aerospace Technology. <http://www.aerospace-technology.com/projects/index.html>

This website describes the equipment used in the construction of many different aircraft. Of the most important to this project is the characterization of the navigation systems in a variety of different commercial airliners. These include the Airbus A300-600, the Airbus A330 and A340, the Airbus A380, the Boeing 767, and the Boeing 777.

While this particular article has had little effect on the industry, it does accurately portray its state. It does this showing how different manufacturers have actually implemented navigation systems in their different aircraft models. It also provides an interesting contrast between the types of navigation technology used between models meant for different purposes.

21. Kayton, Myron, and Walter R. Fried. *Avionics Navigation Systems*. New York: John Wiley and Sons, Incorporated, 1997.

This book thoroughly describes the general state of navigation technology. It does this by defining what navigation is, how it relates to avionics, how navigation technologies are categorized, how the different technologies work, and how navigation has evolved. Among the specific system types covered are: terrestrial radio navigation, satellite radio navigation, inertial navigation, map matching navigation, star tracking navigators, compasses, and pitot tubes.

Additionally, this source contains information pertinent to the design of navigation systems. It does this by first characterizing the general form of navigation systems where information from different navigation systems is used to probabilistically determine the most accurate description and present it to the flight crews and other systems. Secondly, the tradeoffs that must be considered in the design of any navigation system are discussed. Such tradeoffs include cost, accuracy, coverage, autonomy, latency, capacity, and integrity. Finally, this source describes the software development process for navigation systems.

Many experts regard this as the definitive information source on aircraft navigation systems. It has been used as a text book in many courses on aircraft navigation.

22. J. R. Kerr, C. H. Luk, D. Hammerstrom, and M. Pavel. "Advanced integrated enhanced vision systems." SPIE Aerosense, April 21-25, 2003, Orlando, Florida, 2003. http://www.cse.ogi.edu/~strom/papers/spie03.pdf

This paper states the basic rationale for enhanced vision. It presents the evolution of sensors that enhanced vision systems use and image processing techniques. The authors are pursuing Bayesian fusion, a neural-net based approach that can be implemented using FPGA-based boards at minimal cost.

23. Knight, J. C. and N. G. Leveson. "An Experimental Evaluation of the Assumption of Independence in Multi-version Programming." *IEEE Transactions on Software Engineering*, 12(1), January 1986.

This paper discusses issues with the n-version programming method of incorporating fault tolerance into software. With n-version programming, multiple versions of a program are prepared and executed in parallel. The outputs are collected by a voter and, if there is disagreement, the results of the majority are assumed to be correct. This method hinges on the assumption that programs that have been developed independently will fail independently.

In this paper, an experiment is described which tests the assumption of independence. In the experiment, graduate and senior level students at two universities wrote programs from a single

requirements specification. A total of one million tests were run on the 27 software versions produced. The test failures were recorded, and it was found that multiple versions failed on the same inputs. The paper concludes that the assumption of independence of errors does not hold, and that further experiments should be performed before n-version programming is used in crucial systems.

This paper is important in the field of dependable systems and avionics, because it suggests that a technique commonly employed in dependable systems may not be as dependable as is assumed.

24. W. J. Kubbat, P. M. Lenhart, and H. V. Viebahn. "4d flight guidance displays, a gate to gate solution." 17th DASC-Electronics in Motion, 1998.

This paper argues that a geographical database, a precision navigation system and 4D flight guidance displays are the three pillars of synthetic vision. The authors argue that synthetic vision will greatly reduce CFIT accidents. The authors propose ideas for how the primary flight displays and navigational displays will interact with the synthetic vision system.

25. Nancy Leveson. *Safeware: System Safety and Computers.* Addison Wesley, 1995.

This book is foundational for understanding issues related to computer systems and safety. It is an important reference for discussions of human-centered design, as it considers interactions between humans and automation. It does not deal with specific systems.

Chapter 5 deals with human error and risk and addresses the need for humans in automated systems. Chapter 6 deals with the roles played by humans in automated systems: the human as monitor, backup and partner, as well as addressing mental models. Chapter 10 deals with Accident and human error models and includes a section on human task and error models. Finally, Chapter 17 deals with the design of the human-machine interface and includes sections on training and skill maintenance and guidelines for safe HMI design.

26. Lidén, Sam. "The Evolution of Flight Management Systems." IEEE, 1994, 157-169. Paper 0-7803-2425-0/94.

This paper has very useful information on the history and development of the FMS. After providing a brief overview of the FMS functionality and components, the author talks about first generation FMS and how the first generation affected the later systems.

The author discusses some specific design issues such as differences between the gray and the brown systems, and talks about the factors that caused these differences. There is also a comparison of the current Gray and Brown systems which was extremely helpful. The paper provides some design information on various FMS, such as the name of the manufacturer, some of the hardware used in the system, and improvements based on the previous systems. Some example FMS improvements discussed in this paper were to the MD-80, Boeing 777, MD-11, and A320/A321 FMSs. This paper was definitely one of the most useful sources for our FMS research.

In conclusion, this paper talks about some specific design issues of gray and brown FMS systems. It also addresses the functionality and design improvements of various FMS across different airframes. The biggest impact of this paper is that it makes rare information accessible. It summarizes two main FMS designs, as well as providing improvements made to FMSs on specific aircraft. Most FMS design information is proprietary and it is difficult to find comparisons or improvement summaries.

