UNIQUE DEPENDABILITY ISSUES FOR COMMERCIAL AIRPLANE FLY BY WIRE SYSTEMS

Ying C. (Bob) Yeh
Boeing Commercial Airplanes, Seattle, WA, USA – ying.c.yeh@boeing.com

Abstract: The fundamental concept of dependability is applied to the design of commercial airplane FBW systems beyond the lessons learned from the NASA FBW and industry/military FBW research and development projects. The considerations of generic errors and common mode failures play important role for configuring commercial airplane FBW system architectures and the FBW computer architectures.

Key words: Dependability, Fly-By-Wire (FBW)

1. INTRODUCTION

The NASA FBW projects provide the fundamental framework for functional integrity and functional availability requirements [1], [2] for the FBW computers. The Byzantine General Problems [3] and its solutions are illustrated in [1], [3], [4]. Further the lessons learned from the military FBW project [5] and other industry/academic experiences in dealing with generic faults [6], near-coincidence fault [7], and design paradigm [8] provide ground rules or derived design requirements for Boeing commercial airplane FBW programs [9], [10], [11].

A tutorial of fundamental concepts of dependability [12] can be used for referenced discussions. Two unique design requirements or design considerations for Commercial Airplane FBW are that of generic error/fault and common mode failure. The purpose of this article is to describe how

these two requirements/considerations play an important role for the Commercial Airplane FBW computers [13], [14], [15].

2. GENERIC ERROR AND DISSIMILARITY CONSIDERATIONS

The concept of design diversity [16] [17] has played a central role in academic research and its follow on experiments [18] [19] while the commercial airplane industry is using dissimilarity for flight critical systems, such as Autopilot computers and the FBW research. The experiments [18] [19] has influenced the final decision for the 777 FBW system design [13]. The Airbus [15] and Boeing FBW computers design considerations for generic errors and dissimilarity considerations are studied [20] and can be summarized as follows.

Two types of computers are used in the A320 FBW system: the ELAC (Elevator and Aileron computers) and the SEC (Spoiler and Elevator computers). The ELAC is produced by Thomson-CSF using Motorola 68010 processor, and the SEC is produced by SFENA/Aerospatiale using Intel 80186 processor. Each computer consists of two channels: control channel and monitor channel. The software and its programming language of the control channel are different from that of the monitor channel. Likewise the software of ELAC is different from that of SEC. Thus at software level, the architecture leads to the use of 4 software packages.

Two types of computers are also used on A340: the PRIM (primary computers) and SEC (secondary computers). The basic design philosophy is similar to A320. The PRIM uses Intel 80386 processors with a difference in software. Further the control channel is programmed in Assembler, while the monitor channel is programmed in PL/M. The SEC uses Intel 80186 processors. Assembly language is used for control channel, and Pascal is used for the monitor channel. Also for dissimilarity reasons, only the PRIM computer is coded automatically (the SEC being coded manually) and that the PRIM automatic coding tool has two different coded translators, one for control channel and another for monitor channel.

In addition to the ELAC and SEC of the A320, two computers are used for rudder control (FAC). On A330 and A340 FBW, these rudder control functions are integrated in the PRIM and SEC.

The overview of Boeing 777 Primary Flight Control System (or FBW) is depicted in Figure 1. The Boeing FBW system design considerations [13] extend the concept of triple hardware resources (hydraulics, airplane electrical power, FBW ARINC 629 bus) to triple dissimilar processors and their Ada compilers to construct triple-triple redundant PFC (primary flight

computer) [14]. Further, dissimilarity is invoked in the design and implementation of the PFC system where it is judged to be a necessary feature to satisfy critical minds of Boeing engineers. The design diversity issue [11] is integrated to the system design issue of dealing with all possible errors for a complex flight controls systems, experienced in Boeing and in industry/academia.