27. H. Lowery, and B. Mitchell. "Mission Computer Replacement Prototype for Special Operations Forces Aircraft - An Application of Commercial Technology to Avionics." Proceedings of the 19th Digital Avionics Systems Conference. (October, 2002): 4.E.1-1 - 4.E.1-6.

A short paper that describes the existing and proposed prototype replacement ("mission computer") for avionics architectures aboard the AC 130 and MC 130, a special operations variant of the

gunship, proposed by the Warner Robins Air Logistics Center. As the avionics systems aboard these aircraft continue to age, maintenance and replacement of legacy systems is seen as cost prohibitive. The Mission Computer replacement constitutes a Line Replaceable Unit configured with contractor-modified commercial circuit cards. For experimentation, prototypes had been fielded and the authors were awaiting the results of ground tests aboard AC-130s. There is no mention in the literature of the subsequent success or failure of this initiative.

28. Nelson, Robert C. *Flight Stability and Automatic Control.* New York: McGraw-Hill. 1989.

This book provides a rich source of information for the aiding the readers' understanding of the fundamentals of flight dynamics. The principles of stability, flight control design and stability augmentation are discussed at length in both physical and mathematical terms.

Nelson's text may be considered a reference source for the more advanced reader interested in delving into the physical principles governing aircraft flight.

29. Steven P. Miller and James N. Potts. "Detecting Mode Confusion Through Formal Modeling and Analysis." NASA/CR-1999-208971.
<http://techreports.larc.nasa.gov/ltrs/dublincore/1999/cr/NASA-99-cr208971.html>

This paper examines sources of mode confusion in a theoretical Flight Guidance System. It presents a formal methods technique for reasoning about mode confusion in the system and detecting potential sources of mode confusion prior to system deployment. It demonstrates a system for mode visualization that is based upon a notation similar to Statecharts.

30. Moir, I., Seabridge, A. *Civil Avionics Systems*. Reston, VA: American Institute of Aeronautics, 2003, pp. 273 – 281.

This section of the book deals with the controls of a typical autopilot and autothrottle system. It has systems diagrams (design diagrams) for the autopilot pitch control, yaw control as well as the autothrottle or speed control. It also lists a number of modes usually found on an autopilot system but does not detail the function of each mode. It provides a chart which gives information about which sensors are needed for a particular autopilot mode to function. The integrated autopilot system found in the Boeing 777 is briefly discussed and a diagram of the system is provided.

The impact of this section of the book on the avionics field is that it is one of the only public sources available that provides diagrams for the general autopilot control channels.

31. M. T. Palmer et. al. "A crew-centered flight deck design philosophy for high-speed civil transport (hsct) aircraft." NASA Technical Memorandum 109171, 1995. http://techreports.larc.nasa.gov/ltrs/PDF/NASA-95-tm109171.pdf

This report raises the reconsideration of overall flight deck design. The motivation for this is due to the complexity of modern avionics and the partial or full opaqueness of these systems to the flight crew. The authors propose implementing a written design philosophy that focuses on human performance and system operability. The design philosophy would be a crew-centered philosophy placing emphasis on the overall flight crew/flight deck system. The philosophy seeks to elevate human performance to the same levels as technological and mechanical/structural issues. It presents design principles in terms of pilot roles.

32. Charles Perrow. *Normal Accidents*. Princeton University Press. 1999

Perrow argues that for certain kinds of systems, particularly highly complex, tightly coupled systems, accidents are a normal property of the system. He cites many fields in which this is the case, including the nuclear industry and the aviation industry. This book is mostly useful in that provides examples of human error from the past and suggests causes related to poor user interface leading to mode confusion and other error modes.

33. James Reason. *Human Error.* Cambridge University Press. 1990.

James Reason gives a systematic treatment of human error based on cognitive psychology. In Chapter 1 he suggests why errors occur. In Chapter 3 he categorizes error types. In Chapter 9 he presents some suggestions for assessing and reducing the risk of human error.

34. W. Reynish. "Enhanced vision: Final link in situational awareness." www.defensedaily.com, 2004. http://www.defensedaily.com/cgi/av/show_mag.cgi?pub=av&mon=0104&file=0104enhanced_vision.htm

This article discusses enhanced vision systems as a technology to increase situational awareness and introduces non-HUD, head down, enhanced vision systems technology, that is a much cheaper alternative to HUD enhanced vision systems technology. It also mentions the need for short, medium and long range infrared sensors in enhanced vision systems technology and FAA regulations concerning enhanced vision systems.

35. Riley, V., B. DeMers, C. Misiak, and B. Schmalz. "A Pilot-Centered Autoflight System Concept." Digital Avionics Systems Conference, IEEE. 1998.

This paper addresses the complexity of the autoflight systems by providing a new human centered design. With their design, the authors integrate the autopilot and flight management system functions into a single user interface. According to the author's claim, most pilots can learn to operate this new interface in fifteen minutes; however, they do not provide any detailed explanation or scientific proof to support this claim.

The authors choose to apply a user-centered design in order to avoid the gap between how the engineers think and how the pilots think. The authors claim that the new design will naturally reduce the need for pilot training. Similar to the desktop metaphor used in the computer industry, the authors of this paper employ the cockpit control language metaphor. In the end, the authors provide prototype illustrations for the new design. They indicate that the prototype was shown to one hundred and fifty pilots but they do not provide any usability test results.

This paper has an impact on future designs of aviation equipment by showing how the use of human factors ideas can improve design. Ease of use is very important in avionics to reduce the training costs and to improve safety. The user-centered approach that this paper provides is one that designers should investigate.

36. Rushby, J. "Bus Architectures For Safety-Critical Embedded Systems." Lecture Notes on Computer Science 2211 (2001): 306-323.

This paper focuses on design issues of bus architectures for safety-critical embedded systems. Rushby first looks at general issues in bus architectures, including their evolution for use in integrated systems. Basic bus components, interconnect configurations, types, and the issues with each are depicted. Next, the faults that buses must tolerate are described. Finally, the paper describes four representative architectures: SAFEbus, TTA, SPIDER, and FlexRay.

First, Rushby examines Honeywell's SAFEbus. The paper describes the motivation for developing SAFEbus (needed for the integrated Boeing 777 Airplane Information Management System), the bus's architecture, and its positive and negative aspects. Second, the paper describes TTA, developed by Hermann Kopetz and colleagues at the Technical University of Vienna. The TTA architecture is described, as well as its positive aspects. No information is provided about the bus's success in deployment. Lastly, the paper describes SPIDER, a bus being developed by Paul Miner and colleagues at the NASA Langley Research Center. The bus architecture is described; however it is not compared with the other buses listed. FlexRay is not of interest, as it is designed for use in cars.

This paper is a good introduction to bus architectures in general and the specific bus architectures designed for integrated systems.

37. Sanford, B. "Airbus Flight Control Laws." <http://www.airbusdriver.net/airbus_fltlaws.htm>

This document lists the flight control laws for the Airbus A320/330/340/380 flight control systems.

Sanford's description of the Airbus flight control laws is one of the most thorough descriptions available. Although the document is entirely factual, it reveals some of the design decisions Airbus made in deciding what levels of service to provide and when to change service levels based on detected failure conditions.

38. "Satellite Navigation Product Teams." Federal Aviation Administration. <http://gps.faa.gov/index.htm>.

This website served as the principal resource on the satellite radio navigation Global Positioning System (GPS) and the more accurate Differential GPS technologies, the Wide Area Augmentation System (WAAS) and the Local Area Augmentation System (LAAS). It describes what each of these technologies are, what purpose they serve, how they work, and what news they are generating.

GPS, WAAS, and LAAS extrapolate position based on communications delays with satellites. WAAS and LAAS have the added benefit of utilizing a terrestrial network of known, fixed GPS receivers that transmit signal corrections to aircraft, thus improving the original GPS signal.

GPS, WAAS, and LAAS have all had extraordinary influences on airplane navigation. GPS has been incorporated into the navigation systems of nearly every commercial airliner. WAAS has just been approved for use and will likely appear in commercial aircraft in the near future. The developers of WAAS and LAAS claim that these products' key selling point is how much cheaper the development of navigation systems will be because the need for other navigation technologies will be eliminated. This has created concern amongst many experts who feel that such a lack of independent redundancy will prove dangerous.

39. Schmitt, V. R., J. C. Morris, and G. D. Jenny. *Fly-by-Wire: A Historical and Design Perspective.* Warrendale, PA: SAE International. 1998.

This source provides a background of fly-by-wire flight control systems beginning with the guided missiles and autopilots of the World War II era up to the design and performance considerations of the Boeing 777. The book surmises the current state of the industry for both commercial and military aircraft manufacturers. Each major system and its design considerations are fully addressed including control input methods, flight control computers, and control surface actuators.

This book provides an excellent overview of the development of FBW FCS citing a plethora of example airframes for the reader to identify with. By outlining FBW design in terms of developmental milestones the concessions made in modern aircraft design are made more readily apparent… without being overly technical.

40. P. C. Schutte and K. F. Willshire. "Designing to control flight crew errors." NASA Technical Memorandum, 1997. http://techreports.larc.nasa.gov/ltrs/PDF/1997/mtg/NASA-97-ieeesmc-pcs.pdf

The authors state that current flight deck designs take human physical behavior, but not cognitive behavior, into account. They also mention the reluctance to change the current flight deck since it does function well (as evidenced by low aircraft accident rate). The authors state that the human operator must be in command as a basic principle of human centered flight deck design. They also cover design philosophy, function allocation and involvement, flight deck mission categories (flight management, communications management, systems management and task management), task oriented display designs and fault management.

41. Seeling, K. A. "Reconfiguration in an Integrated Avionics Design." Proceedings of the 15th AIAA/IEEE Digital Avionics Systems Conference, Atlanta, GA, October 1996.

This report covers the fault tolerance and reconfiguration implementation of the F-22 core processing subsystem. This includes the core processing architecture, detection of failures, recovery, and return to operations. The core processing subsystem features Common Integrated Processors (CIPs). CIPs can reconfigure based on both hardware and software-detected failures. Various types of reconfiguration are possible. For instance, a minor reconfiguration changes the distribution of software on the available hardware resources, while a major reconfiguration may reallocate function to another CIP or replace function with a degraded functional load. The type of reconfiguration employed is based on the currently available resources the current priorities of avionics functions.