3. COMMON MODE FAILURE AND SINGLE POINT FAILURE

Common mode or common area faults [21] are considered for multiple redundant systems such as the FBW. Airplane susceptibility to common mode and common area damage is addressed by designing the systems to both component and functional separation requirement. This includes criteria for providing installations resistant to maintenance crew error or mishandling, such as the followings:

- impact of objects
- electrical faults
- electrical power failure
- electromagnetic environment
- lightning strike
- hydraulic failure
- structure damage
- radiation environment in the atmosphere
- ash cloud environment in the atmosphere
- fire
- rough or unsafe installation and maintenance

The single point failure consideration is integrated to the safety requirements. For instance, the derived 777 PFC safety requirements include numerical and non-numerical requirements as follows.

Safety requirements apply to PFC failures which could preclude continued safe flight and landing, and include both passive failures (loss of function without significant immediate airplane transient) and active failures (malfunction) with significant immediate airplane transient).

The numerical probability requirements are both 1.0E-10 per flight hour for functional integrity requirement (relative to active failures affecting 777 Airplane Structure) and functional availability requirement (relative to passive failures).

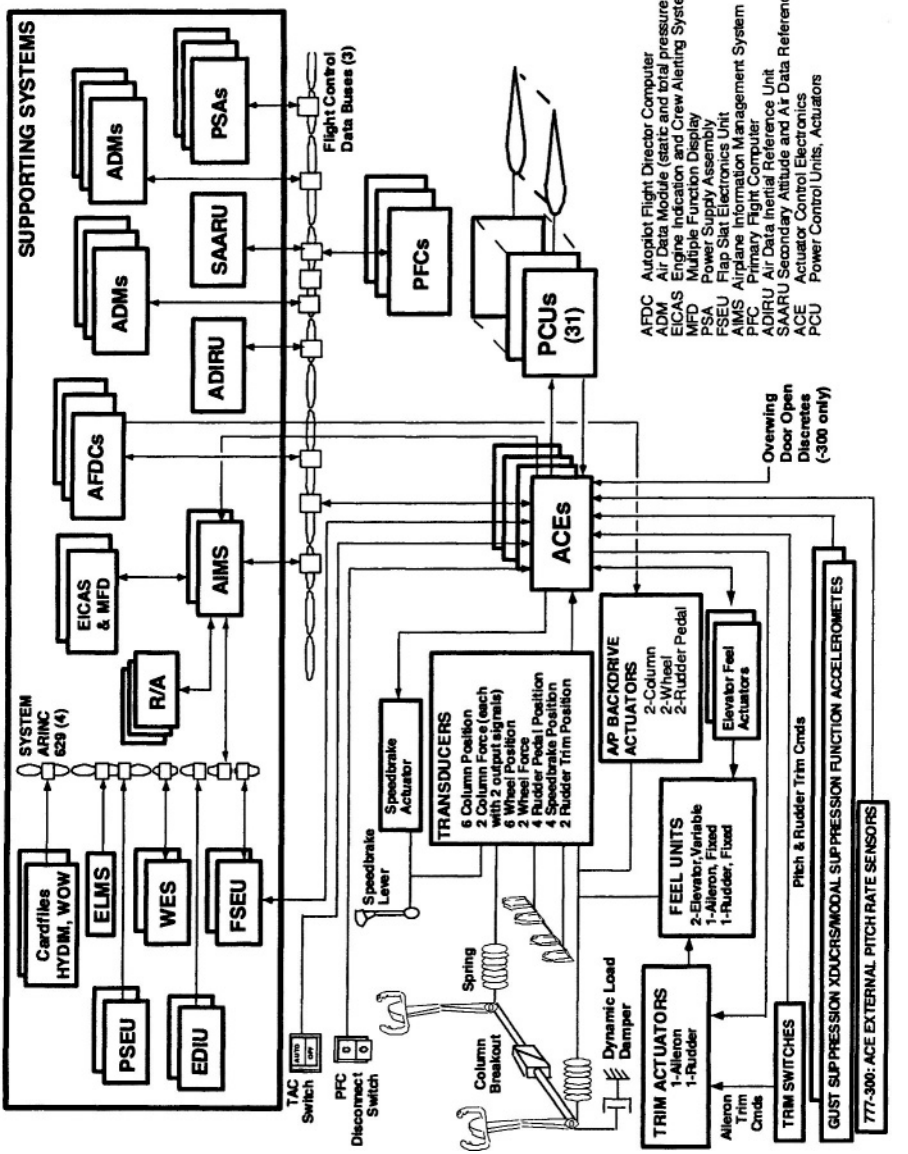

Figure 1. 777 Primary Flight Control System

The PFC is designed to comply with the following non-numerical safety requirements described as follows:

a) No single fault, including common mode hardware fault, regardless of probability of occurrence, shall result in an erroneous (assumed active failures for the worst case) transmission of output signals without a failure indication.
b) No single fault, including common mode hardware fault, regardless of probability of occurrence shall result in loss of function in more than one PFC channel.

Extensive validation process [22] is undertaken to comply with the 777 Flight Controls certification plan approved by the certification agencies, and to satisfy critical minds of Boeing engineers.

4. SUMMARY DISCUSSION: SIMPLEX DIRECT MODE CONTROL