The author discusses only the fault tolerance approach used by the Lockheed Martin Aeronautical Systems Company. He states, "it is hoped that the fault tolerance concept and reconfiguration design approach reflected herein, will be useful to other designers of other designers of integrated computing systems." No experimental results are given. Thus, this paper is very helpful in giving an overview of the fault tolerance design of the F-22; however, the reader must look elsewhere for information about the effectiveness of the design.

42. Sherry, L., Deborah, M. D., "Why the FMC/MCDU is Hard to Train and Difficult to Use." IEEE, 2003, Paper 0-7803-7844-X/03.

This paper discusses the difficulty of FMC/MCDU use for the pilots and conducts an analysis to reveal the reasons for the acknowledged difficulty. After identifying the reason as over-reliance of the system on memorized actions, the paper provides implications for training and design of new interfaces. The suggestions the authors make to improve the FMS training includes teaching the underlying structure of how the system works and providing more memorization training. The design guidelines provided in this paper also aim to address the over-reliance on memory problem by entailing more visual cues, labels and user prompts.

The authors base their results on the analysis of 102 mission tasks for the Boeing 777 MCDU. This paper has an important impact on the aviation field because it draws attention to a very important problem that has not yet been addressed. The non user-friendly interface of FMC/MCDU not only causes more heads down time for the pilots, but also increases training costs. The implications provided for the training are quite useful; however the design implications are not as specific as they could be.

43. Signargout, L. "AIM-FANS: A Flexible Approach to CNS/ATM." FAST June 1995. <http://www.airbus.com/pdf/customer/fast17/p27to31.pdf>

This article describes the airbus approach to the deployment of FANS technologies. It describes the goals of the AIM-FANS organization Airbus created to handle the deployment of FANS. It also discusses Airbus's development of FANS-A, its first implementation of FANS, and makes projections about the evolution of FANS technology. The goals of AIM-FANS established for FANS-A include all of the following: it would be adaptable; it would be easy to transition to form other systems; it would establish retrofitting standards; it would minimize the effects on peripheral equipment; and it would be user friendly.

This article is important because it demonstrates how one of the largest commercial airliner manufacturers is approaching the shift to FANS technology. The fact that Airbus is adopting FANS gives direction to the future of airline navigation.

44. K. Smith and G. Miyahara. "Leveraging an Avionics Support Environment for Shared Applications to Multiple Platforms." Proceedings of the 16th Digital Avionics Systems Conference. (October 1997): 1.3-9 - 1.3-15.

This paper describes the software systems and common support environment for the APQ-180 ground attack radar found aboard the AC-130. The authors discuss the commonality between the APQ-180 software and that of the APG-70 software, which is normally found on the F-15. This further exemplifies the AC-130 role in ground attack and special operations, as well as exposing some of the architectural details of the APQ-180 and how software for it and the APG-70 are developed together. Since these two systems share much of the same functionality and are highly complex, there is tremendous cost savings in this development approach.

45. Sparaco, Pierre. "Human Factors Cited in French A320 Crash." Aviation Week & Space Technology, January 3, 1997, pp. 30 – 31.

A number of possible causes for the A320 crash in Strasbourg, France in 1992 are presented. Of these factors, mode confusion and a subsequent mode error make the top of the list. The article presents a number of other possible contributing factors, many of which were human factors issues, which may have led to the crash of the aircraft as it attempted to land. Low flight hours by the pilots as well as the lack of a Ground Proxy Warning System (GPWS) are also noted as possible factors.

This paper highlights the fact that human factors issues in the cockpit are of critical importance. The impact of this paper is that the implications of man/machine interfaces is not fully understood and should continue to be investigated. Understanding the interaction between man and machine is of immediate importance as more advanced technology is being implemented in civil aircraft.

46. Spitzer, C. R. *The Avionics Handbook*. New York: CRC Press, 2001.

This book provides valuable information to anyone who is not familiar with various elements of aviation. Various sections of the book are dedicated to flight control, navigational systems, flight management systems and so forth. For our paper, few chapters provided valuable information. Chapter 25 was on Electromagnetic Interference. Because of the increasing use of electronics on all aircraft, faults can be induced in the hardware due to interaction with energy radiating from various sources. These include lightning, electronic jamming devices and portable electronic items aboard. Chapter 23 discussed certification of systems in avionics. It talked about in detail the various phases required for the certification process. Chapter 27 was devoted to the software development process and how DO-178 B is used widely for certification for civil aircraft. Chapter 28 was on computing platforms and dependability requirements for avionics.

47. Tischler, Mark B. (ed.) *Advances in Aircraft Flight Control.* London: Taylor & Francis. 1996.

This text should be regarded as a primary source for some of the more advanced FBW features of the Airbus family DFCS equipped aircraft. It is relevant for its thorough description of the Airbus lineage of FBW aircraft. The text contains a discussion of the lessons learned, design methodologies and envelope protection systems as implemented on the A320, 330 and 340 aircraft.

Tischler's book provides the single best source for insight into the Airbus design philosophy. It is exceptional for its historical overview of the of the Airbus FBW development program.

48. M. Tischler, J. Colbourne, M. Morel, D. Biezad, K. Cheung, W. Levine, and V. Moldoveanu. "A Multidisciplinary Flight Control Development Environment and Its Application to a Helicopter." IEEE Control Systems Magazine. 19. (August, 1999): 22-33.

This article describes the flight control system development of the baseline UH 60 airframe using the CONDUIT system. This was a joint venture of the U.S. Army Aeroflightdynamics Directorate, NASA Ames Research Center, the University of Maryland, the California Polytechnic State University (San Luis Obispo) and Raytheon ITSS.

49. Uczekaj, J.S., "Reusable avionics software. Evolution of the flight management system." Digital Avionics Systems Conference, IEEE. 1995.

Uczekaj looks at the evolution of the reuse of the Honeywell FMS software, from the first generation FMS of the 1980's (Boeing 757/767, Airbus 310) to the advanced FMS designs of 1990's. This paper points out the difficulties the developers faced due to the customer's desire to add more capability and functionality along with unwillingness to pay full development costs in 1980's. The paper also talks about Honeywell's reusability strategic plan as a solution to the problem. At this point, the author gets into the revolutionary design of the Boeing 777 program and describes the use of partitioning in this system. Finally, the paper presents the lessons learned and anticipated future challenges in reusable strategies.

In summary, this paper studies the necessity of reusability for FMS and talks about a particular manufacturer's (Honeywell) reusability strategy. The biggest impact of this paper is that it brings up a very important topic: reusability in FMS design and production. It reveals Honeywell's strategy as well as the company's lessons learned which would be helpful in improving current FMS reusability techniques.

50. Yeh, Y. C. "Triple-Triple Redundant 777 Primary Flight Computer." Proceedings of the 1996 Aerospace Applications Conference. IEEE Computer Society Press: Los Alamitos, CA, 1996, pp. 293-307.

This paper focuses on the Boeing 777 Fly-By-Wire (FBW) system. Particular attention is given to the Primary Flight Computer (PFC), the central computation element of the FBW system. First, the paper gives an overview of the FBW system. Included is an explanation of the safety constraints considered during the design process, such as common mode faults.

Additionally, the paper focuses on the design of the 777 PFC architecture and the PFC's safety requirements. It is explained that the PFCs contain three similar channels, with each channel containing three dissimilar computation lanes. N-version software dissimilarity is also utilized.

This paper is useful in illustrating the state of practice in designing a FBW system with extremely high levels of integrity and availability. It outlines the safety considerations and requirements of the PFC and then describes its fault-tolerant design.

51. Yeh, Y.C. "Design Considerations in Boeing 777 Fly-By-Wire Computers." Proc. Third IEEE International High-Assurance Systems Engineering Symposium, pp. 64-72. November 1998.

In this article, Yeh probes deeper into the inner workings of the Boeing 777 flight control system with an emphasis on the methodologies of fault detection and containment. The author considers the redundancy of hardware, data buses and the error handling capability of the software. In addition, the above considerations are discussed in terms of impact on safety and the introduction of the concept of "deferred maintenance."

Yeh's article is notable for the depth of detail that it goes into the 777's FCS. There are several helpful figures to accompany the textual description of the aircraft's inner workings.

52. Yeh, Y.C. "Safety Critical Avionics for the 777 Primary Flight Controls System." Digital Avionics Systems (DASC) 2001, the 20th Conference, vol. 1, pp. 1C2/1-1C2/11. October 2001.

This paper should be considered a primary source for anyone interested in fly-by-wire control system of the Boeing 777. In it, the author addresses the design considerations of the 777 FCS with an emphasis on their impact on flight safety. The author provides a high-level overview of the 777 FCS hardware and software packages before discussing the considerations of using a completely FBW system onboard a commercial aircraft. Topics of discussion include: error handling, hardware failure and levels of pilot authority in the event of a failure.

List of Acronyms

ACARS: Aircraft Communications Addressing and Reporting Systems
ACE: Actuator Control Electronics
ADC: Air Data Computer
AFDS: Autopilot Flight Director System
AGL: Above Ground Level
AHRS: Attitude and Heading Reference System
AIM-FANS: Airbus Interoperable Modular Future Air Navigation System
AIMS: Airplane Information Management System
AOC: Aeronautical Operational Control
APC: Aeronautical Public Correspondence
API: Application Programming Interface
ARINC: Aeronautical Radio INC.
ATC: Air Traffic Control
ATN: Aeronautical Telecommunication Network
BG: Bus Guardian
CDU: Control/Display Unit
CFIT: Controlled Flight Into Terrain
CMS: Cockpit Management System
CNS/ATM: Communications Navigation Surveillance/Air Traffic Management
COTS: Commercial Off The Shelf
CRT: Cathode Ray Tube
DME: Distance Measuring Equipment
E-MACS: Engine Monitoring And Control System
EFIS: Electronic Flight Information System
EGPWS: Enhanced Ground Proximity Warning System
EICAS: Engine Information and Crew Alerting System
EMI: ElectroMagnetic Interference
EUROCAE: EUropean Organization for Civil Aviation Equipment
FAA: Federal Aviation Administration
FAA AC: Federal Aviation Administration Advisory Circular
FADEC: Full Authority Digital Engine Control
FANS: Future Air Navigation System (a.k.a. CNS/ATM)
FANS-1: The Boeing implementation of FANS
FANS-A: The first Airbus implementation of FANS
FCS: Flight Control System