The virtue of simplicity [23] is not lost on the complex FBW systems due to extremely stringent numerical and non-numerical safety requirements and considerations of generic errors, common mode failure, and single point failure.

The 777 Primary Flight Control Modes is shown in Table 1. The Direct Control mode provides simplex control law in the event of occurrences of "known or unknown" combinations of generic errors and common mode failure, or in the event of pilot decision to engage PFC Disconnect Switch for whatever reasons.

Table 1. 777 Primary Flight Control Modes

	PITCH	ROLL	YAW
NORMAL Control	CONTROL C* Maneuver Cmd with Speed Feedback Manual Trim for Speed Variable Feel	CONTROL Surface Cmds Manual Trim Fixed Feel	CONTROL Surface Cmd Ratio Changer Wheel/Rudder Cross Tie Manual Trim Yaw Damping Fixed Feel Gust Supression
	ENVELOPE PROTECTION Stall Overspeed	ENVELOPE PROTECTION Bank Angle	ENVELOPE PROTECTION Thrust Asymmetry
	AUTOPILOT Backdrive	AUTOPILOT Backdrive	AUTOPILOT Backdrive
SECONDARY Control	CONTROL Surface Cmd (Augmented) Flaps Up/Down Gain Direct Stabilizer Trim Flaps Up/Down Feel	CONTROL Surface Cmd Manual Trim Fixed Feel	CONTROL Surface Cmds, Flaps Up/Down Gain PCU Pressure Reducer Manual Trim Yaw Rate Damper (if available)
DIRECT Control	CONTROL Surface Cmd (Augmented) Flaps Up/Down Gain Direct Stabilizer Trim Flaps Up/Down Feel	CONTROL Surface Cmd Manual Trim Fixed Feel	CONTROL Surface Cmds, Flaps Up/Down Gain PCU Pressure Reducer Manual Trim

The 777 Actuator Control Electronics Architecture is shown in Figure 2. The hardware circuitry for Direct Mode control function in ACE is designed to be as simple as possible. Further the hardware system architectures resided in all critical LRUs supporting the Normal Mode function are designed to be fail-passive so that the ISM (input signal management function) can direct/fail to the defaulted condition of Direct Mode control.

REFERENCES

[1] J.H. Wenseley et al "SIFT: Design and Analysis of a Fault Tolerant Computer for Aircraft Control" Proceeding of the IEEE, Vol. 66, No. 10, October 1978.

[2] A.L. Hopkins Jr., T.B. Smith,III, J.H. Lala, "FTMP-A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft", Proceeding of the IEEE, Vol. 66, No. 10, October 1978.