FMA: Flight Mode Annunciator
FMC: Flight Management Computer
FMCS: Flight Management Computer System
FMS: Flight Management System
FMSP: Flight Mode Selector Panel
FPGA: Field Programmable Gate Array
GLONASS: GLObalnaya Navigatsionnaya Sputnikovaya Sistem or GLobal Orbiting NAvigation Satellite System
GPS: Global Positioning System
GPWS: Ground Proximity Warning System
HM: Hazard Monitor
HQM: Handling Qualities Model
HUD: Head Up Display
ICAO: International Civil Aviation Organization
IEEE: Institute for Electrical and Electronics Engineers
IFR: Instrument Flight Rules
ILS: Instrument Landing System
IMA: Integrated Modular Avionics
IMC: Instrument Meteorological Conditions
INS: Inertial Navigation System
JAA: Joint Aviation Authorities
LAAS: Local Area Augmentation System
LCD: Liquid Crystal Display
LNAV: Lateral NAVigation
LORAN: LOng RAnge Navigation
LRU: Line Replaceable Unit
MCDC: Modified Condition Decision Coverage
MCDU: Multifunction Control Display Unit
MCP: Mode Control Panel
MFD: MultiFunction Display
MSL: Mean Sea Level
NASA: National Aeronautics and Space Administration
ND: Navigation Display
NDB: Non-Directional Beacon
NMR: N-Modular Redundancy
OFM: Operator Function Model
PCU: Power Control Unit
PFC: Primary Flight Computer
PFD: Primary Flight Display
RNAV: aRea NAVigation
RNP: Required Navigation Performance
RTCA: meaningless acronym; formerly: Radio Technical Commission for Aeronautics
RWR: Radar Warning Receiver
SA: Situational Awareness
SATS: Small Aircraft Transportation System
SCADE: Safety Critical Application Development Environment
TACAN: TACtical Aid to Navigation
TCAS: Traffic alert and Collision Avoidance System
TMR: Triple Modular Redundancy
TRACON: Terminal Radar Approach CONtrol

TTP: Time Triggered Protocol
VFR: Visual Flight Rules
VHF: Very High Frequency
VMC: Visual Meteorological Conditions
VNAV: Vertical NAVigation
VOR: VHF OmniRange
VORTAC: VHF OmniRange/TACAN
WAAS: Wide Area Augmentation System
WFCS: Wireless Flight Control System

Glossary

Editors' note: many of these definitions are taken from other texts. For the citations, see the context in which they are defined in the main text.

absolute navigation systems: see *positioning navigation systems*
active flight plan: a flight plan programmed into the FMS that is being used for aircraft guidance
adverse yaw: see *aileron drag*
aerodynamic load: the forces placed on the airframe during flight, measured by *load factor*
aileron drag: rotation toward the raised wing, resulting from the increased drag that accompanies the increased lift produced by the wing with increased *camber*
ailerons: the hinged wing extensions designed to be deflected down or up into the airstream so as to increase or decrease the *camber* of the wing
airspace: the space in which aircraft are permitted to operate
airway: a specific linear route to which *enroute flight* is restricted
alternative flight plan: a flight plan programmed into the FMS that is selected if changes need to be made to the *active flight plan*
altitude measurement: an indication of the distance between an aircraft and either ground level or *mean sea level*
angle of attack: the angle at which an aircraft wing intersects the oncoming airflow in the *pitch* (*xz*) plane
angle of incidence: see *angle of attack*
angle of yaw: the angle formed between the aircraft and the oncoming airflow in the *yaw* (*xy*) plane
attitude: the orientation of an aircraft to the horizon; a combination of *angle of attack* and *bank angle*
autopilot: *autopilot flight director system* component providing *pitch*, *roll* and *yaw* control
autopilot flight director system: an avionics component that allows a digital flight guidance computer to control the aircraft instead of the pilot
autothrottle: *autopilot flight director system* component providing *thrust* control
availability: the probability that a system will be operating correctly at a specified time
bank angle: the angle formed between the aircraft and the ground in the *roll* (*yz*) plane
boosted FCS: a *flight control system* that mechanically augments pilot inputs to control surfaces
bus guardian: component that protects an *interconnect* from defective *nodes*
camber: curvature of a section of an airfoil
category I landing: landing that must be aborted if the runway is not visible at 200 feet
category II landing: landing that must be aborted if the runway is not visible at 100 feet
category III landing: landing that must be aborted if the runway is not visible at 50 feet

celestial navigation system: a *positioning navigation system* using a combination of *dead reckoning* and observations of heavenly bodies to determine a more precise location estimate than is possible through dead reckoning alone

centralized architecture: architecture in which computations take place in one or more computers in a centralized area of the aircraft

cockpit: see *flight deck*

confidentiality: the absence of unauthorized disclosure of information

control objectives: control column and rudder pedal deflections; serve as input to a digital *flight control system*

correctness by construction: software development approach in which software is essentially correct from the moment it is written (focusing on *fault prevention*)

damped oscillation: an oscillation in aircraft *attitude* that lessens in amplitude over time and converges on the *trimmed* state

dead-reckoning navigation system: a *navigation* system that derives an aircraft's position based on the aircraft's initial state and the integral of the aircraft's velocity over time

decision height: the point at which the pilot must decide whether to land or to execute a *missed approach*

degradation fault: a fault that is the result of an in-service failure of a previously functioning component due to wear on that component

dependability: the quality of a system's being fit for the use for which it is intended

design diversity: an attempt to overcome *design faults* by using several components of dissimilar design

design fault: fault present in a system that is introduced at design time

differential global positioning system: an enhancement to *GPS* that improves the accuracy of GPS by measuring the error between a fixed location and the GPS position of that location and broadcasting the offset to receivers in the area

digital flight control system: a *flight control system* in which pilot inputs are measured and converted into digital signals that are processed by a computer

directional gyroscope: see *gyrocompass*

directional stability: *stability* in *yaw*

distributed architecture: architecture in which several processors throughout the aircraft are assigned different computing tasks

dynamic redundancy: redundancy approach in which one unit is in use and one or more standby units are available

dynamic stability: *stability* exhibiting *damped oscillations*

earth-bound coordinate frame: coordinate frame that encompasses the entire earth and moves in space with the earth's rotation

earth-centered coordinate frame: coordinate frame in which the origin is at the earth's center and the coordinates of an object in space do not change as the earth rotates

electromagnetic interference: disturbance in a system caused by electromagnetic waves

elevators: control surfaces attached to the *horizontal stabilizer* that control the *pitch* of the aircraft

envelope protection system: software system responsible for keeping the aircraft within its *flight envelope* or aiding the pilot in doing so

enroute flight: stage of flight when an aircraft is neither approaching nor departing an airport.

error: deviation of observed output from intended output

EUROCAE ED-12B: avionics software standard document for European certification

event-triggered bus: bus that deals with events as they occur (not at a predefined time)

fault: problem with a system component

fault forecasting: the estimation of the number of *faults* remaining in a system prior to deployment, their future rate of occurrence, and the likely consequences of those faults

fault prevention: prevention of the occurrence or introduction of *faults*

fault removal: the reduction of the number or severity of *faults* that exist in the system

fault tolerance: the use of system structures that allow the effects of *faults* to be tolerated should they manifest themselves during operation

fault tree analysis: technique in which events that could lead to a hazardous state are identified

federated architecture: architecture in which each function is implemented in one or more LRUs

final approach fix: the point at which an aircraft should be positioned in a straight line to the runway

flaps: hinged wing sections designed to increase the *camber* of the wing, providing greater *lift* at slower speeds

flight control laws: the rules that govern the *handling qualities* of the aircraft in different phases of flight

flight control system: the system providing the basic interface between the pilot and the aircraft's control surfaces

flight deck: the part of the aircraft from which the crew controls the aircraft

flight envelope: the set of aerodynamic conditions under which an aircraft can safely fly

flight level: partitions of airspace based on altitude, expressed in units of 100 ft. above ground level or *mean sea level*

flight mode annunciator: the *AFDS* component that displays the current AFDS mode

flight mode selector panel: the *AFDS* component that allows the pilot to select, change, and engage all AFDS modes

flight plan: a series of *waypoints* specifying the route a flight is expected to take

flight planning: the identification of the correct sequence of *waypoints*, *flight levels,* & *airways*, and the arrangement of procedures for takeoff and arrival

fly-by-wire flight control: flight control based on electrical signal connections between pilot controls and aircraft control surfaces

form interference drag: drag due to the shape and smoothness of an aircraft

global positioning system: a *satellite radio navigation system* that is composed of 24 satellites and 6 ground stations

ground controller: an individual responsible for assigning aircraft to taxiways for movement over ground to an from a runway

gyrocompass: a heading sensor based on a gyroscope with gimbals allowing two degrees of freedom

handling qualities: aircraft responses to pilot inputs

horizontal stabilizer: part of the tail section of the aircraft that acts like a small wing to counteract the moment generated by the wings (adjusting aircraft pitch)

host: computer on which an application runs

hybrid redundancy: redundancy approach that combines static and dynamic redundancy

hydraulic flight control system: a *flight control system* based on hydraulic linkage of pilot controls and aircraft control surfaces

induced drag: drag resulting from the *lifting* force created by the airfoils

initial approach fix: the beginning of an *IFR* approach

instrument flight rules: flight procedures under which pilots are required to use instruments for navigation and guidance

instrument landing system: a radio navigation system used during landing that provides horizontal and vertical guidance via transmitters placed adjacent to runways

instrument meteorological conditions: weather conditions under which *instrument flight rules* are required

integrated modular avionics: architecture in which several functions are implemented on a single computing unit

integrity: absence of improper state alterations

interconnect: medium that provides communications between *hosts*

interface: (in distributed architectures) a device that connects a *host* to an *interconnect*

intermittent fault: fault that manifests itself temporarily but repeatedly and irregularly