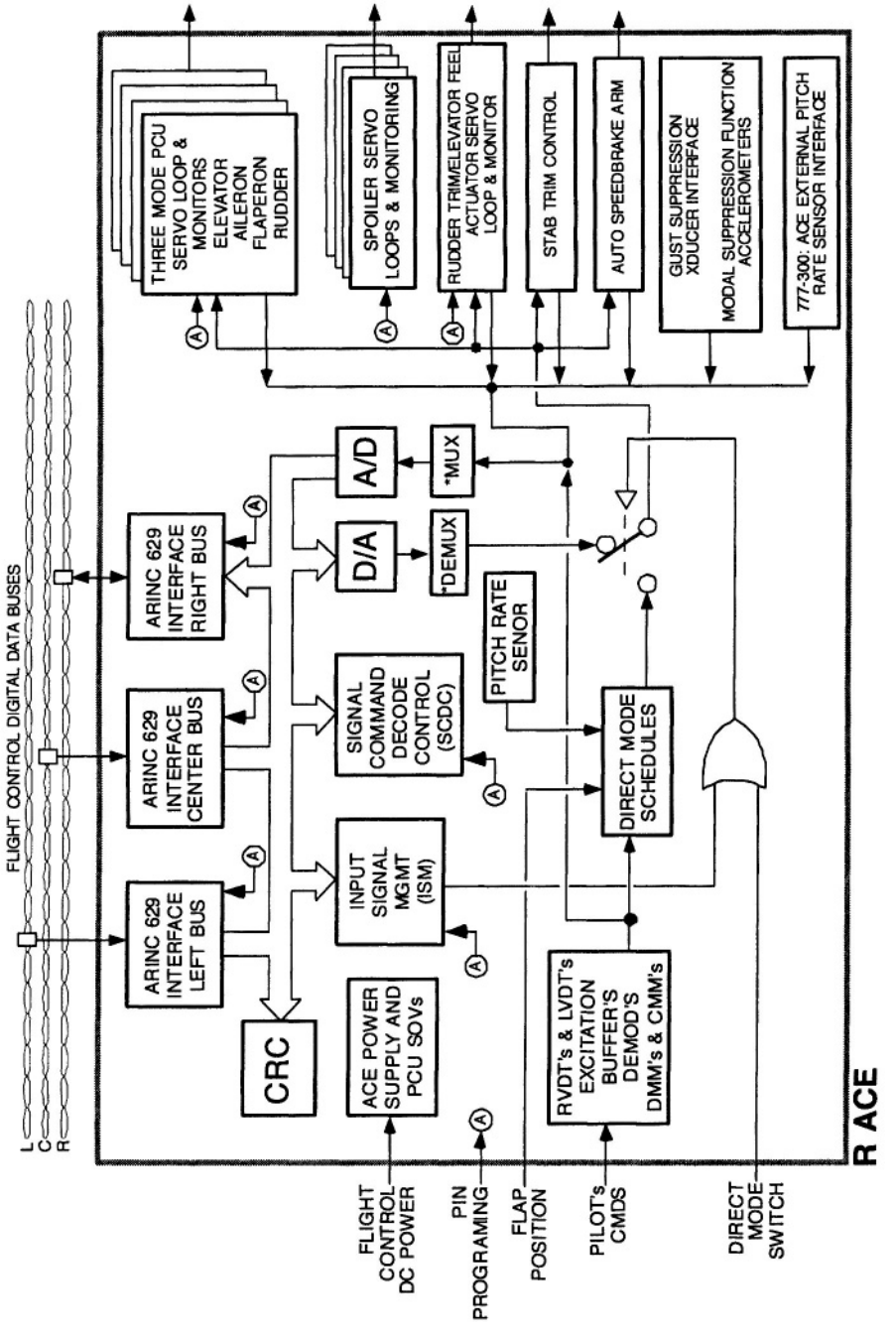

Figure 2. 777 Actuator Control Electronics Architecture

[3] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem", ACM Trans. On Programming Languages and Systems, Vol.4, No.3, July 1982.
[4] R. Kieckhafer, C. Walter, A. Finn, and P. Thambidurai, "The MAFT Architecture for Distributed Fault Tolerance", IEEE Trans. On Computers, 37(4):398-405, 1988.
[5] J.H. Watson, W.J. Yousey, A.M. Arabian, and T.M. Schindler, "Lessons learned from development of AFTI/F-16 Digital Flight Control System", 5th digital avionics systems conference, Seattle, Washington, 1983.
[6] S.S. Osder, "Generic Faults and Architecture Design Considerations in Flight-Critical Systems", AIAA Journal of Guidance, Vol.6, No.2, March-April 1983.
[7] J. McGough, "Effects of Near-Coincident Faults in Multiprocessor Systems", Fifth AIAA/IEEE Digital Avionics Conference, October 1983.
[8] A. Avizienis, "A Design Paradigm for Fault-Tolerant Systems", AIAA Computers in Aerospace Conference, October 1987, Paper 87-2764.
[9] R.J. Bleeg, "Commercial Jet Transport Fly-By-Wire Architecture Consideration", Ninth AIAA/IEEE Digital Avionics Conference, October 1988.
[10] J. McWha, "777 Systems Overview", RAeS Presentation, November 1993.
[11] Y.C. Yeh, "Design Considerations in 777 Fly-By-Wire Computers", 3rd IEEE International High-Assurance Systems Engineering Conference, Washington, DC, October 1988.
[12] A. Avizienis, J.-C. Laprie, and B. Randell, "Fundamental Concepts of Dependability", Research Report NO1145, LAAS-CNRS, April 2001.
[13] Y.C. Yeh, "Dependability of the 777 Primary Flight Control System", in Dependable Computing for Critical Applications (DCCS-5), Dependable Computing and Fault-Tolerant Systems, 10, pp.3-17, IEEE Computer Society Press, 1998.