Kalman filter: algorithm that calculates the best estimate of the state vector by considering the dynamics of each sensor

lateral navigation: navigation in the *xy*-plane, governed by *roll* and *yaw*

lateral stability: *stability* in *roll*

lift: the upward force generated by the movement of air around and over the wings of an aircraft

lift over drag maximum: the point at which *total drag* is lowest

line replaceable unit: physical computing components used in an avionics architecture

load factor: the magnitude of the force on the airframe, divided by the weight of the aircraft

local controller: an air traffic controller responsible for runway scheduling

longitudinal stability: *stability* in *pitch*

maintainability: the ability to have repairs or modifications made

map grid: grid overlaid on a map of the earth's surface

mapping navigation system: a *positioning navigation system* that use radars, optical sensors, and/or digitized video to map the terrain underneath the aircraft

mean sea level: the average sea level of the earth

mechanical flight control: flight control based on direct, mechanical linkage between pilot controls and aircraft control surfaces

missed approach: an aircraft's approach to a runway in which the aircraft did not land on the runway

mode control panel: see *flight mode selector panel*

model-based development: software development approach in which application experts define a software specification in a domain-specific language and synthesize code from that specification

modified condition decision coverage: test coverage metric denoting whether every decision in a program has been executed to ensure that every condition within each decision has been tested with values of True and False and has been shown to affect the flow of control directly

N-modular redundancy: redundancy technique in which each of N modules computes an output, and a voter then compares the outputs to mask faults in one of the modules

national airspace system: the *airspace* over the United States

navigation: the determination of the position and velocity of a moving vehicle

negative stability: being *statically unstable*, *dynamically unstable*, or both

neutral stability: being *statically stable* but *dynamically neutral*; exhibiting continued oscillation around the *trimmed* state

node: a *host* and its *interface*

normal stability: similar to *total stability*, but exhibiting longer intervals of *damped oscillation* about the *trimmed* state

ownship: term used to denote the aircraft on which a system resides

parasitic drag: friction, the resistance of an aircraft to the air through which it moves

permanent fault: fault that does not disappear

pitch: movement around the axis running roughly parallel to the wings of an aircraft

pitot tube: a sensor that derives airspeed by examining the pressure of the airstream

position location: see *surveillance*

positioning navigation system: a navigation system that calculate an aircraft state vector without considering the previous path of an aircraft

pseudolites: a terrestrially based pseudo-satellite, typically used for *LAAS*

radio navigation system: a *positioning navigation system* based upon networks of transmitters and receivers that interact with the vehicle

reliability: the probability that a system will operate correctly in a specified operating environment for a specified period of time

roll: movement around the axis running from nose to tail of an aircraft

RTCA DO-178B: avionics software standard document for U.S. certification

rudder: the hinged portion of the *vertical stabilizer* used to keep the aircraft aligned with the relative airflow, to avoid *slips* or skids induced by *adverse yaw*

runway incursion: an incident in which an aircraft or other obstacle is present on the runway when another aircraft is attempting land or take off

safety: the absence of catastrophic consequences on the users or the environment

satellite radio navigation system: a space-based *radio navigation system*

slats: extensions to the leading edge of the wing, allowing an increased *angle of attack* before the aircraft *stalls*

slip angle: see *angle of yaw*

small aircraft transportation system: a program designed to allow greater utilization of the many airports that do not possess a control tower

software development process: an implementation of the *software lifecycle*

software lifecycle: overall approach to software development

stability: the quality of returning to an original *angle of attack*, *slip*, or *bank* after a disturbance from an equilibrium

stability augmentation system: software system that adjusts control surfaces to compensate for *static instability* or *dynamic instability* of an aircraft's design

stall: condition during which the wings of an aircraft no longer generate lift because of their angle to the airstream

static code analysis: the automated analysis of program source by a software tool that does not require execution of the code

static redundancy: see *N-modular redundancy*

static stability: the quality of initially moving towards a *trimmed* state immediately after a disturbance

surveillance: using the position of a known, outside entity to derive a vehicle's state vector

tailplane: see *horizontal stabilizer*

terminal radar control center: a center responsible for management of aircraft in the airspace around an airport (terminal) using airport radar

terrestrial radio navigation system: earth-based *radio navigation systems*

thrust: the force created by an aircraft's engines

time-triggered bus: bus that interacts with *nodes* according to an internal time schedule

total drag: the sum of all of the types of drag acting on an aircraft

total stability: an aircraft that is both *statically stable* and *dynamically stable*

transient fault: fault that manifests itself briefly and might not manifest itself again

trimmed: term referring to an aircraft state in which the *angle of attack* and *bank angle* are both zero

triple-modular redundancy: *N-modular redundancy* where N = 3

validation: the process of checking whether a system provides the functionality that is actually needed

verification: determining whether a system implements the required functionality correctly

vertical navigation: navigation in the *xz*-plane, along the *pitch* axis

vertical separation: separation of two aircraft in altitude

vertical stabilizer: portion of the tail section responsible controlling the left and right movement of the nose of the aircraft

visual flight rules: flight classification under which a pilot may navigate and land using visual cues

visual meteorological conditions: weather conditions under which *visual flight rules* are allowed

wake disturbance: disturbance in the airflow caused by the passage of an aircraft

wave drag: drag affecting flight as an aircraft approaches the speed of sound, starting around Mach 0.75

waypoint: an intermediate destination in a *flight plan*
weight: the downward force of gravity, acting in opposition to *lift*
yaw: movement around the axis running vertically through an aircraft