[14] Y.C. Yeh, "Triple-Triple Redundant 777 Primary Flight Computers", 1996 IEEE Aerospace Applications Conference, February 1996.
[15] D. Briere and P. Traverse, "AIRBUS A320/A330/A340 Electrical Flight Controls – A Family of Fault-Tolerant Systems", in FTCS-23, pp.616-23, IEEE Computer Society Press, 1993.
[16] A. Avizienis, "Design Diversity – the Challenge of the Eighties", FTCS-12, pages 44-45, June 1982.
[17] A. Avizienis, and J.P.J Kelly, "Fault Tolerance by Design Diversity: Concepts and Experiments", Computer, August 1984.
[18] A. Avizienis, M.R. Lyu, and W. Schutz, "In Search of Effective Diversity: A Six-Language Study of Fault-Tolerant of Fault-Tolerant Flight Computer Software", FTCS-18, 1988.
[19] J.C. Knight, N.G. Leveson, "An Experimental Evaluation of the Assumption of Independence in Multiversion Programming", IEEE Trans. on Software Engineering, January 1986.
[20] D. Powell, J.P. Blanquart, Y. Crouzet, and J.C. Fabre, "Architecture Approaches for using COTS Components in Critical Applications", 11th European Workshop on Dependable Computing (EWDC-11), Budapest, Hungary, 2000.
[21] SAE ARP4761, "Guidelines and Methods for Conducting The Safety Assessment Process on Civil Airborne Systems and Equipment", Society of Automotive Engineers.
[22] H. Buss, et al, "777 Flight Controls Validation Process", Fourteen AIAA/IEEE Digital Avionics System Conference, November 1995.
[23] L. Sha, B. Goodenough, and B. Pollak, "Simplex Architecture: Meeting the Challenges of Using COTS in High-Reliability Systems", Cross Talk, The Journal of Defense Software Engineering, April 1998.